Общество с ограниченной ответственностью «Платформа Данных Селена» 123022, г. Москва, вн.тер.г. муниципальный округ Пресненский, ул. 2-я Звенигородская, д. 13, стр. 37 ИНН 9703195303, КПП 770301001, ОГРН 1247700725491

Платформа Данных Селена

РУКОВОДСТВО АДМИНИСТРАТОРА ПЛАТФОРМЫ

Версия программы 3.0

Москва
2026

Аннотация

Настоящее руководство предназначено для администраторов Платформы Данных Селена (далее Selena), и описывает задачи по администрированию и настройке интеграций.

Материал ориентирован на специалистов, отвечающих за инфраструктуру и эксплуатацию платформы, и предполагает знание основ Linux и представление об архитектуре распределённых систем. Руководство может использоваться как для последовательного изучения, так и в качестве справочника при решении практических задач.

© ООО «Платформа Данных Селена», 2026

Оглавление

1 Интеграции

1.1 Аутентификация CSP

1.1.1 Аутентификация для доступа к ресурсам AWS

1.1.2 Аутентификация для доступа к Microsoft Azure Storage

1.1.3 Аутентификация для доступа к Google Cloud Storage

1.2 Data Lakehouse

1.3 Потоковые данные

1.3.1 Apache Flink

1.3.3 Apache Kafka

Kafka connector

Routine Load

AutoMQ Kafka

1.3.4 Потоковая загрузка с Selena Pipe

HDFS

S3

1.3.5 Выгрузка данных из RisingWave в Selena

1.4 Пакетная загрузка данных

1.4.1 Selena Spark Connector

1.4.3 Apache Spark Load

1.5 Loading_tools_integration

1.6.5 Selena Migration Tool (SMT)

1.6.6 DataX writer

1.6.7 BladePipe

Коннектор Kettle

1.7 Business Intelligence

1.7.1 Apache Superset

1.7.2 FineBI

1.7.3 Hex

1.7.4 Querybook

1.7.5 QuickBI

1.7.6 Tableau

1.7.7 Metabase

1.8 IDE-инструменты

1.8.1 DataGrip

1.8.2 Dataphin

1.8.3 DBeaver

1.8.4 Jupyter

1.9 Мониторинг с помощью Datadog

1.10 Apache Airflow

1.11 Dbt

1.12 Управление схемами и миграция с помощью SQLAlchemy и Alembic

2 Администрирование

2.1 Конфигурация

- 2.1.1 Конфигурация FE
- 2.1.2 Конфигурация BE

2.2 Управление

- 2.2.1 Logs
- 2.2.2 Масштабирование cluster
- 2.2.3 Управление чёрным списком BE и CN
- 2.2.4 Резервное копирование и восстановление данных
- 2.2.5 Cluster Snapshot
- 2.2.6 Мониторинг и алертинг с Prometheus и Grafana
  - Общие метрики мониторинга
  - Метрики мониторинга для асинхронных материализованных представлений
  - Метрики мониторинга для cluster с разделяемыми данными
  - Управление алертами
- 2.2.7 Управление журналами аудита в Selena через AuditLoader
- 2.2.8 Включение доступа по FQDN
- 2.2.9 Настройка часового пояса
- 2.2.10 Мониторинг и управление большими запросами
- 2.2.11 Compaction для cluster с общими данными
- 2.2.12 Плавное завершение работы
- 2.2.13 Управление ресурсами
  - Группа ресурсов
  - Очереди запросов
  - Управление запросами
  - Управление памятью
  - Сброс на диск
  - Балансировка нагрузки
  - Управление репликами
  - Управление чёрным списком
  - Добавление меток на BE
  - Файловый менеджер

2.3 Восстановление данных

2.4 Восстановление метаданных

2.5 Инструмент миграции данных между cluster

2.6 HTTP-интерфейс

- 2.6.4 API деталей запроса

2.7 SQL Digest

2.8 Привилегии и аутентификация

- 2.8.1 Authentication
  - Нативная аутентификация
  - Аутентификация с Security Integration
  - Аутентификация LDAP
  - Аутентификация JSON Web Token
  - Аутентификация OAuth 2.0
- 2.8.2 Authorization
  - Обзор привилегий
  - Элементы привилегий
  - Встроенные роли
  - Управление привилегиями пользователей
  - Управление правами доступа с помощью Apache Ranger

Часто задаваемые вопросы о привилегиях
2.8.3 Аутентификация групп пользователей
2.8.4 SSL-аутентификация

3 Benchmarks

3.1 SSB Flat-table Benchmarking
3.2 TPC-DS Benchmarking

1 Интеграции

Интеграции, начиная от бизнес-аналитики, до потоковых данных, до вашего хранилища данных и многого другого.

1.1 Аутентификация CSP

1.1.1 Аутентификация для доступа к ресурсам AWS

Selena поддерживает три метода аутентификации для интеграции с ресурсами AWS: аутентификация на основе профиля экземпляра, аутентификация на основе предполагаемой роли и аутентификация на основе пользователя IAM. В этом разделе описывается, как настроить учётные данные AWS с использованием этих методов аутентификации.

Методы аутентификации

Аутентификация на основе профиля экземпляра

Метод аутентификации на основе профиля экземпляра позволяет вашему cluster Selena наследовать привилегии, указанные в профиле экземпляра EC2, на котором работает cluster. Теоретически любой пользователь cluster, который может войти в cluster, может выполнять разрешённые действия с вашими ресурсами AWS в соответствии с политиками AWS IAM, которые вы настроили. Типичный сценарий для этого случая использования - когда вам не нужен контроль доступа к ресурсам AWS между несколькими пользователями cluster в cluster. Этот метод аутентификации означает, что изоляция внутри одного cluster не требуется.

Однако этот метод аутентификации всё ещё можно рассматривать как решение безопасного контроля доступа на уровне cluster, потому что администратор cluster контролирует, кто может войти в cluster.

Аутентификация на основе предполагаемой роли

В отличие от аутентификации на основе профиля экземпляра, метод аутентификации на основе предполагаемой роли поддерживает принятие роли AWS IAM для получения доступа к вашим ресурсам AWS. Подробнее см. Принятие роли.

Аутентификация на основе пользователя IAM

Метод аутентификации на основе пользователя IAM поддерживает использование учётных данных пользователя IAM для получения доступа к вашим ресурсам AWS. Подробнее см. Пользователи IAM.

Подготовка

Сначала найдите роль IAM, связанную с экземпляром EC2, на котором работает ваш cluster Selena (эта роль далее в этом разделе называется ролью экземпляра EC2), и получите ARN роли. Вам понадобится роль экземпляра EC2 для аутентификации на основе профиля экземпляра и нужны роль экземпляра EC2 и её ARN для аутентификации на основе предполагаемой роли.

Следующий шаг - создайте политику IAM на основе типа ресурса AWS, к которому вы хотите получить доступ, и конкретного сценария операции в Selena. Политика в AWS IAM объявляет набор разрешений на определённый ресурс AWS. После создания политики вам нужно прикрепить её к роли или пользователю IAM. Таким образом, роль или пользователь IAM получает разрешения, объявленные в политике, для доступа к указанному ресурсу AWS.

> **ВНИМАНИЕ**
>
> Для выполнения этих подготовительных действий у вас должно быть разрешение на вход в консоль AWS IAM и редактирование пользователей и ролей IAM.

Для политики IAM, которая вам понадобится для доступа к определённому ресурсу AWS, см. следующие разделы:

- Пакетная загрузка данных из AWS S3
- Чтение/запись AWS S3
- Интеграция с AWS Glue

Подготовка для аутентификации на основе профиля экземпляра

Прикрепите политики IAM для доступа к необходимым ресурсам AWS к роли экземпляра EC2.

Подготовка для аутентификации на основе предполагаемой роли

Создание ролей IAM и прикрепление к ним политик

Создайте одну или несколько ролей IAM в зависимости от ресурсов AWS, к которым вы хотите получить доступ. См. Создание ролей IAM. Затем прикрепите политики IAM для доступа к необходимым ресурсам AWS к созданным вами ролям IAM.

Например, вы хотите, чтобы ваш cluster Selena имел доступ к AWS S3 и AWS Glue. В этой ситуации вы можете создать одну роль IAM (например, `s3_assumed_role`) и прикрепить к этой роли политику для доступа к AWS S3 и политику для доступа к AWS Glue. Альтернативно вы можете создать две разные роли IAM (например, `s3_assumed_role` и `glue_assumed_role`) и прикрепить эти политики к двум разным ролям соответственно (то есть прикрепить политику для доступа к AWS S3 к `s3_assumed_role` и политику для доступа к AWS Glue к `glue_assumed_role`).

Созданные вами роли IAM будут приняты ролью экземпляра EC2 cluster Selena для доступа к указанным ресурсам AWS.

В этом разделе предполагается, что вы создали только одну предполагаемую роль `s3_assumed_role` и добавили к этой роли политику для доступа к AWS S3 и политику для доступа к AWS Glue.

Настройка доверительных отношений

Настройте вашу предполагаемую роль следующим образом:

1. Войдите в консоль AWS IAM.

2. На левой панели навигации выберите **Access management** > **Roles**.

3. Найдите предполагаемую роль (`s3_assumed_role`) и нажмите на её имя.

4. На странице сведений о роли нажмите вкладку **Trust relationships**, а затем на вкладке **Trust relationships** нажмите **Edit trust policy**.

5. На странице **Edit trust policy** удалите существующий JSON-документ политики и вставьте следующую политику IAM, в которой вы должны заменить `<cluster_EC2_iam_role_ARN>` на ARN роли экземпляра EC2, который вы получили выше. Затем нажмите **Update policy**.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "<cluster_EC2_iam_role_ARN>"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Если вы создали разные предполагаемые роли для доступа к разным ресурсам AWS, вам нужно повторить предыдущие шаги для настройки других предполагаемых ролей. Например, вы создали `s3_assumed_role` и `glue_assumed_role` для доступа к AWS S3 и AWS Glue соответственно. В этой ситуации вам нужно повторить предыдущие шаги для настройки `glue_assumed_role`.

Настройте вашу роль экземпляра EC2 следующим образом:

1. Войдите в консоль AWS IAM.

2. На левой панели навигации выберите **Access management** > **Roles**.

3. Найдите роль экземпляра EC2 и нажмите на её имя.

4. В разделе **Permissions policies** страницы сведений о роли нажмите **Add permissions** и выберите **Create inline policy**.

5. На шаге **Specify permissions** нажмите вкладку **JSON**, удалите существующий JSON-документ политики и вставьте следующую политику IAM, в которой вы должны заменить `<s3_assumed_role_ARN>` на ARN предполагаемой роли `s3_assumed_role`. Затем нажмите **Review policy**.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["sts:AssumeRole"],
            "Resource": [
                "<s3_assumed_role_ARN>"
            ]
        }
    ]
}
```

Если вы создали разные предполагаемые роли для доступа к разным ресурсам AWS, вам нужно указать ARN всех этих предполагаемых ролей в элементе **Resource** предыдущей политики IAM и разделить их запятой (,). Например, вы создали `s3_assumed_role` и `glue_assumed_role` для доступа к AWS S3 и AWS Glue соответственно. В этой ситуации вам нужно указать ARN `s3_assumed_role` и ARN `glue_assumed_role` в элементе **Resource** в следующем формате: `"<s3_assumed_role_ARN>","<glue_assumed_role_ARN>"`.

6. На шаге **Review Policy** введите имя политики и нажмите **Create policy**.

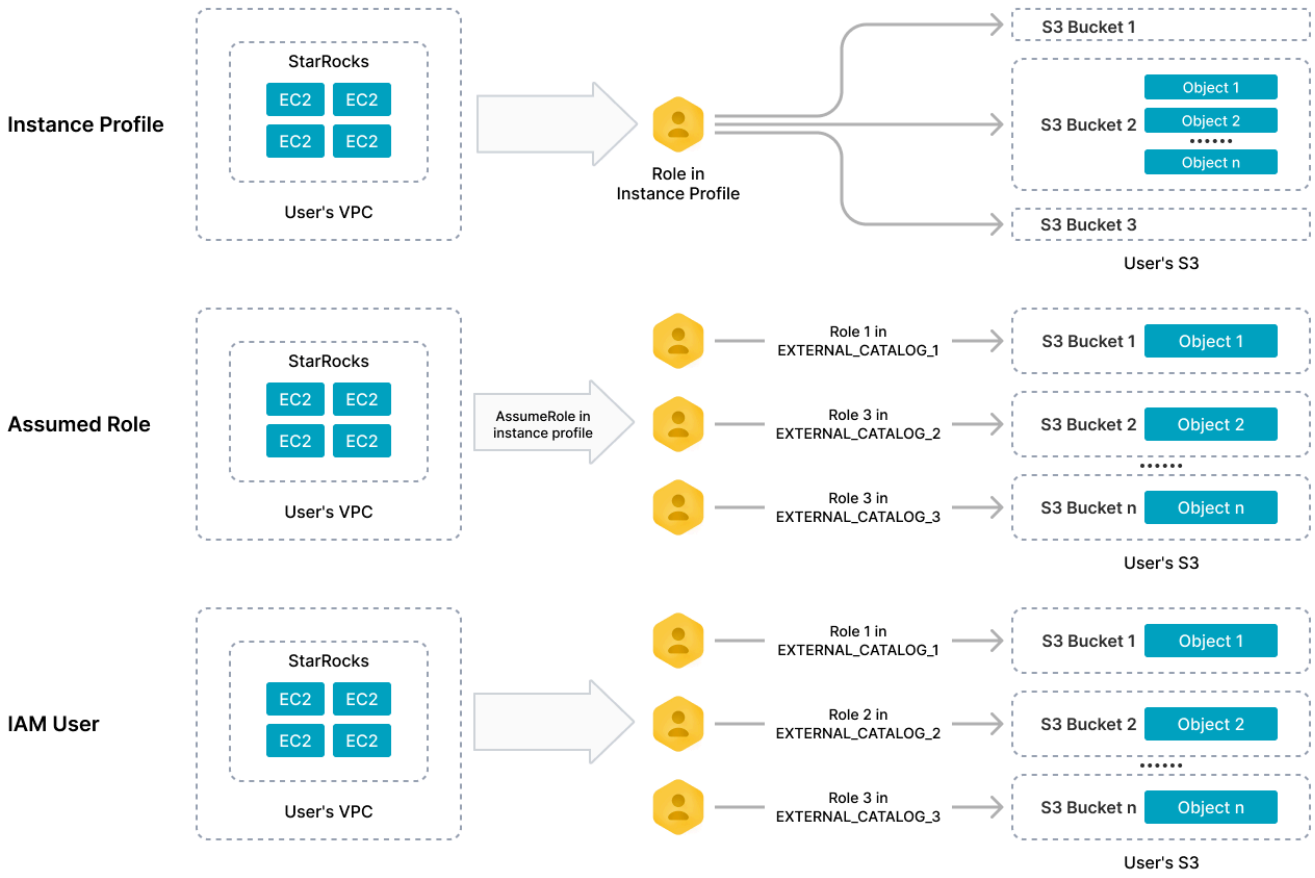
Подготовка для аутентификации на основе пользователя IAM

Создайте пользователя IAM. См. Создание пользователя IAM в вашей учётной записи AWS.

Затем прикрепите политики IAM для доступа к необходимым ресурсам AWS к созданному вами пользователю IAM.

Сравнение методов аутентификации

Следующая диаграмма даёт общее объяснение различий в механизме между аутентификацией на основе профиля экземпляра, аутентификацией на основе предполагаемой роли и аутентификацией на основе пользователя IAM в Selena.



Построение подключений к ресурсам AWS

Параметры аутентификации для доступа к AWS S3

В различных сценариях, когда Selena необходимо интегрироваться с AWS S3, например, при создании внешних каталогов или файловых внешних таблиц, или при загрузке, резервном копировании или восстановлении данных из AWS S3, настройте параметры аутентификации для доступа к AWS S3 следующим образом:

- Для аутентификации на основе профиля экземпляра установите `aws.s3.use_instance_profile` в `true`.
- Для аутентификации на основе предполагаемой роли установите `aws.s3.use_instance_profile` в `true` и настройте `aws.s3.iam_role_arn` как ARN предполагаемой роли, которую вы используете для доступа к AWS S3 (например, ARN предполагаемой роли `s3_assumed_role`, которую вы создали выше).
- Для аутентификации на основе пользователя IAM установите `aws.s3.use_instance_profile` в `false` и настройте `aws.s3.access_key` и `aws.s3.secret_key` как ключ доступа и секретный ключ вашего пользователя AWS IAM.

В следующей таблице описаны параметры.

Параметр	Обязательный	Описание
aws.s3.use_instance_profile	Да	Указывает, следует ли включить метод аутентификации на основе профиля экземпляра и метод аутентификации на основе предполагаемой роли. Допустимые значения: `true` и `false`. Значение по умолчанию: `false`.
aws.s3.iam_role_arn	Нет	ARN роли IAM, имеющей привилегии на ваш бакет AWS S3. Если вы используете метод аутентификации на основе предполагаемой роли для доступа к AWS S3, вы должны указать этот параметр.
aws.s3.access_key	Нет	Ключ доступа вашего пользователя IAM. Если вы используете метод аутентификации на основе пользователя IAM для доступа к AWS S3, вы должны указать этот параметр.
aws.s3.secret_key	Нет	Секретный ключ вашего пользователя IAM. Если вы используете метод аутентификации на основе пользователя IAM для доступа к AWS S3, вы должны указать этот параметр.

Параметры аутентификации для доступа к AWS Glue

В различных сценариях, когда Selena необходимо интегрироваться с AWS Glue, например, при создании внешних каталогов, настройте параметры аутентификации для доступа к AWS Glue следующим образом:

- Для аутентификации на основе профиля экземпляра установите `aws.glue.use_instance_profile` в `true`.
- Для аутентификации на основе предполагаемой роли установите `aws.glue.use_instance_profile` в `true` и настройте `aws.glue.iam_role_arn` как ARN предполагаемой роли, которую вы используете для доступа к AWS Glue (например, ARN предполагаемой роли `glue_assumed_role`, которую вы создали выше).
- Для аутентификации на основе пользователя IAM установите `aws.glue.use_instance_profile` в `false` и настройте `aws.glue.access_key` и `aws.glue.secret_key` как ключ доступа и секретный ключ вашего пользователя AWS IAM.

В следующей таблице описаны параметры.

Параметр	Обязательный	Описание
aws.glue.use_instance_profile	Да	Указывает, следует ли включить метод аутентификации на основе профиля экземпляра и аутентификацию на основе предполагаемой роли. Допустимые значения: `true` и `false`. Значение по умолчанию: `false`.
aws.glue.iam_role_arn	Нет	ARN роли IAM, имеющей привилегии на ваш AWS Glue Data Catalog. Если вы используете метод аутентификации на основе предполагаемой роли для доступа к AWS Glue, вы должны указать этот параметр.
aws.glue.access_key	Нет	Ключ доступа вашего пользователя AWS IAM. Если вы используете метод аутентификации на основе пользователя IAM для доступа к AWS Glue, вы должны указать этот параметр.
aws.glue.secret_key	Нет	Секретный ключ вашего пользователя AWS IAM. Если вы используете метод аутентификации на основе пользователя IAM для доступа к AWS Glue, вы должны указать этот параметр.

Примеры интеграции

Внешний каталог

Создание внешнего каталога в вашем cluster Selena означает построение интеграции с целевой системой озера данных, которая состоит из двух ключевых компонентов:

- Файловое хранилище, такое как AWS S3, для хранения файлов таблиц
- Хранилище метаданных, такое как Hive metastore или AWS Glue, для хранения метаданных и расположений файлов таблиц

Selena поддерживает следующие типы каталогов:

- Каталог Hive
- Каталог Iceberg
- Каталог Hudi
- Каталог Delta Lake

Следующие примеры создают каталог Hive с именем `hive_catalog_hms` или `hive_catalog_glue`, в зависимости от типа используемого хранилища метаданных, для запроса данных из вашего cluster Hive. Подробный синтаксис и параметры см. в Каталог Hive.

Аутентификация на основе профиля экземпляра

- Если вы используете Hive metastore в вашем cluster Hive, выполните команду, подобную следующей:

```
CREATE EXTERNAL CATALOG hive_catalog_hms
PROPERTIES
(
    "type" = "hive",
    "aws.s3.use_instance_profile" = "true",
    "aws.s3.region" = "us-west-2",
    "hive.metastore.uris" = "thrift://xx.xx.xx.xx:9083"
);
```

- Если вы используете AWS Glue в вашем cluster Amazon EMR Hive, выполните команду, подобную следующей:

```
CREATE EXTERNAL CATALOG hive_catalog_glue
PROPERTIES
(
    "type" = "hive",
    "aws.s3.use_instance_profile" = "true",
    "aws.s3.region" = "us-west-2",
    "hive.metastore.type" = "glue",
    "aws.glue.use_instance_profile" = "true",
    "aws.glue.region" = "us-west-2"
);
```

Аутентификация на основе предполагаемой роли

- Если вы используете Hive metastore в вашем cluster Hive, выполните команду, подобную следующей:

```
CREATE EXTERNAL CATALOG hive_catalog_hms
PROPERTIES
(
    "type" = "hive",
    "aws.s3.use_instance_profile" = "true",
    "aws.s3.iam_role_arn" =
"arn:aws:iam::081976408565:role/s3_assumed_role",
    "aws.s3.region" = "us-west-2",
    "hive.metastore.uris" = "thrift://xx.xx.xx.xx:9083"
);
```

- Если вы используете AWS Glue в вашем cluster Amazon EMR Hive, выполните команду, подобную следующей:

```
CREATE EXTERNAL CATALOG hive_catalog_glue
PROPERTIES
(
    "type" = "hive",
    "aws.s3.use_instance_profile" = "true",
    "aws.s3.iam_role_arn" =
"arn:aws:iam::081976408565:role/s3_assumed_role",
    "aws.s3.region" = "us-west-2",
    "hive.metastore.type" = "glue",
    "aws.glue.use_instance_profile" = "true",
    "aws.glue.iam_role_arn" =
"arn:aws:iam::081976408565:role/glue_assumed_role",
    "aws.glue.region" = "us-west-2"
);
```

Аутентификация на основе пользователя IAM

- Если вы используете Hive metastore в вашем cluster Hive, выполните команду, подобную следующей:

```
CREATE EXTERNAL CATALOG hive_catalog_hms
PROPERTIES
(
    "type" = "hive",
    "aws.s3.use_instance_profile" = "false",
    "aws.s3.access_key" = "<iam_user_access_key>",
    "aws.s3.secret_key" = "<iam_user_access_key>",
    "aws.s3.region" = "us-west-2",
    "hive.metastore.uris" = "thrift://xx.xx.xx.xx:9083"
);
```

- Если вы используете AWS Glue в вашем cluster Amazon EMR Hive, выполните команду, подобную следующей:

```
CREATE EXTERNAL CATALOG hive_catalog_glue
PROPERTIES
(
    "type" = "hive",
    "aws.s3.use_instance_profile" = "false",
    "aws.s3.access_key" = "<iam_user_access_key>",
    "aws.s3.secret_key" = "<iam_user_secret_key>",
    "aws.s3.region" = "us-west-2",
    "hive.metastore.type" = "glue",
    "aws.glue.use_instance_profile" = "false",
    "aws.glue.access_key" = "<iam_user_access_key>",
    "aws.glue.secret_key" = "<iam_user_secret_key>",
    "aws.glue.region" = "us-west-2"
);
```

Файловая внешняя таблица

Файловые внешние таблицы должны создаваться во внутреннем каталоге с именем `default_catalog`.

Следующие примеры создают файловую внешнюю таблицу с именем `file_table` в существующей базе данных с именем `test_s3_db`. Подробный синтаксис и параметры см. в Файловая внешняя таблица.

Аутентификация на основе профиля экземпляра

Выполните команду, подобную следующей:

```
CREATE EXTERNAL TABLE test_s3_db.file_table
(
    id varchar(65500),
    attributes map<varchar(100), varchar(2000)>
)
ENGINE=FILE
PROPERTIES
(
    "path" = "s3://selena-test/",
    "format" = "ORC",
    "aws.s3.use_instance_profile" = "true",
    "aws.s3.region" = "us-west-2"
);
```

Аутентификация на основе предполагаемой роли

Выполните команду, подобную следующей:

```
CREATE EXTERNAL TABLE test_s3_db.file_table
(
    id varchar(65500),
    attributes map<varchar(100), varchar(2000)>
)
ENGINE=FILE
PROPERTIES
(
    "path" = "s3://selena-test/",
    "format" = "ORC",
    "aws.s3.use_instance_profile" = "true",
    "aws.s3.iam_role_arn" =
"arn:aws:iam::081976408565:role/s3_assumed_role",
    "aws.s3.region" = "us-west-2"
);
```

Аутентификация на основе пользователя IAM

Выполните команду, подобную следующей:

```
CREATE EXTERNAL TABLE test_s3_db.file_table
(
    id varchar(65500),
    attributes map<varchar(100), varchar(2000)>
)
ENGINE=FILE
PROPERTIES
(
    "path" = "s3://selena-test/",
    "format" = "ORC",
    "aws.s3.use_instance_profile" = "false",
    "aws.s3.access_key" = "<iam_user_access_key>",
    "aws.s3.secret_key" = "<iam_user_secret_key>",
    "aws.s3.region" = "us-west-2"
);
```

Загрузка данных

Вы можете использовать LOAD LABEL для загрузки данных из AWS S3.

Следующие примеры загружают данные из всех файлов данных Parquet, хранящихся по пути `s3a://test-bucket/test_brokerload_ingestion`, в таблицу `test_ingestion_2` в существующей базе данных с именем `test_s3_db`. Подробный синтаксис и параметры см. в BROKER LOAD.

Аутентификация на основе профиля экземпляра

Выполните команду, подобную следующей:

```
LOAD LABEL test_s3_db.test_credential_instanceprofile_7
(
    DATA INFILE("s3a://test-bucket/test_brokerload_ingestion/*")
    INTO TABLE test_ingestion_2
    FORMAT AS "parquet"
)
WITH BROKER
(
    "aws.s3.use_instance_profile" = "true",
    "aws.s3.region" = "us-west-1"
)
PROPERTIES
(
    "timeout" = "1200"
);
```

Аутентификация на основе предполагаемой роли

Выполните команду, подобную следующей:

```
LOAD LABEL test_s3_db.test_credential_instanceprofile_7
(
    DATA INFILE("s3a://test-bucket/test_brokerload_ingestion/*")
    INTO TABLE test_ingestion_2
    FORMAT AS "parquet"
)
WITH BROKER
(
    "aws.s3.use_instance_profile" = "true",
    "aws.s3.iam_role_arn" =
"arn:aws:iam::081976408565:role/s3_assumed_role",
    "aws.s3.region" = "us-west-1"
)
PROPERTIES
(
    "timeout" = "1200"
);
```

Аутентификация на основе пользователя IAM

Выполните команду, подобную следующей:

```
LOAD LABEL test_s3_db.test_credential_instanceprofile_7
(
    DATA INFILE("s3a://test-bucket/test_brokerload_ingestion/*")
    INTO TABLE test_ingestion_2
    FORMAT AS "parquet"
)
WITH BROKER
(
    "aws.s3.use_instance_profile" = "false",
    "aws.s3.access_key" = "<iam_user_access_key>",
    "aws.s3.secret_key" = "<iam_user_secret_key>",
    "aws.s3.region" = "us-west-1"
)
PROPERTIES
(
    "timeout" = "1200"
);
```

1.1.2 Аутентификация для доступа к Microsoft Azure Storage

Начиная с версии v1.5.2, Selena может интегрироваться с Microsoft Azure Storage (Azure Blob Storage или Azure Data Lake Storage) в следующих сценариях:

- Пакетная загрузка данных из Azure Storage.
- Резервное копирование данных в Azure Storage и восстановление данных из Azure Storage.
- Запрос файлов Parquet и ORC в Azure Storage.
- Запрос таблиц Hive, Iceberg, Hudi и Delta Lake в Azure Storage.

Selena поддерживает следующие типы учётных записей Azure Storage:

- Azure Blob Storage
- Azure Data Lake Storage Gen1
- Azure Data Lake Storage Gen2

В этом разделе каталог Hive, файловая внешняя таблица и Broker Load используются в качестве примеров, показывающих, как Selena интегрируется с Azure Storage с использованием этих типов учётных записей Azure Storage. Информацию о параметрах в примерах см. в Каталог Hive, Файловая внешняя таблица и Broker Load.

Blob Storage

Selena поддерживает один из следующих методов аутентификации для доступа к Blob Storage:

- Shared Key
- SAS Token

> **ПРИМЕЧАНИЕ**
>
> При загрузке данных или прямом запросе файлов из Blob Storage вы должны использовать протокол wasb или wasbs для доступа к вашим данным:
>
> - Если ваша учётная запись хранения разрешает доступ по HTTP, используйте протокол wasb и записывайте путь к файлу как `wasb://<container>@<storage_account>.blob.core.windows.net/<path>/<file_name>`.
> - Если ваша учётная запись хранения разрешает доступ по HTTPS, используйте протокол wasbs и записывайте путь к файлу как `wasbs://<container>@<storage_account>.blob.core.windows.net/<path>/<file_name>`.

Shared Key

Внешний каталог

Настройте `azure.blob.storage_account` и `azure.blob.shared_key` следующим образом в операторе CREATE EXTERNAL CATALOG:

```
CREATE EXTERNAL CATALOG hive_catalog_azure
PROPERTIES
(
    "type" = "hive",
    "hive.metastore.uris" = "thrift://xx.xx.xx.xx:9083",
    "azure.blob.storage_account" = "<blob_storage_account_name>",
    "azure.blob.shared_key" = "<blob_storage_account_shared_key>"
);
```

Файловая внешняя таблица

Настройте `azure.blob.storage_account`, `azure.blob.shared_key` и путь к файлу (`path`) следующим образом в операторе CREATE EXTERNAL TABLE:

```
CREATE EXTERNAL TABLE external_table_azure
(
    id varchar(65500),
    attributes map<varchar(100), varchar(2000)>
)
ENGINE=FILE
PROPERTIES
(
    "path" =
"wasb[s]://<container>@<storage_account>.blob.core.windows.net/<path>/<file_n
    "format" = "ORC",
    "azure.blob.storage_account" = "<blob_storage_account_name>",
    "azure.blob.shared_key" = "<blob_storage_account_shared_key>"
);
```

Broker Load

Настройте `azure.blob.storage_account`, `azure.blob.shared_key` и путь к файлу (`DATA INFILE`) следующим образом в операторе LOAD LABEL:

```
LOAD LABEL test_db.label000
(
    DATA
INFILE("wasb[s]://<container>@<storage_account>.blob.core.windows.net/<path>/
    INTO TABLE test_ingestion_2
    FORMAT AS "parquet"
)
WITH BROKER
(
    "azure.blob.storage_account" = "<blob_storage_account_name>",
    "azure.blob.shared_key" = "<blob_storage_account_shared_key>"
);
```

SAS Token

Внешний каталог

Настройте `azure.blob.storage_account`, `azure.blob.container` и `azure.blob.sas_token` следующим образом в операторе CREATE EXTERNAL CATALOG:

```
CREATE EXTERNAL CATALOG hive_catalog_azure
PROPERTIES
(
    "type" = "hive",
    "hive.metastore.uris" = "thrift://xx.xx.xx.xx:9083",
    "azure.blob.storage_account" = "<blob_storage_account_name>",
    "azure.blob.container" = "<blob_container_name>",
    "azure.blob.sas_token" = "<blob_storage_account_SAS_token>"
);
```

Файловая внешняя таблица

Настройте `azure.blob.storage_account`, `azure.blob.container`, `azure.blob.sas_token` и путь к файлу (`path`) следующим образом в операторе CREATE EXTERNAL TABLE:

```
CREATE EXTERNAL TABLE external_table_azure
(
    id varchar(65500),
    attributes map<varchar(100), varchar(2000)>
)
ENGINE=FILE
PROPERTIES
(
    "path" =
"wasb[s]://<container>@<storage_account>.blob.core.windows.net/<path>/<file_n
    "format" = "ORC",
    "azure.blob.storage_account" = "<blob_storage_account_name>",
    "azure.blob.container" = "<blob_container_name>",
    "azure.blob.sas_token" = "<blob_storage_account_SAS_token>"
);
```

Broker load

Настройте `azure.blob.storage_account`, `azure.blob.container`, `azure.blob.sas_token` и путь к файлу (`DATA INFILE`) следующим образом в операторе LOAD LABEL:

```
LOAD LABEL test_db.label000
(
    DATA
INFILE("wasb[s]://<container>@<storage_account>.blob.core.windows.net/<path>/
    INTO TABLE target_table
    FORMAT AS "parquet"
)
WITH BROKER
(
    "azure.blob.storage_account" = "<blob_storage_account_name>",
    "azure.blob.container" = "<blob_container_name>",
    "azure.blob.sas_token" = "<blob_storage_account_SAS_token>"
);
```

Data Lake Storage Gen1

Selena поддерживает один из следующих методов аутентификации для доступа к Data Lake Storage Gen1:

- Managed Service Identity
- Service Principal

> **ПРИМЕЧАНИЕ**
>
> При загрузке данных или запросе файлов из Data Lake Storage Gen1 вы должны использовать протокол adl для доступа к вашим данным и записывать путь к файлу как `adl://<data_lake_storage_gen1_name>.azuredatalakestore.net/<path>/<file_name>`.

Managed Service Identity

Внешний каталог

Настройте `azure.adls1.use_managed_service_identity` следующим образом в операторе CREATE EXTERNAL CATALOG:

```
CREATE EXTERNAL CATALOG hive_catalog_azure
PROPERTIES
(
    "type" = "hive",
    "hive.metastore.uris" = "thrift://xx.xx.xx.xx:9083",
    "azure.adls1.use_managed_service_identity" = "true"
);
```

Файловая внешняя таблица

Настройте `azure.adls1.use_managed_service_identity` и путь к файлу (`path`) следующим образом в операторе CREATE EXTERNAL TABLE:

```
CREATE EXTERNAL TABLE external_table_azure
(
    id varchar(65500),
    attributes map<varchar(100), varchar(2000)>
)
ENGINE=FILE
PROPERTIES
(
    "path" =
"adl://<data_lake_storage_gen1_name>.azuredatalakestore.net/<path>/<file_name
    "format" = "ORC",
    "azure.adls1.use_managed_service_identity" = "true"
);
```

Broker Load

Настройте `azure.adls1.use_managed_service_identity` и путь к файлу (`DATA INFILE`) следующим образом в операторе LOAD LABEL:

```
LOAD LABEL test_db.label000
(
    DATA
INFILE("adl://<data_lake_storage_gen1_name>.azuredatalakestore.net/<path>/<fi
    INTO TABLE target_table
    FORMAT AS "parquet"
)
WITH BROKER
(
    "azure.adls1.use_managed_service_identity" = "true"
);
```

Service Principal

Внешний каталог

Настройте `azure.adls1.oauth2_client_id`, `azure.adls1.oauth2_credential` и `azure.adls1.oauth2_endpoint` следующим образом в операторе CREATE EXTERNAL CATALOG:

```
CREATE EXTERNAL CATALOG hive_catalog_azure
PROPERTIES
(
    "type" = "hive",
    "hive.metastore.uris" = "thrift://xx.xx.xx.xx:9083",
    "azure.adls1.oauth2_client_id" = "<application_client_id>",
    "azure.adls1.oauth2_credential" = "<application_client_credential>",
    "azure.adls1.oauth2_endpoint" = "<OAuth_2.0_authorization_endpoint_v2>"
);
```

Файловая внешняя таблица

Настройте `azure.adls1.oauth2_client_id`, `azure.adls1.oauth2_credential`, `azure.adls1.oauth2_endpoint` и путь к файлу (`path`) следующим образом в операторе CREATE EXTERNAL TABLE:

```
CREATE EXTERNAL TABLE external_table_azure
(
    id varchar(65500),
    attributes map<varchar(100), varchar(2000)>
)
ENGINE=FILE
PROPERTIES
(
    "path" =
"adl://<data_lake_storage_gen1_name>.azuredatalakestore.net/<path>/<file_name
    "format" = "ORC",
    "azure.adls1.oauth2_client_id" = "<application_client_id>",
    "azure.adls1.oauth2_credential" = "<application_client_credential>",
    "azure.adls1.oauth2_endpoint" = "<OAuth_2.0_authorization_endpoint_v2>"
);
```

Broker Load

Настройте `azure.adls1.oauth2_client_id`, `azure.adls1.oauth2_credential`, `azure.adls1.oauth2_endpoint` и путь к файлу (`DATA INFILE`) следующим образом в операторе LOAD LABEL:

```
LOAD LABEL test_db.label000
(
    DATA
INFILE("adl://<data_lake_storage_gen1_name>.azuredatalakestore.net/<path>/<fi
    INTO TABLE target_table
    FORMAT AS "parquet"
)
WITH BROKER
(
    "azure.adls1.oauth2_client_id" = "<application_client_id>",
    "azure.adls1.oauth2_credential" = "<application_client_credential>",
    "azure.adls1.oauth2_endpoint" = "<OAuth_2.0_authorization_endpoint_v2>"
);
```

Data Lake Storage Gen2

Selena поддерживает один из следующих методов аутентификации для доступа к Data Lake Storage Gen2:

- Managed Identity
- Shared Key
- Service Principal

> **ПРИМЕЧАНИЕ**
>
> При загрузке данных или запросе файлов из Data Lake Storage Gen2 вы должны использовать протокол abfs или abfss для доступа к вашим данным:
>
> - Если ваша учётная запись хранения разрешает доступ по HTTP, используйте протокол abfs и записывайте путь к файлу как `abfs://<container>@<storage_account>.dfs.core.windows.net/<path>/<file_name>`.
> - Если ваша учётная запись хранения разрешает доступ по HTTPS, используйте протокол abfss и записывайте путь к файлу как `abfss://<container>@<storage_account>.dfs.core.windows.net/<path>/<file_name>`.

Managed Identity

Перед началом работы необходимо выполнить следующие подготовительные действия:

- Отредактируйте виртуальные машины (VM), на которых развёрнут ваш cluster Selena.
- Добавьте управляемые идентификаторы к этим VM.
- Убедитесь, что управляемые идентификаторы связаны с ролью (**Storage Blob Data Reader**), авторизованной для чтения данных в вашей учётной записи хранения.

Внешний каталог

Настройте `azure.adls2.oauth2_use_managed_identity`, `azure.adls2.oauth2_tenant_id` и `azure.adls2.oauth2_client_id` следующим образом в операторе CREATE EXTERNAL CATALOG:

```
CREATE EXTERNAL CATALOG hive_catalog_azure
PROPERTIES
(
    "type" = "hive",
    "hive.metastore.uris" = "thrift://xx.xx.xx.xx:9083",
    "azure.adls2.oauth2_use_managed_identity" = "true",
    "azure.adls2.oauth2_tenant_id" = "<service_principal_tenant_id>",
    "azure.adls2.oauth2_client_id" = "<service_client_id>"
);
```

Файловая внешняя таблица

Настройте `azure.adls2.oauth2_use_managed_identity`, `azure.adls2.oauth2_tenant_id`, `azure.adls2.oauth2_client_id` и путь к файлу (`path`) следующим образом в операторе CREATE EXTERNAL TABLE:

```
CREATE EXTERNAL TABLE external_table_azure
(
    id varchar(65500),
    attributes map<varchar(100), varchar(2000)>
)
ENGINE=FILE
PROPERTIES
(
    "path" =
"abfs[s]://<container>@<storage_account>.dfs.core.windows.net/<path>/<file_na
    "format" = "ORC",
    "azure.adls2.oauth2_use_managed_identity" = "true",
    "azure.adls2.oauth2_tenant_id" = "<service_principal_tenant_id>",
    "azure.adls2.oauth2_client_id" = "<service_client_id>"
);
```

Broker Load

Настройте `azure.adls2.oauth2_use_managed_identity`, `azure.adls2.oauth2_tenant_id`, `azure.adls2.oauth2_client_id` и путь к файлу (`DATA INFILE`) следующим образом в операторе LOAD LABEL:

```
LOAD LABEL test_db.label000
(
    DATA
INFILE("adfs[s]://<container>@<storage_account>.dfs.core.windows.net/<path>/<
    INTO TABLE target_table
    FORMAT AS "parquet"
)
WITH BROKER
(
    "azure.adls2.oauth2_use_managed_identity" = "true",
    "azure.adls2.oauth2_tenant_id" = "<service_principal_tenant_id>",
    "azure.adls2.oauth2_client_id" = "<service_client_id>"
);
```

Shared Key

Внешний каталог

Настройте `azure.adls2.storage_account` и `azure.adls2.shared_key` следующим образом в операторе CREATE EXTERNAL CATALOG:

```
CREATE EXTERNAL CATALOG hive_catalog_azure
PROPERTIES
(
    "type" = "hive",
    "hive.metastore.uris" = "thrift://xx.xx.xx.xx:9083",
    "azure.adls2.storage_account" = "<storage_account_name>",
    "azure.adls2.shared_key" = "<shared_key>"
);
```

Файловая внешняя таблица

Настройте `azure.adls2.storage_account`, `azure.adls2.shared_key` и путь к файлу (`path`) следующим образом в операторе CREATE EXTERNAL TABLE:

```
CREATE EXTERNAL TABLE external_table_azure
(
    id varchar(65500),
    attributes map<varchar(100), varchar(2000)>
)
ENGINE=FILE
PROPERTIES
(
    "path" =
"abfs[s]://<container>@<storage_account>.dfs.core.windows.net/<path>/<file_na
    "format" = "ORC",
    "azure.adls2.storage_account" = "<storage_account_name>",
    "azure.adls2.shared_key" = "<shared_key>"
);
```

Broker Load

Настройте `azure.adls2.storage_account`, `azure.adls2.shared_key` и путь к файлу (`DATA INFILE`) следующим образом в операторе LOAD LABEL:

```
LOAD LABEL test_db.label000
(
    DATA
INFILE("adfs[s]://<container>@<storage_account>.dfs.core.windows.net/<path>/<
    INTO TABLE target_table
    FORMAT AS "parquet"
)
WITH BROKER
(
    "azure.adls2.storage_account" = "<storage_account_name>",
    "azure.adls2.shared_key" = "<shared_key>"
);
```

Service Principal

Перед началом работы необходимо создать субъект-службу, создать назначение роли для назначения роли субъекту-службе, а затем добавить назначение роли к вашей учётной записи хранения. Таким образом, вы можете убедиться, что этот субъект-служба может успешно получить доступ к данным в вашей учётной записи хранения.

Внешний каталог

Настройте `azure.adls2.oauth2_client_id`, `azure.adls2.oauth2_client_secret` и `azure.adls2.oauth2_client_endpoint` следующим образом в операторе CREATE EXTERNAL CATALOG:

```
CREATE EXTERNAL CATALOG hive_catalog_azure
PROPERTIES
(
    "type" = "hive",
    "hive.metastore.uris" = "thrift://xx.xx.xx.xx:9083",
    "azure.adls2.oauth2_client_id" = "<service_client_id>",
    "azure.adls2.oauth2_client_secret" = "
<service_principal_client_secret>",
    "azure.adls2.oauth2_client_endpoint" = "
<service_principal_client_endpoint>"
);
```

Файловая внешняя таблица

Настройте `azure.adls2.oauth2_client_id`, `azure.adls2.oauth2_client_secret`, `azure.adls2.oauth2_client_endpoint` и путь к файлу (`path`) следующим образом в операторе CREATE EXTERNAL TABLE:

```
CREATE EXTERNAL TABLE external_table_azure
(
    id varchar(65500),
    attributes map<varchar(100), varchar(2000)>
)
ENGINE=FILE
PROPERTIES
(
    "path" =
"abfs[s]://<container>@<storage_account>.dfs.core.windows.net/<path>/<file_na
    "format" = "ORC",
    "azure.adls2.oauth2_client_id" = "<service_client_id>",
    "azure.adls2.oauth2_client_secret" = "<service_principal_client_secret>",
    "azure.adls2.oauth2_client_endpoint" = "<service_principal_client_endpoin
);
```

Broker Load

Настройте `azure.adls2.oauth2_client_id`, `azure.adls2.oauth2_client_secret`, `azure.adls2.oauth2_client_endpoint` и путь к файлу (`DATA INFILE`) следующим образом в операторе LOAD LABEL:

```
LOAD LABEL test_db.label000
(
    DATA
INFILE("adfs[s]://<container>@<storage_account>.dfs.core.windows.net/<path>/<
    INTO TABLE target_table
    FORMAT AS "parquet"
)
WITH BROKER
(
    "azure.adls2.oauth2_client_id" = "<service_client_id>",
    "azure.adls2.oauth2_client_secret" = "<service_principal_client_secret>",
    "azure.adls2.oauth2_client_endpoint" = "<service_principal_client_endpoin
);
```

1.1.3 Аутентификация для доступа к Google Cloud Storage

Методы аутентификации

Начиная с версии v1.5.2, Selena поддерживает один из следующих методов аутентификации для доступа к Google Cloud Storage (GCS):

- Аутентификация на основе VM

  Использование учётных данных, прикреплённых к Google Cloud Compute Engine, для аутентификации GCS.

- Аутентификация на основе сервисного аккаунта

  Использование сервисного аккаунта для аутентификации GCS.

- Аутентификация на основе имперсонации

  Позволяет сервисному аккаунту или экземпляру виртуальной машины (VM) имперсонировать другой сервисный аккаунт.

Сценарии

Selena может аутентифицироваться в GCS в следующих сценариях:

- Пакетная загрузка данных из GCS.
- Резервное копирование данных в GCS и восстановление данных из GCS.
- Запрос файлов Parquet и ORC в GCS.
- Запрос таблиц Hive, Iceberg, Hudi и Delta Lake в GCS.

В этом разделе каталог Hive, файловая внешняя таблица и Broker Load используются в качестве примеров, показывающих, как Selena интегрируется с GCS в различных сценариях. Информацию о `StorageCredentialParams` в примерах см. в разделе "Параметры" этого документа.

> **ПРИМЕЧАНИЕ**
>
> Selena поддерживает загрузку данных или прямой запрос файлов из GCS только по протоколу gs. Поэтому при загрузке данных или запросе файлов из GCS вы должны включить `gs` в качестве префикса в пути к файлу.

Внешний каталог

Используйте оператор CREATE EXTERNAL CATALOG для создания каталога Hive с именем `hive_catalog_gcs` следующим образом, чтобы запрашивать файлы из GCS:

```
CREATE EXTERNAL CATALOG hive_catalog_gcs
PROPERTIES
(
    "type" = "hive",
    "hive.metastore.uris" = "thrift://xx.xx.xx.xx:9083",
    StorageCredentialParams
);
```

Файловая внешняя таблица

Используйте оператор CREATE EXTERNAL TABLE для создания файловой внешней таблицы с именем `external_table_gcs` следующим образом, чтобы запрашивать файл данных с именем `test_file_external_tbl` из GCS без какого-либо хранилища метаданных:

```
CREATE EXTERNAL TABLE external_table_gcs
(
    id varchar(65500),
    attributes map<varchar(100), varchar(2000)>
)
ENGINE=FILE
PROPERTIES
(
    "path" = "gs:////test-gcs/test_file_external_tbl",
    "format" = "ORC",
    StorageCredentialParams
);
```

Broker load

Используйте оператор LOAD LABEL для создания задания Broker Load с меткой `test_db.label000`, чтобы пакетно загрузить данные из GCS в таблицу Selena `target_table`:

```
LOAD LABEL test_db.label000
(
    DATA INFILE("gs://bucket_gcs/test_brokerload_ingestion/*")
    INTO TABLE target_table
    FORMAT AS "parquet"
)
WITH BROKER
(
    StorageCredentialParams
);
```

Параметры

`StorageCredentialParams` представляет набор параметров, описывающих, как аутентифицироваться в GCS с использованием различных методов аутентификации.

Аутентификация на основе VM

Если ваш cluster Selena развёрнут на экземпляре VM, размещённом на Google Cloud Platform (GCP), и вы хотите использовать этот экземпляр VM для аутентификации GCS, настройте `StorageCredentialParams` следующим образом:

```
"gcp.gcs.use_compute_engine_service_account" = "true"
```

В следующей таблице описаны параметры, которые необходимо настроить в `StorageCredentialParams`.

Параметр	Значение по умолчанию	Пример значения	Описание
gcp.gcs.use_compute_engine_service_account	false	true	Указывает, следует ли напрямую использовать сервисный аккаунт, привязанный к вашему Compute Engine.

Аутентификация на основе сервисного аккаунта

Если вы напрямую используете сервисный аккаунт для аутентификации GCS, настройте `StorageCredentialParams` следующим образом:

```
"gcp.gcs.service_account_email" = "<google_service_account_email>",
"gcp.gcs.service_account_private_key_id" = "
<google_service_private_key_id>",
"gcp.gcs.service_account_private_key" = "<google_service_private_key>"
```

В следующей таблице описаны параметры, которые необходимо настроить в `StorageCredentialParams`.

Параметр	Значение по умолчанию	Пример значения	Описание
gcp.gcs.service_account_email	""	"`user@hello.iam.gserviceaccount.com`"	Адрес электронной почты в JSON-файле, сгенерированном при создании сервисного аккаунта.
gcp.gcs.service_account_private_key_id	""	"61d257bd8479547cb3e04f0b9b6b9ca07af3b7ea"	ID закрытого ключа в JSON-файле, сгенерированном при создании сервисного аккаунта.
gcp.gcs.service_account_private_key	""	"-----BEGIN PRIVATE KEY----xxxx-----END PRIVATE KEY-----\n"	Закрытый ключ в JSON-файле, сгенерированном при создании сервисного аккаунта.

Аутентификация на основе имперсонации

Имперсонация сервисного аккаунта экземпляром VM

Если ваш cluster Selena развёрнут на экземпляре VM, размещённом на GCP, и вы хотите, чтобы этот экземпляр VM имперсонировал сервисный аккаунт, чтобы Selena унаследовала привилегии от сервисного аккаунта для доступа к GCS, настройте `StorageCredentialParams` следующим образом:

```
"gcp.gcs.use_compute_engine_service_account" = "true",
"gcp.gcs.impersonation_service_account" = "
<assumed_google_service_account_email>"
```

В следующей таблице описаны параметры, которые необходимо настроить в `StorageCredentialParams`.

Параметр	Значение по умолчанию	Пример значения	Описание
gcp.gcs.use_compute_engine_service_account	false	true	Указывает, следует ли напрямую использовать сервисный аккаунт, привязанный к вашему Compute Engine.
gcp.gcs.impersonation_service_account	""	"hello"	Сервисный аккаунт, который вы хотите имперсонировать.

Имперсонация одного сервисного аккаунта другим

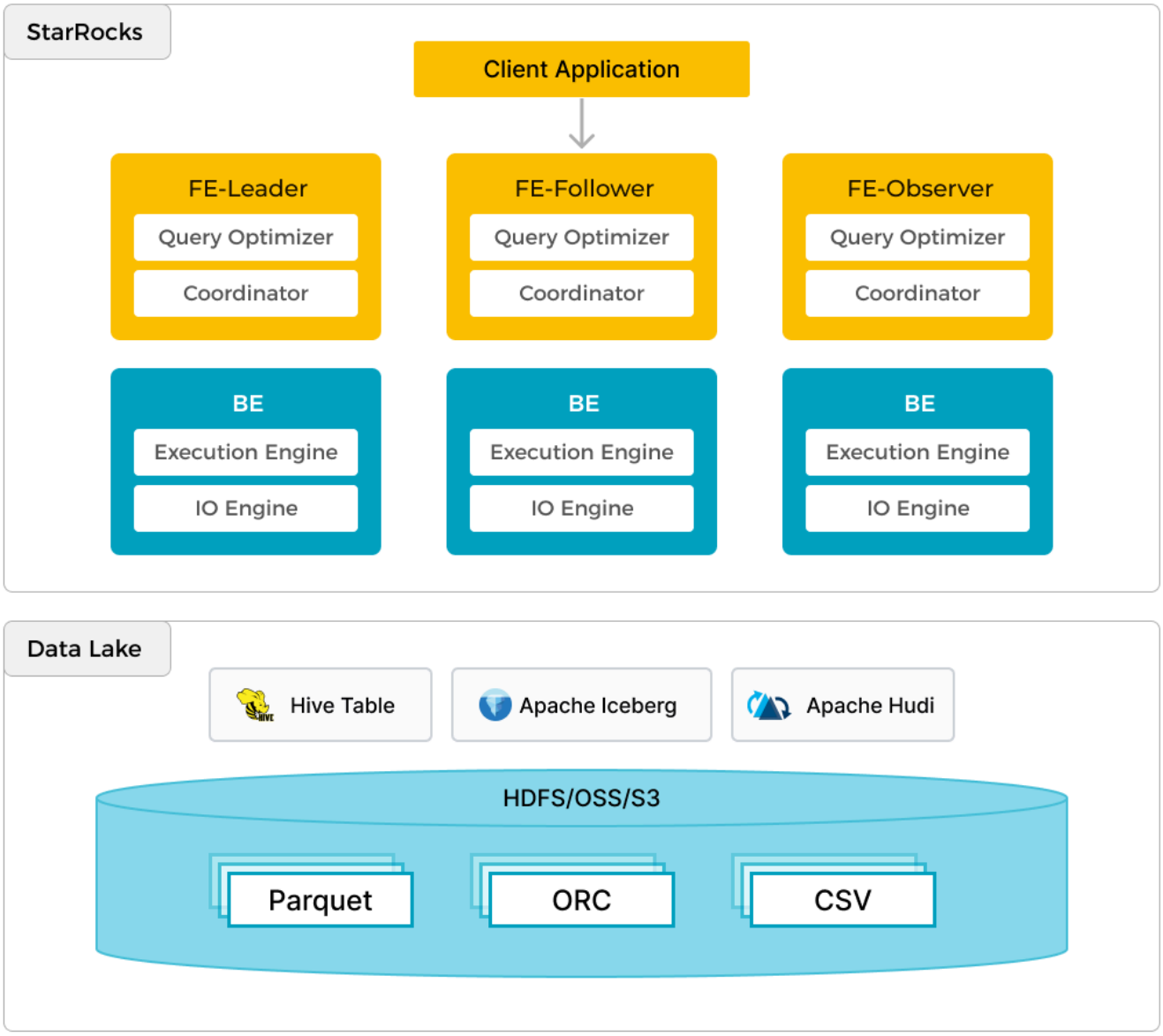
Если вы хотите, чтобы один сервисный аккаунт (временно называемый мета-сервисным аккаунтом) имперсонировал другой сервисный аккаунт (временно называемый сервисным аккаунтом данных) и чтобы Selena унаследовала привилегии от сервисного аккаунта данных для доступа к GCS, настройте `StorageCredentialParams` следующим образом:

```
"gcp.gcs.service_account_email" = "<google_service_account_email>",
"gcp.gcs.service_account_private_key_id" = "
<meta_google_service_account_email>",
"gcp.gcs.service_account_private_key" = "
<meta_google_service_account_email>",
"gcp.gcs.impersonation_service_account" = "
<data_google_service_account_email>"
```

В следующей таблице описаны параметры, которые необходимо настроить в `StorageCredentialParams`.

Параметр	Значение по умолчанию	Пример значения	Описание
gcp.gcs.service_account_email	""	"`user@hello.iam.gserviceaccount.com`"	Адрес электронной почты в JSON-файле, сгенерированном при создании мета-сервисного аккаунта.
gcp.gcs.service_account_private_key_id	""	"61d257bd8479547cb3e04f0b9b6b9ca07af3b7ea"	ID закрытого ключа в JSON-файле, сгенерированном при создании мета-сервисного аккаунта.
gcp.gcs.service_account_private_key	""	"-----BEGIN PRIVATE KEY----xxxx-----END PRIVATE KEY-----\n"	Закрытый ключ в JSON-файле, сгенерированном при создании мета-сервисного аккаунта.
gcp.gcs.impersonation_service_account	""	"hello"	Сервисный аккаунт данных, который вы хотите имперсонировать.

1.2 Data Lakehouse



Помимо эффективной аналитики локальных данных, Selena может работать как вычислительный движок для анализа данных, хранящихся в озёрах данных, таких как Apache Hudi, Apache Iceberg и Delta Lake. Одной из ключевых особенностей Selena является внешний каталог, который служит связующим звеном с внешним metastore. Эта функциональность предоставляет пользователям возможность бесшовно выполнять запросы к внешним источникам данных, устраняя необходимость в миграции данных. Таким образом, пользователи могут анализировать данные из различных систем, таких как HDFS и Amazon S3, в различных форматах файлов, таких как Parquet, ORC, CSV и др.

На приведённом рисунке показан сценарий аналитики озера данных, где Selena отвечает за вычисления и анализ данных, а озеро данных — за хранение, организацию и

обслуживание данных. Озёра данных позволяют пользователям хранить данные в открытых форматах хранения и использовать гибкие схемы для создания отчётов на основе «единого источника истины» для различных сценариев использования BI, AI, ad-hoc и отчётности. Selena в полной мере использует преимущества своего векторизованного движка и CBO, значительно повышая производительность аналитики озёр данных.

1.2.1 Ключевые концепции

- Открытые форматы данных: Поддержка различных типов данных, включая JSON, Parquet и Avro, облегчает хранение и обработку как структурированных, так и неструктурированных данных.
- Управление метаданными: Реализует общий слой метаданных, часто использующий такие форматы как Iceberg table format, для эффективной организации и управления данными.
- Управление и безопасность: Предоставляет надежные встроенные механизмы для безопасности данных, конфиденциальности и соответствия нормативным требованиям, обеспечивая целостность данных и их надежность.

1.2.2 Преимущества архитектуры Data Lakehouse

- Гибкость и масштабируемость: Легко управляет различными типами данных и масштабируется в соответствии с потребностями организации.
- Экономическая эффективность: Предлагает экономичную альтернативу для хранения и обработки данных по сравнению с традиционными методами.
- Улучшенное управление данными: Повышает контроль, управление и целостность данных, обеспечивая надежную и безопасную обработку данных.
- Готовность к AI и аналитике: Идеально подходит для сложных аналитических задач, включая машинное обучение и обработку данных на основе AI.

1.2.3 Подход Selena

Ключевые аспекты для рассмотрения:

- Стандартизация интеграции с catalog или metadata-сервисами
- Эластичная масштабируемость compute-узлов
- Гибкие механизмы кеширования

1.2.4 Catalogs

Selena имеет два типа catalog: внутренние и внешние. Внутренний catalog содержит метаданные для данных, хранящихся в базах данных Selena. Внешние catalogs используются для работы с данными, хранящимися внешне, включая данные, управляемые Hive, Iceberg, Delta Lake и Hudi. Существует много других внешних систем, ссылки находятся в разделе "Дополнительная информация" внизу страницы.

1.2.5 Масштабирование compute-узлов (CN)

Разделение хранения и вычислений снижает сложность масштабирования. Поскольку compute-узлы Selena хранят только локальный cache, узлы могут быть добавлены или удалены в зависимости от нагрузки.

1.2.6 Data cache

Cache на compute-узлах опционален. Если ваши compute-узлы быстро создаются и удаляются в зависимости от быстро меняющихся паттернов нагрузки, или если ваши запросы часто касаются только самых последних данных, кеширование данных может не иметь смысла.

Дополнительная информация находится в документации по Catalog.

1.3 Потоковые данные

1.3.1 Apache Flink

1.3.2 Непрерывная загрузка данных из Apache Flink®

Selena предоставляет самостоятельно разработанный connector под названием Selena Connector for Apache Flink® (сокращённо Flink connector) для загрузки данных в таблицу Selena с использованием Flink. Основной принцип заключается в накоплении данных с последующей загрузкой всех данных за один раз в Selena через STREAM LOAD.

Flink connector поддерживает DataStream API, Table API & SQL и Python API. Он имеет более высокую и стабильную производительность по сравнению с flink-connector-jdbc, предоставляемым Apache Flink®.

> **ВНИМАНИЕ**

> Загрузка данных в таблицы Selena с помощью Flink connector требует привилегий SELECT и INSERT на целевую таблицу Selena. Если у вас нет этих привилегий, следуйте инструкциям, представленным в GRANT, чтобы предоставить эти привилегии пользователю, которого вы используете для подключения к вашему cluster Selena.

Требования к версиям

Connector	Flink	Selena	Java	Scala
1.2.11	1.15,1.16,1.17,1.18,1.19,1.20	2.1 и позже	8	2.11,2.12
1.2.10	1.15,1.16,1.17,1.18,1.19	2.1 и позже	8	2.11,2.12
1.2.9	1.15,1.16,1.17,1.18	2.1 и позже	8	2.11,2.12
1.2.8	1.13,1.14,1.15,1.16,1.17	2.1 и позже	8	2.11,2.12
1.2.7	1.11,1.12,1.13,1.14,1.15	2.1 и позже	8	2.11,2.12

Получение Flink connector

Вы можете получить JAR-файл Flink connector следующими способами:

- Напрямую скачать скомпилированный JAR-файл Flink connector.
- Добавить Flink connector в качестве зависимости в ваш проект Maven, а затем скачать JAR-файл.
- Скомпилировать исходный код Flink connector в JAR-файл самостоятельно.

Формат наименования JAR-файла Flink connector следующий:

- Начиная с Flink 1.15, это `flink-connector-selena-${connector_version}_flink-${flink_version}.jar`. Например, если вы установили Flink 1.15 и хотите использовать Flink connector 1.2.7, вы можете использовать `flink-connector-selena-1.2.7_flink-1.15.jar`.

- До Flink 1.15 это `flink-connector-selena-${connector_version}_flink-${flink_version}_${scala_version}.jar`. Например, если вы установили Flink 1.14 и Scala 2.12 в вашей среде, и хотите использовать Flink connector 1.2.7, вы можете использовать `flink-connector-selena-1.2.7_flink-1.14_2.12.jar`.

> **ВНИМАНИЕ**

> В общем случае последняя версия Flink connector поддерживает совместимость только с тремя последними версиями Flink.

Скачивание скомпилированного Jar-файла

Напрямую скачайте соответствующую версию JAR-файла Flink connector из Maven Central Repository.

Maven Dependency

В файле `pom.xml` вашего проекта Maven добавьте Flink connector в качестве зависимости в соответствии со следующим форматом. Замените `flink_version`, `scala_version` и `connector_version` на соответствующие версии.

- В Flink 1.15 и позже

```
<dependency>
    <groupId>com.selena</groupId>
    <artifactId>flink-connector-selena</artifactId>
    <version>${connector_version}_flink-${flink_version}</version>
</dependency>
```

- В версиях до Flink 1.15

```
<dependency>
    <groupId>com.selena</groupId>
    <artifactId>flink-connector-selena</artifactId>

<version>${connector_version}_flink-${flink_version}_${scala_version}
</version>
</dependency>
```

Компиляция самостоятельно

1. Скачайте исходный код Flink connector.

2. Выполните следующую команду для компиляции исходного кода Flink connector в JAR-файл. Обратите внимание, что `flink_version` заменяется на соответствующую версию Flink.

```
sh build.sh <flink_version>
```

Например, если версия Flink в вашей среде 1.15, вам нужно выполнить следующую команду:

```
sh build.sh 1.15
```

3. Перейдите в каталог `target/`, чтобы найти JAR-файл Flink connector, такой как `flink-connector-selena-1.2.7_flink-1.15-SNAPSHOT.jar`, сгенерированный после компиляции.

> **ПРИМЕЧАНИЕ**
>
> Имя Flink connector, который не выпущен официально, содержит суффикс `SNAPSHOT`.

Параметры

connector

Обязательный: Да
Значение по умолчанию: NONE
Описание: Connector, который вы хотите использовать. Значение должно быть "selena".

jdbc-url

Обязательный: Да
Значение по умолчанию: NONE
Описание: Адрес, который используется для подключения к серверу MySQL FE. Вы можете указать несколько адресов, которые должны быть разделены запятой (,). Формат: `jdbc:mysql://<fe_host1>:<fe_query_port1>,<fe_host2>:<fe_query_port2>,<fe_host3>:<fe_query_port3>`.

load-url

Обязательный: Да
Значение по умолчанию: NONE
Описание: Адрес, который используется для подключения к HTTP-серверу FE. Вы можете указать несколько адресов, которые должны быть разделены точкой с запятой (;). Формат: `<fe_host1>:<fe_http_port1>;<fe_host2>:<fe_http_port2>`.

database-name

Обязательный: Да
Значение по умолчанию: NONE
Описание: Имя базы данных Selena, в которую вы хотите загрузить данные.

table-name

Обязательный: Да
Значение по умолчанию: NONE
Описание: Имя таблицы, которую вы хотите использовать для загрузки данных в Selena.

username

Обязательный: Да
Значение по умолчанию: NONE
Описание: Имя пользователя учетной записи, которую вы хотите использовать для загрузки данных в Selena. Учетная запись должна иметь привилегии SELECT и INSERT на целевую таблицу Selena.

password

Обязательный: Да
Значение по умолчанию: NONE
Описание: Пароль предыдущей учетной записи.

sink.version

Обязательный: Нет
Значение по умолчанию: AUTO
Описание: Интерфейс, используемый для загрузки данных. Этот параметр поддерживается начиная с версии Flink connector 1.2.4.

- `V1`: Использовать интерфейс Stream Load для загрузки данных. Connectors до 1.2.4 поддерживают только этот режим.
- `V2`: Использовать интерфейс Stream Load transaction для загрузки данных. Требуется Selena версии не ниже 2.4. Рекомендуется `V2`, так как он оптимизирует использование памяти и обеспечивает более стабильную реализацию exactly-once.
- `AUTO`: Если версия Selena поддерживает транзакционный Stream Load, автоматически выберет `V2`, иначе выберет `V1`

sink.label-prefix

Обязательный: Нет
Значение по умолчанию: NONE
Описание: Префикс метки, используемый Stream Load. Рекомендуется настроить его, если вы используете exactly-once с connector 1.2.8 и позже. См. примечания по использованию exactly-once.

sink.semantic

Обязательный: Нет
Значение по умолчанию: at-least-once
Описание: Семантика, гарантируемая sink. Допустимые значения: **at-least-once** и **exactly-once**.

sink.buffer-flush.max-bytes

Обязательный: Нет
Значение по умолчанию: 94371840(90M)
Описание: Максимальный размер данных, которые могут быть накоплены в памяти перед отправкой в Selena за один раз. Максимальное значение варьируется от 64 МБ до 10 ГБ. Установка этого параметра на большее значение может улучшить производительность загрузки, но может увеличить задержку загрузки. Этот параметр вступает в силу только тогда, когда `sink.semantic` установлен на `at-least-once`. Если `sink.semantic` установлен на `exactly-once`, данные в памяти сбрасываются при запуске checkpoint Flink. В этом случае этот параметр не действует.

sink.buffer-flush.max-rows

Обязательный: Нет
Значение по умолчанию: 500000
Описание: Максимальное количество строк, которые могут быть накоплены в памяти перед отправкой в Selena за один раз. Этот параметр доступен только тогда, когда `sink.version` равен `V1`, а `sink.semantic` равен `at-least-once`. Допустимые значения: от 64000 до 5000000.

sink.buffer-flush.interval-ms

Обязательный: Нет
Значение по умолчанию: 300000

Описание: Интервал сброса данных. Этот параметр доступен только тогда, когда `sink.semantic` равен `at-least-once`. Допустимые значения: от 1000 до 3600000. Единица измерения: мс.

sink.max-retries

Обязательный: Нет
Значение по умолчанию: 3
Описание: Количество раз, когда система повторяет попытку выполнить задачу Stream Load. Этот параметр доступен только тогда, когда вы устанавливаете `sink.version` в `V1`. Допустимые значения: от 0 до 10.

sink.connect.timeout-ms

Обязательный: Нет
Значение по умолчанию: 30000
Описание: Тайм-аут для установки HTTP-соединения. Допустимые значения: от 100 до 60000. Единица измерения: мс. До Flink connector v1.2.9 значение по умолчанию было `1000`.

sink.socket.timeout-ms

Обязательный: Нет
Значение по умолчанию: -1
Описание: Поддерживается с версии 1.2.10. Продолжительность времени, в течение которого HTTP-клиент ожидает данных. Единица измерения: мс. Значение по умолчанию `-1` означает, что тайм-аута нет.

sink.sanitize-error-log

Обязательный: Нет
Значение по умолчанию: false
Описание: Поддерживается с версии 1.2.12. Следует ли очищать конфиденциальные данные в журнале ошибок для производственной безопасности. Когда этот элемент установлен в `true`, конфиденциальные данные строк и значения столбцов в журналах ошибок Stream Load редактируются как в журналах connector, так и в журналах SDK. Значение по умолчанию - `false` для обратной совместимости.

sink.wait-for-continue.timeout-ms

Обязательный: Нет
Значение по умолчанию: 10000
Описание: Поддерживается с версии 1.2.7. Тайм-аут ожидания ответа HTTP 100-continue от FE. Допустимые значения: от `3000` до `60000`. Единица измерения: мс

sink.ignore.update-before

Обязательный: Нет
Значение по умолчанию: true
Описание: Поддерживается с версии 1.2.8. Следует ли игнорировать записи `UPDATE_BEFORE` из Flink при загрузке данных в таблицы Primary Key. Если этот параметр установлен в false, запись рассматривается как операция удаления в таблице Selena.

sink.parallelism

Обязательный: Нет
Значение по умолчанию: NONE
Описание: Параллелизм загрузки. Доступно только для Flink SQL. Если этот параметр не указан, планировщик Flink определяет параллелизм. **В сценарии с мультипараллелизмом пользователи должны гарантировать, что данные записываются в правильном порядке.**

sink.properties.*

Обязательный: Нет
Значение по умолчанию: NONE
Описание: Параметры, которые используются для управления поведением Stream Load. Например, параметр `sink.properties.format` указывает формат, используемый для Stream Load, такой как CSV или JSON. Список поддерживаемых параметров и их описания см. в STREAM LOAD.

sink.properties.format

Обязательный: Нет
Значение по умолчанию: csv
Описание: Формат, используемый для Stream Load. Flink connector преобразует каждый пакет данных в формат перед их отправкой в Selena. Допустимые значения: `csv` и `json`.

sink.properties.column_separator

Обязательный: Нет
Значение по умолчанию: \t
Описание: Разделитель столбцов для данных в формате CSV.

sink.properties.row_delimiter

Обязательный: Нет
Значение по умолчанию: \n
Описание: Разделитель строк для данных в формате CSV.

sink.properties.max_filter_ratio

Обязательный: Нет
Значение по умолчанию: 0
Описание: Максимальная толерантность к ошибкам Stream Load. Это максимальный процент записей данных, которые могут быть отфильтрованы из-за недостаточного качества данных. Допустимые значения: от `0` до `1`. Значение по умолчанию: `0`. Подробности см. в Stream Load.

sink.properties.partial_update

Обязательный: НЕТ
Значение по умолчанию: `FALSE`
Описание: Следует ли использовать частичные обновления. Допустимые значения: `TRUE` и `FALSE`. Значение по умолчанию: `FALSE`, указывающее на отключение этой функции.

sink.properties.partial_update_mode

Обязательный: НЕТ
Значение по умолчанию: `row`
Описание: Указывает режим частичных обновлений. Допустимые значения: `row` и `column`.

- Значение `row` (по умолчанию) означает частичные обновления в режиме строк, что более подходит для обновлений в реальном времени с большим количеством столбцов и небольшими пакетами.
- Значение `column` означает частичные обновления в режиме столбцов, что более подходит для пакетных обновлений с небольшим количеством столбцов и большим

количеством строк. В таких сценариях включение режима столбцов обеспечивает более быструю скорость обновления. Например, в таблице с 100 столбцами, если обновляются только 10 столбцов (10% от общего числа) для всех строк, скорость обновления в режиме столбцов в 10 раз быстрее.

sink.properties.strict_mode

Обязательный: Нет
Значение по умолчанию: false
Описание: Указывает, следует ли включать строгий режим для Stream Load. Это влияет на поведение загрузки при наличии неквалифицированных строк, таких как несогласованные значения столбцов. Допустимые значения: `true` и `false`. Значение по умолчанию: `false`. Подробности см. в Stream Load.

sink.properties.compression

Обязательный: Нет
Значение по умолчанию: NONE
Описание: Алгоритм сжатия, используемый для Stream Load. Допустимые значения: `lz4_frame`. Сжатие для формата JSON требует Flink connector 1.2.10+ и Selena v1.5.2+. Сжатие для формата CSV требует только Flink connector 1.2.11+.

sink.properties.prepared_timeout

Обязательный: Нет
Значение по умолчанию: NONE
Описание: Поддерживается с версии 1.2.12 и действует только тогда, когда `sink.version` установлен в `V2`. Требуется Selena 3.5.4 или позже. Устанавливает тайм-аут в секундах для фазы Transaction Stream Load от `PREPARED` до `COMMITTED`. Обычно требуется только для exactly-once; at-least-once обычно не требует установки этого (connector по умолчанию использует 300 с). Если не установлено в exactly-once, применяется конфигурация Selena FE `prepared_transaction_default_timeout_second` (по умолчанию 86400 с). См. управление тайм-аутом транзакций Selena.

Сопоставление типов данных между Flink и Selena

Тип данных Flink	Тип данных Selena
BOOLEAN	BOOLEAN
TINYINT	TINYINT

Тип данных Flink	Тип данных Selena
SMALLINT	SMALLINT
INTEGER	INTEGER
BIGINT	BIGINT
FLOAT	FLOAT
DOUBLE	DOUBLE
DECIMAL	DECIMAL
BINARY	INT
CHAR	STRING
VARCHAR	STRING
STRING	STRING
DATE	DATE
TIMESTAMP_WITHOUT_TIME_ZONE(N)	DATETIME
TIMESTAMP_WITH_LOCAL_TIME_ZONE(N)	DATETIME
ARRAY<T>	ARRAY<T>
MAP<KT,VT>	JSON STRING
ROW<arg T...>	JSON STRING

Примечания по использованию

Exactly Once

- Если вы хотите, чтобы sink гарантировал семантику exactly-once, мы рекомендуем вам обновить Selena до версии 2.5 или позже, а Flink connector до версии 1.2.4 или позже
    - Начиная с Flink connector 1.2.4, exactly-once был переработан на основе интерфейса транзакций Stream Load, предоставленного Selena с версии 2.4. По сравнению с предыдущей реализацией на основе нетранзакционного интерфейса Stream Load, новая реализация снижает использование памяти и накладные расходы на checkpoint, тем самым повышая производительность в реальном времени и стабильность загрузки.
    - Если версия Selena ранее 2.4 или версия Flink connector ранее 1.2.4, sink автоматически выберет реализацию на основе нетранзакционного интерфейса Stream Load.
- Конфигурации для гарантии exactly-once

- Значение `sink.semantic` должно быть `exactly-once`.
- Если версия Flink connector 1.2.8 и позже, рекомендуется указать значение `sink.label-prefix`. Обратите внимание, что префикс метки должен быть уникальным среди всех типов загрузки в Selena, таких как задачи Flink, Routine Load и Broker Load.
  - Если префикс метки указан, Flink connector будет использовать префикс метки для очистки оставшихся транзакций, которые могут быть сгенерированы в некоторых сценариях сбоя Flink, таких как сбой задачи Flink, когда checkpoint все еще выполняется. Эти оставшиеся транзакции обычно находятся в статусе `PREPARED`, если вы используете `SHOW PROC '/transactions/<db_id>/running';` для их просмотра в Selena. Когда задача Flink восстанавливается из checkpoint, Flink connector найдет эти оставшиеся транзакции в соответствии с префиксом метки и некоторой информацией в checkpoint и прервет их. Flink connector не может прервать их при выходе задачи Flink из-за механизма двухфазной фиксации для реализации exactly-once. Когда задача Flink завершается, Flink connector еще не получил уведомление от координатора checkpoint Flink о том, должны ли транзакции быть включены в успешный checkpoint, и это может привести к потере данных, если эти транзакции будут прерваны в любом случае. Вы можете получить обзор о том, как достичь сквозного exactly-once в Flink в этом blogpost.
  - Если префикс метки не указан, оставшиеся транзакции будут очищены Selena только после их истечения. Однако количество выполняющихся транзакций может достичь ограничения Selena `max_running_txn_num_per_db`, если задачи Flink часто терпят неудачу до истечения времени ожидания транзакций. Вы можете установить меньший тайм-аут для транзакций `PREPARED` чтобы они истекали быстрее, когда префикс метки не указан. См. ниже о том, как установить подготовленный тайм-аут.

- Если вы уверены, что задача Flink в конечном итоге восстановится из checkpoint или savepoint после длительного простоя из-за остановки или непрерывного отказа, пожалуйста, соответствующим образом настройте следующие конфигурации Selena, чтобы избежать потери данных.

- Настройте тайм-аут транзакций `PREPARED`. См. ниже о том, как установить тайм-аут.

  Тайм-аут должен быть больше времени простоя задачи Flink. В противном случае оставшиеся транзакции, которые включены в успешный checkpoint, могут быть прерваны из-за тайм-аута до перезапуска задачи Flink, что приводит к потере данных.

  Обратите внимание, что когда вы устанавливаете большее значение для этой конфигурации, лучше указать значение `sink.label-prefix`, чтобы оставшиеся транзакции могли быть очищены в соответствии с префиксом метки и некоторой информацией в checkpoint, вместо того чтобы из-за тайм-аута (что может привести к потере данных).

- `label_keep_max_second` и `label_keep_max_num`: конфигурации Selena FE, значения по умолчанию - `259200` и `1000` соответственно. Подробности см. в конфигурациях FE. Значение `label_keep_max_second` должно быть больше времени простоя задачи Flink. В противном случае Flink connector не сможет проверить состояние транзакций в Selena, используя метки транзакций, сохраненные в savepoint или checkpoint Flink, и выяснить, зафиксированы ли эти транзакции или нет, что в конечном итоге может привести к потере данных.

- Как установить тайм-аут для транзакций PREPARED

  - Для Connector 1.2.12+ и Selena 3.5.4+ вы можете установить тайм-аут, настроив параметр connector `sink.properties.prepared_timeout`. По умолчанию значение не установлено, и оно возвращается к глобальной конфигурации Selena FE `prepared_transaction_default_timeout_second` (значение по умолчанию - `86400`).

  - Для других версий Connector или Selena вы можете установить тайм-аут, настроив глобальную конфигурацию Selena FE `prepared_transaction_default_timeout_second` (значение по умолчанию - `86400`).

Политика сброса

Flink connector будет буферизовать данные в памяти и сбрасывать их пакетами в Selena через Stream Load. Способ запуска сброса отличается между at-least-once и exactly-once.

Для at-least-once сброс будет запущен, когда выполняется любое из следующих условий:

- количество байтов буферизованных строк достигает предела `sink.buffer-flush.max-bytes`
- количество буферизованных строк достигает предела `sink.buffer-flush.max-rows`. (Действительно только для версии sink V1)
- время, прошедшее с момента последнего сброса, достигает предела `sink.buffer-flush.interval-ms`
- запущен checkpoint

Для exactly-once сброс происходит только при запуске checkpoint.

Мониторинг метрик загрузки

Flink connector предоставляет следующие метрики для мониторинга загрузки.

Метрика	Тип	Описание
totalFlushBytes	counter	успешно сброшенные байты.
totalFlushRows	counter	количество успешно сброшенных строк.
totalFlushSucceededTimes	counter	количество раз, когда данные были успешно сброшены.
totalFlushFailedTimes	counter	количество раз, когда сброс данных не удался.
totalFilteredRows	counter	количество отфильтрованных строк, которое также включено в totalFlushRows.

Примеры

Следующие примеры показывают, как использовать Flink connector для загрузки данных в таблицу Selena с помощью Flink SQL или Flink DataStream.

Подготовка

Создание таблицы Selena

Создайте базу данных `test` и создайте таблицу Primary Key `score_board`.

```
CREATE DATABASE `test`;

CREATE TABLE `test`.`score_board`
(
    `id` int(11) NOT NULL COMMENT "",
    `name` varchar(65533) NULL DEFAULT "" COMMENT "",
    `score` int(11) NOT NULL DEFAULT "0" COMMENT ""
)
ENGINE=OLAP
PRIMARY KEY(`id`)
COMMENT "OLAP"
DISTRIBUTED BY HASH(`id`);
```

Настройка среды Flink

- Скачайте бинарный файл Flink Flink 1.15.2 и распакуйте его в каталог `flink-1.15.2`.
- Скачайте Flink connector 1.2.7 и поместите его в каталог `flink-1.15.2/lib`.
- Выполните следующие команды для запуска cluster Flink:

  ```
  cd flink-1.15.2
  ./bin/start-cluster.sh
  ```

Конфигурация сети

Убедитесь, что машина, на которой находится Flink, может получить доступ к узлам FE cluster Selena через `http_port` (по умолчанию: `8030`) и `query_port` (по умолчанию: `9030`), а к узлам BE через `be_http_port` (по умолчанию: `8040`).

Запуск с Flink SQL

- Выполните следующую команду для запуска клиента Flink SQL.

  ```
  ./bin/sql-client.sh
  ```

- Создайте таблицу Flink `score_board` и вставьте значения в таблицу через Flink SQL Client. Обратите внимание, что вы должны определить первичный ключ в Flink DDL, если хотите загрузить данные в таблицу Primary Key Selena. Это необязательно для других типов таблиц Selena.

```sql
CREATE TABLE `score_board` (
    `id` INT,
    `name` STRING,
    `score` INT,
    PRIMARY KEY (id) NOT ENFORCED
) WITH (
    'connector' = 'selena',
    'jdbc-url' = 'jdbc:mysql://127.0.0.1:9030',
    'load-url' = '127.0.0.1:8030',
    'database-name' = 'test',

    'table-name' = 'score_board',
    'username' = 'root',
    'password' = ''
);

INSERT INTO `score_board` VALUES (1, 'selena', 100), (2, 'flink', 100);
```

Запуск с Flink DataStream

Существует несколько способов реализации задачи Flink DataStream в зависимости от типа входных записей, таких как CSV Java `String`, JSON Java `String` или пользовательский объект Java.

- Входные записи представляют собой `String` в формате CSV. Полный пример см. в LoadCsvRecords.

```
/**
 * Генерация записей в формате CSV. Каждая запись имеет три значения,
разделенных "\t".
 * Эти значения будут загружены в столбцы `id`, `name` и `score` в
таблице Selena.
 */
String[] records = new String[]{
        "1\tselena-csv\t100",
        "2\tflink-csv\t100"
};
DataStream<String> source = env.fromElements(records);

/**
 * Настройте connector с необходимыми свойствами.
 * Вам также необходимо добавить свойства "sink.properties.format" и
"sink.properties.column_separator"
 * чтобы сообщить connector, что входные записи в формате CSV, а
разделитель столбцов - "\t".
 * Вы также можете использовать другие разделители столбцов в записях
формата CSV,
 * но не забудьте соответственно изменить
"sink.properties.column_separator".
 */
SelenaSinkOptions options = SelenaSinkOptions.builder()
        .withProperty("jdbc-url", jdbcUrl)
        .withProperty("load-url", loadUrl)
        .withProperty("database-name", "test")
        .withProperty("table-name", "score_board")
        .withProperty("username", "root")
        .withProperty("password", "")
        .withProperty("sink.properties.format", "csv")
        .withProperty("sink.properties.column_separator", "\t")
        .build();
// Создайте sink с параметрами.
SinkFunction<String> selenaink = SelenaSink.sink(options);
source.addSink(selenaink);
```

- Входные записи представляют собой `String` в формате JSON. Полный пример см. в LoadJsonRecords.

```
/**
 * Генерация записей в формате JSON.
 * Каждая запись имеет три пары ключ-значение, соответствующие столбцам
`id`, `name` и `score` в таблице Selena.
 */
String[] records = new String[]{
        "{\"id\":1, \"name\":\"selena-json\", \"score\":100}",
        "{\"id\":2, \"name\":\"flink-json\", \"score\":100}",
};
DataStream<String> source = env.fromElements(records);

/**
 * Настройте connector с необходимыми свойствами.
 * Вам также необходимо добавить свойства "sink.properties.format" и
"sink.properties.strip_outer_array"
 * чтобы сообщить connector, что входные записи в формате JSON и нужно
удалить внешнюю структуру массива.
 */
SelenaSinkOptions options = SelenaSinkOptions.builder()
        .withProperty("jdbc-url", jdbcUrl)
        .withProperty("load-url", loadUrl)
        .withProperty("database-name", "test")
        .withProperty("table-name", "score_board")
        .withProperty("username", "root")
        .withProperty("password", "")
        .withProperty("sink.properties.format", "json")
        .withProperty("sink.properties.strip_outer_array", "true")
        .build();
// Создайте sink с параметрами.
SinkFunction<String> selenaink = SelenaSink.sink(options);
source.addSink(selenaink);
```

- Входные записи представляют собой пользовательские объекты Java. Полный пример см. в LoadCustomJavaRecords.
    - В этом примере входная запись представляет собой простой POJO `RowData`.

```
public static class RowData {
        public int id;
        public String name;
        public int score;

        public RowData() {}

        public RowData(int id, String name, int score) {
            this.id = id;
            this.name = name;
            this.score = score;
        }
  }
```

- Основная программа выглядит следующим образом:

```
// Генерация записей, которые используют RowData в качестве
контейнера.
RowData[] records = new RowData[]{
        new RowData(1, "selena-rowdata", 100),
        new RowData(2, "flink-rowdata", 100),
    };
DataStream<RowData> source = env.fromElements(records);

// Настройте connector с необходимыми свойствами.
SelenaSinkOptions options = SelenaSinkOptions.builder()
        .withProperty("jdbc-url", jdbcUrl)
        .withProperty("load-url", loadUrl)
        .withProperty("database-name", "test")
        .withProperty("table-name", "score_board")
        .withProperty("username", "root")
        .withProperty("password", "")
        .build();

/**
 * Flink connector будет использовать массив объектов Java
(Object[]) для представления строки, которая будет загружена в
таблицу Selena,
 * и каждый элемент является значением для столбца.
 * Вам необходимо определить схему Object[], которая соответствует
схеме таблицы Selena.
 */
TableSchema schema = TableSchema.builder()
        .field("id", DataTypes.INT().notNull())
        .field("name", DataTypes.STRING())
        .field("score", DataTypes.INT())
        // Когда таблица Selena является таблицей Primary Key, вы
должны указать notNull(), например, DataTypes.INT().notNull(), для
первичного ключа `id`.
        .primaryKey("id")
        .build();
// Преобразуйте RowData в Object[] в соответствии со схемой.
RowDataTransformer transformer = new RowDataTransformer();
// Создайте sink со схемой, параметрами и трансформером.
SinkFunction<RowData> selenaink = SelenaSink.sink(schema, options,
transformer);
source.addSink(selenaink);
```

- `RowDataTransformer` в основной программе определен следующим образом:

```
private static class RowDataTransformer implements
SelenaSinkRowBuilder<RowData> {

    /**
     * Установите каждый элемент массива объектов в соответствии с
входными RowData.
     * Схема массива соответствует схеме таблицы Selena.
     */
    @Override
    public void accept(Object[] internalRow, RowData rowData) {
        internalRow[0] = rowData.id;
        internalRow[1] = rowData.name;
        internalRow[2] = rowData.score;
        // Когда таблица Selena является таблицей Primary Key, вам
нужно установить последний элемент, чтобы указать, является ли
загрузка данных операцией UPSERT или DELETE.
        internalRow[internalRow.length - 1] =
SelenaSinkOP.UPSERT.ordinal();
    }
}
```

Синхронизация данных с Flink CDC 3.0 (с поддержкой изменения схемы)

Фреймворк Flink CDC 3.0 можно использовать для легкого создания потокового конвейера ELT из источников CDC (таких как MySQL и Kafka) в Selena. Конвейер может синхронизировать всю базу данных, объединенные таблицы шардинга и изменения схемы из источников в Selena.

Начиная с v1.2.9, Flink connector для Selena интегрирован в этот фреймворк как Selena Pipeline Connector. Selena Pipeline Connector поддерживает:

- Автоматическое создание баз данных и таблиц
- Синхронизацию изменений схемы
- Полную и инкрементальную синхронизацию данных

Для быстрого старта см. Streaming ELT from MySQL to Selena using Flink CDC 3.0 with Selena Pipeline Connector.

Рекомендуется использовать Selena v1.5.2 и более поздние версии для включения fast_schema_evolution. Это улучшит скорость добавления или удаления столбцов и снизит

использование ресурсов.

Лучшие практики

Загрузка данных в таблицу Primary Key

В этом разделе показано, как загружать данные в таблицу Primary Key Selena для достижения частичных обновлений и условных обновлений. Вы можете увидеть Изменение данных через загрузку для введения этих функций. Эти примеры используют Flink SQL.

Подготовка

Создайте базу данных `test` и создайте таблицу Primary Key `score_board` в Selena.

```
CREATE DATABASE `test`;

CREATE TABLE `test`.`score_board`
(
    `id` int(11) NOT NULL COMMENT "",
    `name` varchar(65533) NULL DEFAULT "" COMMENT "",
    `score` int(11) NOT NULL DEFAULT "0" COMMENT ""
)
ENGINE=OLAP
PRIMARY KEY(`id`)
COMMENT "OLAP"
DISTRIBUTED BY HASH(`id`);
```

Частичное обновление

Этот пример покажет, как загружать данные только в столбцы `id` и `name`.

1. Вставьте две строки данных в таблицу Selena `score_board` в клиенте MySQL.

```
mysql> INSERT INTO `score_board` VALUES (1, 'selena', 100), (2,
'flink', 100);

mysql> select * from score_board;
+------+-----------+-------+
| id   | name      | score |
+------+-----------+-------+
|    1 | selena |   100 |
|    2 | flink     |   100 |
+------+-----------+-------+
2 rows in set (0.02 sec)
```

2. Создайте таблицу Flink `score_board` в клиенте Flink SQL.

   - Определите DDL, который включает только столбцы `id` и `name`.
   - Установите опцию `sink.properties.partial_update` в `true`, что сообщает Flink connector о необходимости выполнения частичных обновлений.
   - Если версия Flink connector `<=` 1.2.7, вам также необходимо установить опцию `sink.properties.columns` в `id,name,__op`, чтобы сообщить Flink connector, какие столбцы нужно обновить. Обратите внимание, что вам нужно добавить поле `__op` в конце. Поле `__op` указывает, что загрузка данных является операцией UPSERT или DELETE, и его значения устанавливаются connector автоматически.

```
CREATE TABLE `score_board` (
    `id` INT,
    `name` STRING,
    PRIMARY KEY (id) NOT ENFORCED
) WITH (
    'connector' = 'selena',
    'jdbc-url' = 'jdbc:mysql://127.0.0.1:9030',
    'load-url' = '127.0.0.1:8030',
    'database-name' = 'test',
    'table-name' = 'score_board',
    'username' = 'root',
    'password' = '',
    'sink.properties.partial_update' = 'true',
    -- только для версии Flink connector <= 1.2.7
    'sink.properties.columns' = 'id,name,__op'
);
```

3. Вставьте две строки данных в таблицу Flink. Первичные ключи строк данных такие же, как у строк в таблице Selena, но значения в столбце `name` изменены.

```
INSERT INTO `score_board` VALUES (1, 'selena-update'), (2, 'flink-
update');
```

4. Запросите таблицу Selena в клиенте MySQL.

```
mysql> select * from score_board;
+------+------------------+-------+
| id   | name             | score |
+------+------------------+-------+
|    1 | selena-update |   100 |
|    2 | flink-update     |   100 |
+------+------------------+-------+
2 rows in set (0.02 sec)
```

Вы можете видеть, что изменились только значения для `name`, а значения для `score` не изменились.

Условное обновление

Этот пример покажет, как выполнить условное обновление в соответствии со значением столбца `score`. Обновление для `id` вступает в силу только тогда, когда новое значение для `score` больше или равно старому значению.

1. Вставьте две строки данных в таблицу Selena в клиенте MySQL.

```
mysql> INSERT INTO `score_board` VALUES (1, 'selena', 100), (2,
'flink', 100);

mysql> select * from score_board;
+------+-----------+-------+
| id   | name      | score |
+------+-----------+-------+
|    1 | selena |   100 |
|    2 | flink     |   100 |
+------+-----------+-------+
2 rows in set (0.02 sec)
```

2. Создайте таблицу Flink `score_board` следующими способами:

   - Определите DDL, включающий все столбцы.
   - Установите опцию `sink.properties.merge_condition` в `score`, чтобы сообщить connector использовать столбец `score` в качестве условия.
   - Установите опцию `sink.version` в `V1`, что сообщает connector использовать Stream Load.

```
CREATE TABLE `score_board` (
    `id` INT,
    `name` STRING,
    `score` INT,
    PRIMARY KEY (id) NOT ENFORCED
) WITH (
    'connector' = 'selena',
    'jdbc-url' = 'jdbc:mysql://127.0.0.1:9030',
    'load-url' = '127.0.0.1:8030',
    'database-name' = 'test',
    'table-name' = 'score_board',
    'username' = 'root',
    'password' = '',
    'sink.properties.merge_condition' = 'score',
    'sink.version' = 'V1'
    );
```

3. Вставьте две строки данных в таблицу Flink. Первичные ключи строк данных такие же, как у строк в таблице Selena. Первая строка данных имеет меньшее значение в столбце `score`, а вторая строка данных имеет большее значение в столбце `score`.

```
INSERT INTO `score_board` VALUES (1, 'selena-update', 99), (2, 'flink-
update', 101);
```

4. Запросите таблицу Selena в клиенте MySQL.

```
mysql> select * from score_board;
+------+--------------+-------+
| id   | name         | score |
+------+--------------+-------+
|    1 | selena     |   100 |
|    2 | flink-update |   101 |
+------+--------------+-------+
2 rows in set (0.03 sec)
```

Вы можете видеть, что изменились только значения второй строки данных, а значения первой строки данных не изменились.

Загрузка данных в столбцы типа BITMAP

`BITMAP` часто используется для ускорения count distinct, например, для подсчета UV, см. Использование Bitmap для точного Count Distinct. Здесь мы возьмем подсчет UV в качестве примера, чтобы показать, как загружать данные в столбцы типа `BITMAP`.

1. Создайте таблицу Aggregate Selena в клиенте MySQL.

   В базе данных `test` создайте таблицу Aggregate `page_uv`, где столбец `visit_users` определен как тип `BITMAP` и настроен с агрегатной функцией `BITMAP_UNION`.

```
CREATE TABLE `test`.`page_uv` (
  `page_id` INT NOT NULL COMMENT 'page ID',
  `visit_date` datetime NOT NULL COMMENT 'access time',
  `visit_users` BITMAP BITMAP_UNION NOT NULL COMMENT 'user ID'
) ENGINE=OLAP
AGGREGATE KEY(`page_id`, `visit_date`)
DISTRIBUTED BY HASH(`page_id`);
```

2. Создайте таблицу Flink в клиенте Flink SQL.

   Столбец `visit_user_id` в таблице Flink имеет тип `BIGINT`, и мы хотим загрузить этот столбец в столбец `visit_users` типа `BITMAP` в таблице Selena. Поэтому при определении DDL таблицы Flink обратите внимание, что:

   - Поскольку Flink не поддерживает `BITMAP`, вам необходимо определить столбец `visit_user_id` как тип `BIGINT` для представления столбца `visit_users` типа `BITMAP` в таблице Selena.

- Вам необходимо установить опцию `sink.properties.columns` в `page_id,visit_date,user_id,visit_users=to_bitmap(visit_user_id)`, что сообщает connector сопоставление столбцов между таблицей Flink и таблицей Selena. Также вам необходимо использовать функцию `to_bitmap` чтобы сообщить connector преобразовать данные типа `BIGINT` в тип `BITMAP`.

```
CREATE TABLE `page_uv` (
    `page_id` INT,
    `visit_date` TIMESTAMP,
    `visit_user_id` BIGINT
) WITH (
    'connector' = 'selena',
    'jdbc-url' = 'jdbc:mysql://127.0.0.1:9030',
    'load-url' = '127.0.0.1:8030',
    'database-name' = 'test',
    'table-name' = 'page_uv',
    'username' = 'root',
    'password' = '',
    'sink.properties.columns' =
'page_id,visit_date,visit_user_id,visit_users=to_bitmap(visit_user_id)'
);
```

3. Загрузите данные в таблицу Flink в клиенте Flink SQL.

```
INSERT INTO `page_uv` VALUES
   (1, CAST('2020-06-23 01:30:30' AS TIMESTAMP), 13),
   (1, CAST('2020-06-23 01:30:30' AS TIMESTAMP), 23),
   (1, CAST('2020-06-23 01:30:30' AS TIMESTAMP), 33),
   (1, CAST('2020-06-23 02:30:30' AS TIMESTAMP), 13),
   (2, CAST('2020-06-23 01:30:30' AS TIMESTAMP), 23);
```

4. Вычислите UV страниц из таблицы Selena в клиенте MySQL.

```
MySQL [test]> SELECT `page_id`, COUNT(DISTINCT `visit_users`) FROM
`page_uv` GROUP BY `page_id`;
+---------+-----------------------------+
| page_id | count(DISTINCT visit_users) |
+---------+-----------------------------+
|       2 |                           1 |
|       1 |                           3 |
+---------+-----------------------------+
2 rows in set (0.05 sec)
```

Загрузка данных в столбцы типа HLL

`HLL` может использоваться для приблизительного count distinct, см. Использование HLL для приблизительного count distinct.

Здесь мы возьмем подсчет UV в качестве примера, чтобы показать, как загружать данные в столбцы типа `HLL`.

1. Создайте таблицу Aggregate Selena

   В базе данных `test` создайте таблицу Aggregate `hll_uv`, где столбец `visit_users` определен как тип `HLL` и настроен с агрегатной функцией `HLL_UNION`.

```
CREATE TABLE `hll_uv` (
  `page_id` INT NOT NULL COMMENT 'page ID',
  `visit_date` datetime NOT NULL COMMENT 'access time',
  `visit_users` HLL HLL_UNION NOT NULL COMMENT 'user ID'
) ENGINE=OLAP
AGGREGATE KEY(`page_id`, `visit_date`)
DISTRIBUTED BY HASH(`page_id`);
```

2. Создайте таблицу Flink в клиенте Flink SQL.

   Столбец `visit_user_id` в таблице Flink имеет тип `BIGINT`, и мы хотим загрузить этот столбец в столбец `visit_users` типа `HLL` в таблице Selena. Поэтому при определении DDL таблицы Flink обратите внимание, что:

   - Поскольку Flink не поддерживает `BITMAP`, вам необходимо определить столбец `visit_user_id` как тип `BIGINT` для представления столбца `visit_users` типа `HLL` в таблице Selena.

- Вам необходимо установить опцию `sink.properties.columns` в `page_id,visit_date,user_id,visit_users=hll_hash(visit_user_id)`, что сообщает connector сопоставление столбцов между таблицей Flink и таблицей Selena. Также вам необходимо использовать функцию `hll_hash` чтобы сообщить connector преобразовать данные типа `BIGINT` в тип `HLL`.

```
CREATE TABLE `hll_uv` (
    `page_id` INT,
    `visit_date` TIMESTAMP,
    `visit_user_id` BIGINT
) WITH (
    'connector' = 'selena',
    'jdbc-url' = 'jdbc:mysql://127.0.0.1:9030',
    'load-url' = '127.0.0.1:8030',
    'database-name' = 'test',
    'table-name' = 'hll_uv',
    'username' = 'root',
    'password' = '',
    'sink.properties.columns' =
'page_id,visit_date,visit_user_id,visit_users=hll_hash(visit_user_id)'
);
```

3. Загрузите данные в таблицу Flink в клиенте Flink SQL.

```
INSERT INTO `hll_uv` VALUES
   (3, CAST('2023-07-24 12:00:00' AS TIMESTAMP), 78),
   (4, CAST('2023-07-24 13:20:10' AS TIMESTAMP), 2),
   (3, CAST('2023-07-24 12:30:00' AS TIMESTAMP), 674);
```

4. Вычислите UV страниц из таблицы Selena в клиенте MySQL.

```
mysql> SELECT `page_id`, COUNT(DISTINCT `visit_users`) FROM `hll_uv`
GROUP BY `page_id`;
**+---------+-----------------------------+
| page_id | count(DISTINCT visit_users) |
+---------+-----------------------------+
|       3 |                           2 |
|       4 |                           1 |
+---------+-----------------------------+
2 rows in set (0.04 sec)
```

1.3.3

Kafka connector для Selena

Загрузка данных с помощью Kafka connector

Selena предоставляет собственный connector под названием Apache Kafka® connector (Selena Connector for Apache Kafka®, сокращённо Kafka connector), который является sink connector'ом и непрерывно потребляет сообщения из Kafka, загружая их в Selena. Kafka connector гарантирует семантику at-least-once.

Kafka connector может легко интегрироваться с Kafka Connect, что позволяет Selena лучше интегрироваться с экосистемой Kafka. Это мудрый выбор, если вы хотите загружать данные в реальном времени в Selena. По сравнению с Routine Load, рекомендуется использовать Kafka connector в следующих сценариях:

- В отличие от Routine Load, которая поддерживает загрузку данных только в форматах CSV, JSON и Avro, Kafka connector может загружать данные в большем количестве форматов, таких как Protobuf. Если данные можно преобразовать в форматы JSON и CSV с помощью конвертеров Kafka Connect, то данные можно загрузить в Selena через Kafka connector.
- Настройка пользовательского преобразования данных, например, CDC-данных в формате Debezium.
- Загрузка данных из нескольких топиков Kafka.
- Загрузка данных из Confluent Cloud.
- Более точный контроль над размерами пакетов загрузки, параллелизмом и другими параметрами для достижения баланса между скоростью загрузки и использованием ресурсов.

Подготовка

Требования к версиям

Connector	Kafka	Selena	Java
1.0.4	3.4	2.5 и выше	8
1.0.3	3.4	2.5 и выше	8

Настройка окружения Kafka

Поддерживаются как самоуправляемые кластеры Apache Kafka, так и Confluent Cloud.

- Для самоуправляемого кластера Apache Kafka вы можете обратиться к Apache Kafka quickstart для быстрого развёртывания кластера Kafka. Kafka Connect уже интегрирован в Kafka.
- Для Confluent Cloud убедитесь, что у вас есть учётная запись Confluent и вы создали кластер.

Загрузка Kafka connector

Добавьте Kafka connector в Kafka Connect:

- Самоуправляемый кластер Kafka:

  Скачайте и распакуйте selena-kafka-connector-xxx.tar.gz.

- Confluent Cloud:

  В настоящее время Kafka connector не загружен в Confluent Hub. Вам нужно скачать и распаковать selena-kafka-connector-xxx.tar.gz, упаковать его в ZIP-файл и загрузить ZIP-файл в Confluent Cloud.

Конфигурация сети

Убедитесь, что машина, на которой расположен Kafka, может получить доступ к узлам FE кластера Selena через `http_port` (по умолчанию: `8030`) и `query_port` (по умолчанию: `9030`), а также к узлам BE через `be_http_port` (по умолчанию: `8040`).

Использование

В этом разделе в качестве примера используется самоуправляемый кластер Kafka для объяснения того, как настроить Kafka connector и Kafka Connect, а затем запустить Kafka Connect для загрузки данных в Selena.

Подготовка набора данных

Предположим, что в топике `test` кластера Kafka существуют данные в формате JSON.

```
{"id":1,"city":"New York"}
{"id":2,"city":"Los Angeles"}
{"id":3,"city":"Chicago"}
```

Создание таблицы

Создайте таблицу `test_tbl` в базе данных `example_db` в кластере Selena в соответствии с ключами данных в формате JSON.

```
CREATE DATABASE example_db;
USE example_db;
CREATE TABLE test_tbl (id INT, city STRING);
```

Настройка Kafka connector и Kafka Connect, запуск Kafka Connect для загрузки данных

Запуск Kafka Connect в автономном режиме

1. Настройте Kafka connector. В директории **config** в каталоге установки Kafka создайте файл конфигурации **connect-Selena-sink.properties** для Kafka connector и настройте следующие параметры. Больше параметров и их описаний см. в разделе Параметры.

> **К СВЕДЕНИЮ**
>
> - В этом примере Kafka connector, предоставляемый Selena, является sink connector'ом, который может непрерывно потреблять данные из Kafka и загружать данные в Selena.
> - Если исходные данные являются CDC-данными, например, данными в формате Debezium, и таблица Selena является таблицей Primary Key, вам также необходимо настроить `transform` в файле конфигурации **connect-Selena-sink.properties** для Kafka connector, предоставляемого Selena, чтобы синхронизировать изменения исходных данных с таблицей Primary Key.

```
name=selena-kafka-connector
connector.class=com.selena.connector.kafka.SelenaSinkConnector
topics=test
key.converter=org.apache.kafka.connect.json.JsonConverter
value.converter=org.apache.kafka.connect.json.JsonConverter
key.converter.schemas.enable=true
value.converter.schemas.enable=false
# HTTP URL узла FE в вашем кластере Selena. Порт по умолчанию - 8030.
selena.http.url=192.168.xxx.xxx:8030
# Если имя топика Kafka отличается от имени таблицы Selena, необходимо
настроить соответствие между ними.
selena.topic2table.map=test:test_tbl
# Введите имя пользователя Selena.
selena.username=user1
# Введите пароль Selena.
selena.password=123456
selena.database.name=example_db
sink.properties.strip_outer_array=true
```

2. Настройте и запустите Kafka Connect.

   i. Настройте Kafka Connect. В файле конфигурации **config/connect-standalone.properties** в директории **config** настройте следующие параметры. Больше параметров и их описаний см. в разделе Running Kafka Connect. Обратите внимание, что в следующих примерах используется версия selena-kafka-connector `1.0.3`. Если вы используете более новую версию, вам необходимо внести соответствующие изменения.

```
# Адреса брокеров Kafka. Несколько адресов брокеров Kafka должны быть разделены запятыми (,).
# Обратите внимание, что в этом примере используется PLAINTEXT в качестве протокола безопасности для доступа к кластеру Kafka. Если вы используете другой протокол безопасности для доступа к кластеру Kafka, вам необходимо настроить соответствующую информацию в этом файле.
bootstrap.servers=<kafka_broker_ip>:9092
offset.storage.file.filename=/tmp/connect.offsets
offset.flush.interval.ms=10000
key.converter=org.apache.kafka.connect.json.JsonConverter
value.converter=org.apache.kafka.connect.json.JsonConverter
key.converter.schemas.enable=true
value.converter.schemas.enable=false
# Абсолютный путь к Kafka connector после распаковки. Например:
plugin.path=/home/kafka-connect/selena-kafka-connector-1.0.3
```

ii. Запустите Kafka Connect.

```
CLASSPATH=/home/kafka-connect/selena-kafka-connector-1.0.3/*
bin/connect-standalone.sh config/connect-standalone.properties
config/connect-selena-sink.properties
```

Запуск Kafka Connect в распределённом режиме

1. Настройте и запустите Kafka Connect.

   i. Настройте Kafka Connect. В файле конфигурации `config/connect-distributed.properties` в директории **config** настройте следующие параметры. Больше параметров и их описаний см. в разделе Running Kafka Connect.

```
# Адреса брокеров Kafka. Несколько адресов брокеров Kafka должны
быть разделены запятыми (,).
# Обратите внимание, что в этом примере используется PLAINTEXT в
качестве протокола безопасности для доступа к кластеру Kafka. Если
вы используете другой протокол безопасности для доступа к кластеру
Kafka, вам необходимо настроить соответствующую информацию в этом
файле.
bootstrap.servers=<kafka_broker_ip>:9092
offset.storage.file.filename=/tmp/connect.offsets
offset.flush.interval.ms=10000
key.converter=org.apache.kafka.connect.json.JsonConverter
value.converter=org.apache.kafka.connect.json.JsonConverter
key.converter.schemas.enable=true
value.converter.schemas.enable=false
# Абсолютный путь к Kafka connector после распаковки. Например:
plugin.path=/home/kafka-connect/selena-kafka-connector-1.0.3
```

ii. Запустите Kafka Connect.

```
CLASSPATH=/home/kafka-connect/selena-kafka-connector-1.0.3/*
bin/connect-distributed.sh config/connect-distributed.properties
```

2. Настройте и создайте Kafka connector. Обратите внимание, что в распределённом режиме вам необходимо настроить и создать Kafka connector через REST API. Параметры и их описания см. в разделе Параметры.

> **К СВЕДЕНИЮ**
>
> - В этом примере Kafka connector, предоставляемый Selena, является sink connector'ом, который может непрерывно потреблять данные из Kafka и загружать данные в Selena.
> - Если исходные данные являются CDC-данными, например, данными в формате Debezium, и таблица Selena является таблицей Primary Key, вам также необходимо настроить `transform` в файле конфигурации **connect-Selena-sink.properties** для Kafka connector, предоставляемого Selena, чтобы синхронизировать изменения исходных данных с таблицей Primary Key.

```
curl -i http://127.0.0.1:8083/connectors -H "Content-Type:
application/json" -X POST -d '{
  "name":"selena-kafka-connector",
  "config":{
    "connector.class":"com.selena.connector.kafka.SelenaSinkConnector",
    "topics":"test",
    "key.converter":"org.apache.kafka.connect.json.JsonConverter",
    "value.converter":"org.apache.kafka.connect.json.JsonConverter",
    "key.converter.schemas.enable":"true",
    "value.converter.schemas.enable":"false",
    "selena.http.url":"192.168.xxx.xxx:8030",
    "selena.topic2table.map":"test:test_tbl",
    "selena.username":"user1",
    "selena.password":"123456",
    "selena.database.name":"example_db",
    "sink.properties.strip_outer_array":"true"
  }
}'
```

Запрос таблицы Selena

Запросите целевую таблицу Selena `test_tbl`.

```
MySQL [example_db]> select * from test_tbl;

+------+-------------+
| id   | city        |
+------+-------------+
|    1 | New York    |
|    2 | Los Angeles |
|    3 | Chicago     |
+------+-------------+
3 rows in set (0.01 sec)
```

Данные успешно загружены, когда возвращается вышеуказанный результат.

Параметры

name

Обязательный: ДА

Значение по умолчанию:

Описание: Имя для этого Kafka connector. Оно должно быть глобально уникальным среди всех Kafka connector'ов в этом кластере Kafka Connect. Например, selena-kafka-connector.

connector.class

Обязательный: ДА
Значение по умолчанию:
Описание: Класс, используемый sink'ом этого Kafka connector. Установите значение `com.selena.connector.kafka.SelenaSinkConnector`.

topics

Обязательный:
Значение по умолчанию:
Описание: Один или несколько топиков для подписки, где каждый топик соответствует таблице Selena. По умолчанию Selena предполагает, что имя топика совпадает с именем таблицы Selena. Таким образом, Selena определяет целевую таблицу Selena, используя имя топика. Пожалуйста, выберите заполнение либо `topics`, либо `topics.regex` (ниже), но не оба одновременно. Однако, если имя таблицы Selena не совпадает с именем топика, используйте необязательный параметр `selena.topic2table.map` (ниже) для указания соответствия между именем топика и именем таблицы.

topics.regex

Обязательный:
Значение по умолчанию: **Описание**: Регулярное выражение для сопоставления одного или нескольких топиков для подписки. Более подробное описание см. в `topics`. Пожалуйста, выберите заполнение либо `topics.regex`, либо `topics` (выше), но не оба одновременно.

selena.topic2table.map

Обязательный: НЕТ
Значение по умолчанию:
Описание: Соответствие между именем таблицы Selena и именем топика, когда имя топика отличается от имени таблицы Selena. Формат: `<topic-1>:<table-1>,<topic-2>:<table-2>,...`.

selena.http.url

Обязательный: ДА
Значение по умолчанию:
Описание: HTTP URL узла FE в вашем кластере Selena. Формат: `<fe_host1>:<fe_http_port1>,<fe_host2>:<fe_http_port2>,...`. Несколько адресов разделяются запятыми (,). Например, `192.168.xxx.xxx:8030,192.168.xxx.xxx:8030`.

selena.database.name

Обязательный: ДА
Значение по умолчанию:
Описание: Имя базы данных Selena.

selena.username

Обязательный: ДА
Значение по умолчанию:
Описание: Имя пользователя вашей учётной записи кластера Selena. Пользователь должен иметь привилегию INSERT на таблицу Selena.

selena.password

Обязательный: ДА
Значение по умолчанию:
Описание: Пароль вашей учётной записи кластера Selena.

key.converter

Обязательный: НЕТ
Значение по умолчанию: Key converter, используемый кластером Kafka Connect
Описание: Этот параметр указывает key converter для sink connector (Kafka-connector-selena), который используется для десериализации ключей данных Kafka. Key converter по умолчанию - тот, который используется кластером Kafka Connect.

value.converter

Обязательный: НЕТ
Значение по умолчанию: Value converter, используемый кластером Kafka Connect
Описание: Этот параметр указывает value converter для sink connector (Kafka-connector-selena), который используется для десериализации значений данных Kafka. Value converter по умолчанию - тот, который используется кластером Kafka Connect.

key.converter.schema.registry.url

Обязательный: НЕТ
Значение по умолчанию:
Описание: URL Schema registry для key converter.

value.converter.schema.registry.url

Обязательный: НЕТ
Значение по умолчанию:
Описание: URL Schema registry для value converter.

tasks.max

Обязательный: НЕТ
Значение по умолчанию: 1
Описание: Верхний лимит для количества потоков задач, которые может создать Kafka connector, который обычно совпадает с количеством ядер CPU на рабочих узлах в кластере Kafka Connect. Вы можете настроить этот параметр для управления производительностью загрузки.

bufferflush.maxbytes

Обязательный: НЕТ
Значение по умолчанию: 94371840(90M)
Описание: Максимальный размер данных, которые могут быть накоплены в памяти перед отправкой в Selena за один раз. Максимальное значение варьируется от 64 MB до 10 GB. Имейте в виду, что буфер Stream Load SDK может создавать несколько заданий Stream Load для буферизации данных. Поэтому упомянутый здесь порог относится к общему размеру данных.

bufferflush.intervalms

Обязательный: НЕТ
Значение по умолчанию: 1000
Описание: Интервал для отправки пакета данных, который контролирует задержку загрузки. Диапазон: [1000, 3600000].

connect.timeoutms

Обязательный: НЕТ
Значение по умолчанию: 1000

Описание: Тайм-аут для подключения к HTTP URL. Диапазон: [100, 60000].

sink.properties.*

Обязательный:
Значение по умолчанию:
Описание: Параметры Stream Load для управления поведением загрузки. Например, параметр `sink.properties.format` указывает формат, используемый для Stream Load, такой как CSV или JSON. Список поддерживаемых параметров и их описания см. в разделе STREAM LOAD.

sink.properties.format

Обязательный: НЕТ
Значение по умолчанию: json
Описание: Формат, используемый для Stream Load. Kafka connector будет преобразовывать каждый пакет данных в указанный формат перед отправкой их в Selena. Допустимые значения: `csv` и `json`. Для получения дополнительной информации см. Параметры CSV и Параметры JSON.

sink.properties.partial_update

Обязательный: НЕТ
Значение по умолчанию: `FALSE`
Описание: Использовать ли частичные обновления. Допустимые значения: `TRUE` и `FALSE`. Значение по умолчанию: `FALSE`, что означает отключение этой функции.

sink.properties.partial_update_mode

Обязательный: НЕТ
Значение по умолчанию: `row`
Описание: Указывает режим для частичных обновлений. Допустимые значения: `row` и `column`.

- Значение `row` (по умолчанию) означает частичные обновления в режиме строк, что более подходит для обновлений в реальном времени с большим количеством столбцов и небольшими пакетами.
- Значение `column` означает частичные обновления в режиме столбцов, что более подходит для пакетных обновлений с небольшим количеством столбцов и большим количеством строк. В таких сценариях включение режима столбцов обеспечивает более высокую скорость обновления. Например, в таблице с 100 столбцами, если

обновляется только 10 столбцов (10% от общего количества) для всех строк, скорость обновления в режиме столбцов в 10 раз выше.

Примечания по использованию

Политика сброса

Kafka connector будет буферизовать данные в памяти и сбрасывать их пакетами в Selena через Stream Load. Сброс будет запущен при выполнении любого из следующих условий:

- Размер буферизованных строк в байтах достигает лимита `bufferflush.maxbytes`.
- Время, прошедшее с момента последнего сброса, достигает лимита `bufferflush.intervalms`.
- Достигается интервал, с которым connector пытается фиксировать смещения для задач. Интервал контролируется конфигурацией Kafka Connect `offset.flush.interval.ms`, и значение по умолчанию составляет `60000`.

Для снижения задержки данных настройте эти конфигурации в настройках Kafka connector. Однако более частые сбросы увеличат использование CPU и I/O.

Ограничения

- Не поддерживается преобразование одного сообщения из топика Kafka в несколько строк данных и загрузка в Selena.
- Sink Kafka connector, предоставляемый Selena, гарантирует семантику at-least-once.

Лучшие практики

Загрузка CDC-данных в формате Debezium

Debezium - это популярный инструмент Change Data Capture (CDC), который поддерживает мониторинг изменений данных в различных системах баз данных и потоковую передачу этих изменений в Kafka. Следующий пример демонстрирует, как настроить и использовать Kafka connector для записи изменений PostgreSQL в **таблицу Primary Key** в Selena.

Шаг 1: Установка и запуск Kafka

> **ПРИМЕЧАНИЕ**
>
> Вы можете пропустить этот шаг, если у вас есть собственное окружение Kafka.

1. Скачайте последний релиз Kafka с официального сайта и распакуйте пакет.

```
tar -xzf kafka_2.13-3.7.0.tgz
cd kafka_2.13-3.7.0
```

2. Запустите окружение Kafka.

   Сгенерируйте UUID кластера Kafka.

```
KAFKA_CLUSTER_ID="$(bin/kafka-storage.sh random-uuid)"
```

   Отформатируйте директории логов.

```
bin/kafka-storage.sh format -t $KAFKA_CLUSTER_ID -c
config/kraft/server.properties
```

   Запустите сервер Kafka.

```
bin/kafka-server-start.sh config/kraft/server.properties
```

Шаг 2: Настройка PostgreSQL

1. Убедитесь, что пользователю PostgreSQL предоставлены привилегии `REPLICATION`.

2. Настройте конфигурацию PostgreSQL.

   Установите `wal_level` в `logical` в **postgresql.conf**.

```
wal_level = logical
```

   Перезапустите сервер PostgreSQL для применения изменений.

```
pg_ctl restart
```

3. Подготовьте набор данных.

   Создайте таблицу и вставьте тестовые данные.

```sql
CREATE TABLE customers (
  id int primary key ,
  first_name varchar(65533) NULL,
  last_name varchar(65533) NULL ,
  email varchar(65533) NULL
);

INSERT INTO customers VALUES (1,'a','a','a@a.com');
```

4. Проверьте сообщения CDC-логов в Kafka.

```
{
    "schema": {
        "type": "struct",
        "fields": [
            {
                "type": "struct",
                "fields": [
                    {
                        "type": "int32",
                        "optional": false,
                        "field": "id"
                    },
                    {
                        "type": "string",
                        "optional": true,
                        "field": "first_name"
                    },
                    {
                        "type": "string",
                        "optional": true,
                        "field": "last_name"
                    },
                    {
                        "type": "string",
                        "optional": true,
                        "field": "email"
                    }
                ],
                "optional": true,
                "name": "test.public.customers.Value",
                "field": "before"
            },
            {
                "type": "struct",
                "fields": [
                    {
                        "type": "int32",
                        "optional": false,
                        "field": "id"
                    },
                    {
                        "type": "string",
                        "optional": true,
                        "field": "first_name"
                    },
```

```
                {
                    "type": "string",
                    "optional": true,
                    "field": "last_name"
                },
                {
                    "type": "string",
                    "optional": true,
                    "field": "email"
                }
            ],
            "optional": true,
            "name": "test.public.customers.Value",
            "field": "after"
        },
        {
            "type": "struct",
            "fields": [
                {
                    "type": "string",
                    "optional": false,
                    "field": "version"
                },
                {
                    "type": "string",
                    "optional": false,
                    "field": "connector"
                },
                {
                    "type": "string",
                    "optional": false,
                    "field": "name"
                },
                {
                    "type": "int64",
                    "optional": false,
                    "field": "ts_ms"
                },
                {
                    "type": "string",
                    "optional": true,
                    "name": "io.debezium.data.Enum",
                    "version": 1,
                    "parameters": {
                        "allowed": "true,last,false,incremental"
                    },
```

```
                    "default": "false",
                    "field": "snapshot"
                },
                {
                    "type": "string",
                    "optional": false,
                    "field": "db"
                },
                {
                    "type": "string",
                    "optional": true,
                    "field": "sequence"
                },
                {
                    "type": "string",
                    "optional": false,
                    "field": "schema"
                },
                {
                    "type": "string",
                    "optional": false,
                    "field": "table"
                },
                {
                    "type": "int64",
                    "optional": true,
                    "field": "txId"
                },
                {
                    "type": "int64",
                    "optional": true,
                    "field": "lsn"
                },
                {
                    "type": "int64",
                    "optional": true,
                    "field": "xmin"
                }
            ],
            "optional": false,
            "name": "io.debezium.connector.postgresql.Source",
            "field": "source"
        },
        {
            "type": "string",
            "optional": false,
```

```
                "field": "op"
            },
            {
                "type": "int64",
                "optional": true,
                "field": "ts_ms"
            },
            {
                "type": "struct",
                "fields": [
                    {
                        "type": "string",
                        "optional": false,
                        "field": "id"
                    },
                    {
                        "type": "int64",
                        "optional": false,
                        "field": "total_order"
                    },
                    {
                        "type": "int64",
                        "optional": false,
                        "field": "data_collection_order"
                    }
                ],
                "optional": true,
                "name": "event.block",
                "version": 1,
                "field": "transaction"
            }
        ],
        "optional": false,
        "name": "test.public.customers.Envelope",
        "version": 1
    },
    "payload": {
        "before": null,
        "after": {
            "id": 1,
            "first_name": "a",
            "last_name": "a",
            "email": "a@a.com"
        },
        "source": {
            "version": "2.5.3.Final",
```

```
            "connector": "postgresql",
            "name": "test",
            "ts_ms": 1714283798721,
            "snapshot": "false",
            "db": "postgres",
            "sequence": "[\"22910216\",\"22910504\"]",
            "schema": "public",
            "table": "customers",
            "txId": 756,
            "lsn": 22910504,
            "xmin": null
        },
        "op": "c",
        "ts_ms": 1714283798790,
        "transaction": null
    }
}
```

Шаг 3: Настройка Selena

Создайте таблицу Primary Key в Selena с той же схемой, что и исходная таблица в PostgreSQL.

```
CREATE TABLE `customers` (
  `id` int(11) COMMENT "",
  `first_name` varchar(65533) NULL COMMENT "",
  `last_name` varchar(65533) NULL COMMENT "",
  `email` varchar(65533) NULL COMMENT ""
) ENGINE=OLAP
PRIMARY KEY(`id`)
DISTRIBUTED BY hash(id) buckets 1
PROPERTIES (
"bucket_size" = "4294967296",
"in_memory" = "false",
"enable_persistent_index" = "true",
"replicated_storage" = "true",
"fast_schema_evolution" = "true"
);
```

Шаг 4: Установка connector

1. Скачайте connector'ы и распакуйте пакеты в директорию **plugins**.

```
mkdir plugins
tar -zxvf debezium-debezium-connector-postgresql-2.5.3.zip -C plugins
tar -zxvf selena-kafka-connector-1.0.3.tar.gz -C plugins
```

Эта директория является значением элемента конфигурации `plugin.path` в **config/connect-standalone.properties**.

```
plugin.path=/path/to/kafka_2.13-3.7.0/plugins
```

2. Настройте source connector для PostgreSQL в **pg-source.properties**.

```
{
  "name": "inventory-connector",
  "config": {
    "connector.class":
"io.debezium.connector.postgresql.PostgresConnector",
    "plugin.name": "pgoutput",
    "database.hostname": "localhost",
    "database.port": "5432",
    "database.user": "postgres",
    "database.password": "",
    "database.dbname" : "postgres",
    "topic.prefix": "test"
  }
}
```

3. Настройте sink connector для Selena в **sr-sink.properties**.

```
{
    "name": "selena-kafka-connector",
    "config": {
        "connector.class":
"com.selena.connector.kafka.SelenaSinkConnector",
        "tasks.max": "1",
        "topics": "test.public.customers",
        "selena.http.url": "172.26.195.69:28030",
        "selena.database.name": "test",
        "selena.username": "root",
        "selena.password": "Selena@123",
        "sink.properties.strip_outer_array": "true",
        "connect.timeoutms": "3000",
        "selena.topic2table.map": "test.public.customers:customers",
        "transforms": "addfield,unwrap",
        "transforms.addfield.type":
"com.selena.connector.kafka.transforms.AddOpFieldForDebeziumRecord",
        "transforms.unwrap.type":
"io.debezium.transforms.ExtractNewRecordState",
        "transforms.unwrap.drop.tombstones": "true",
        "transforms.unwrap.delete.handling.mode": "rewrite"
    }
}
```

> **ПРИМЕЧАНИЕ**
>
> - Если таблица Selena не является таблицей Primary Key, вам не нужно указывать трансформацию `addfield`.
> - Трансформация unwrap предоставляется Debezium и используется для распаковки сложной структуры данных Debezium на основе типа операции. Для получения дополнительной информации см. New Record State Extraction.

4. Настройте Kafka Connect.

   Настройте следующие элементы конфигурации в файле конфигурации Kafka Connect **config/connect-standalone.properties**.

```
# Адреса брокеров Kafka. Несколько адресов брокеров Kafka должны быть
разделены запятыми (,).
# Обратите внимание, что в этом примере используется PLAINTEXT в
качестве протокола безопасности для доступа к кластеру Kafka.
# Если вы используете другой протокол безопасности для доступа к
кластеру Kafka, настройте соответствующую информацию в этой части.

bootstrap.servers=<kafka_broker_ip>:9092
offset.storage.file.filename=/tmp/connect.offsets
key.converter=org.apache.kafka.connect.json.JsonConverter
value.converter=org.apache.kafka.connect.json.JsonConverter
key.converter.schemas.enable=true
value.converter.schemas.enable=false

# Абсолютный путь к selena-kafka-connector после распаковки. Например:
plugin.path=/home/kafka-connect/selena-kafka-connector-1.0.3

# Параметры, которые управляют политикой сброса. Для получения
дополнительной информации см. раздел Примечания по использованию.
offset.flush.interval.ms=10000
bufferflush.maxbytes = xxx
bufferflush.intervalms = xxx
```

Для описания дополнительных параметров см. Running Kafka Connect.

Шаг 5: Запуск Kafka Connect в автономном режиме

Запустите Kafka Connect в автономном режиме для инициализации connector'ов.

```
bin/connect-standalone.sh config/connect-standalone.properties config/pg-
source.properties config/sr-sink.properties
```

Шаг 6: Проверка загрузки данных

Протестируйте следующие операции и убедитесь, что данные правильно загружены в Selena.

INSERT

- В PostgreSQL:

```
postgres=# insert into customers values (2,'b','b','b@b.com');
INSERT 0 1
postgres=# select * from customers;
 id | first_name | last_name |  email
----+------------+-----------+---------
  1 | a          | a         | a@a.com
  2 | b          | b         | b@b.com
(2 rows)
```

- В Selena:

```
MySQL [test]> select * from customers;
+------+------------+-----------+---------+
| id   | first_name | last_name | email   |
+------+------------+-----------+---------+
|    1 | a          | a         | a@a.com |
|    2 | b          | b         | b@b.com |
+------+------------+-----------+---------+
2 rows in set (0.01 sec)
```

UPDATE

- В PostgreSQL:

```
postgres=# update customers set email='c@c.com';
UPDATE 2
postgres=# select * from customers;
 id | first_name | last_name |  email
----+------------+-----------+---------
  1 | a          | a         | c@c.com
  2 | b          | b         | c@c.com
(2 rows)
```

- В Selena:

```
MySQL [test]> select * from customers;
+------+------------+-----------+---------+
| id   | first_name | last_name | email   |
+------+------------+-----------+---------+
|    1 | a          | a         | c@c.com |
|    2 | b          | b         | c@c.com |
+------+------------+-----------+---------+
2 rows in set (0.00 sec)
```

DELETE

- В PostgreSQL:

```
postgres=# delete from customers where id=1;
DELETE 1
postgres=# select * from customers;
 id | first_name | last_name |  email
----+------------+-----------+---------
  2 | b          | b         | c@c.com
(1 row)
```

- В Selena:

```
MySQL [test]> select * from customers;
+------+------------+-----------+---------+
| id   | first_name | last_name | email   |
+------+------------+-----------+---------+
|    2 | b          | b         | c@c.com |
+------+------------+-----------+---------+
1 row in set (0.00 sec)
```

Routine Load

Загрузка данных с помощью Routine Load

> **ПОДСКАЗКА**
>
> Попробуйте Routine Load в этом Кратком руководстве

В этом разделе описывается, как создать задачу Routine Load для потоковой передачи сообщений Kafka (событий) в Selena, а также знакомит вас с основными концепциями

Routine Load.

Для непрерывной загрузки сообщений потока в Selena вы можете сохранить поток сообщений в topic Kafka и создать задачу Routine Load для потребления сообщений. Задача Routine Load сохраняется в Selena, генерирует серию подзадач загрузки для потребления сообщений из всех или части partition в topic и загружает сообщения в Selena.

Задача Routine Load поддерживает семантику exactly-once доставки, гарантируя, что данные, загруженные в Selena, не будут потеряны или дублированы.

Routine Load поддерживает преобразование данных при загрузке и поддерживает изменения данных, выполняемые операциями UPSERT и DELETE во время загрузки данных. Для получения дополнительной информации см. Преобразование данных при загрузке и Изменение данных через загрузку.

Вы можете загружать данные в таблицы Selena только как пользователь, имеющий привилегию INSERT на эти таблицы Selena. Если у вас нет привилегии INSERT, следуйте инструкциям в разделе GRANT, чтобы предоставить привилегию INSERT пользователю, которого вы используете для подключения к вашему cluster Selena. Синтаксис: `GRANT INSERT ON TABLE <table_name> IN DATABASE <database_name> TO { ROLE <role_name> | USER <user_identity>}`.

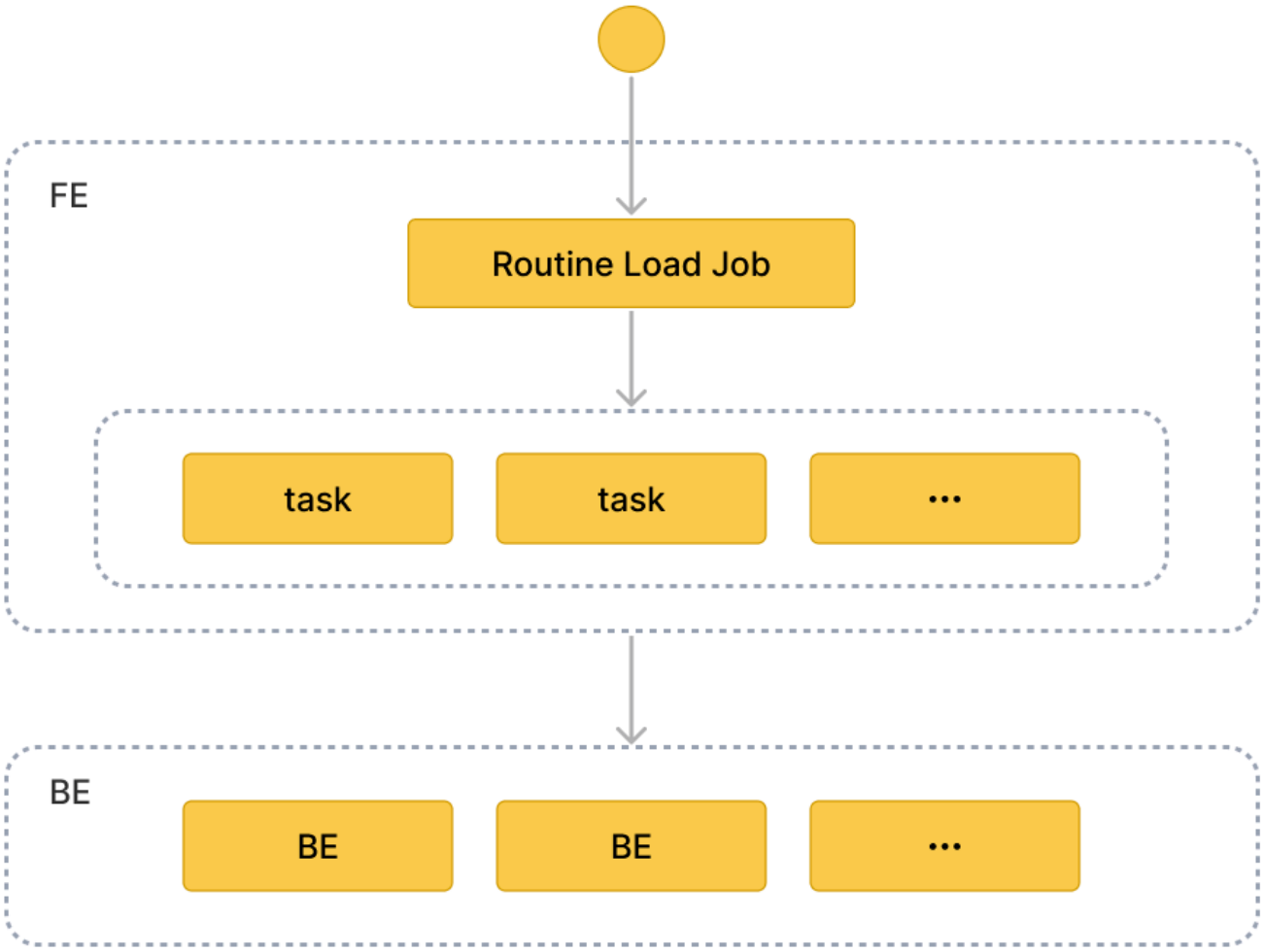
Поддерживаемые форматы данных

Routine Load в настоящее время поддерживает потребление данных в форматах CSV, JSON и Avro (поддерживается начиная с v1.5.2) из cluster Kafka.

> **ПРИМЕЧАНИЕ**
>
> Для данных CSV обратите внимание на следующие моменты:
>
> - Вы можете использовать строку UTF-8, такую как запятая (,), табуляция или вертикальная черта (|), длиной не более 50 байт в качестве текстового разделителя.
> - Null-значения обозначаются с помощью `\N`. Например, файл данных состоит из трёх столбцов, и запись из этого файла данных содержит данные в первом и третьем столбцах, но не содержит данных во втором столбце. В этой ситуации вам нужно использовать `\N` во втором столбце для обозначения null-значения.

> Это означает, что запись должна быть скомпилирована как `a,\N,b` вместо `a,,b`. `a,,b` означает, что второй столбец записи содержит пустую строку.

Основные концепции



Терминология

- **Задача загрузки (Load job)**

  Задача Routine Load — это долгосрочная задача. Пока её статус RUNNING, задача загрузки непрерывно генерирует одну или несколько параллельных подзадач загрузки, которые потребляют сообщения из topic cluster Kafka и загружают данные в Selena.

- **Подзадача загрузки (Load task)**

  Задача загрузки разделяется на несколько подзадач загрузки по определённым правилам. Подзадача загрузки — это базовая единица загрузки данных. Как

отдельное событие, подзадача загрузки реализует механизм загрузки на основе Stream Load. Несколько подзадач загрузки одновременно потребляют сообщения из разных partition topic и загружают данные в Selena.

Рабочий процесс

1. **Создайте задачу Routine Load.** Для загрузки данных из Kafka вам нужно создать задачу Routine Load, выполнив оператор CREATE ROUTINE LOAD. FE анализирует оператор и создаёт задачу в соответствии с указанными вами свойствами.

2. **FE разделяет задачу на несколько подзадач загрузки.**

   FE разделяет задачу на несколько подзадач загрузки на основе определённых правил. Каждая подзадача загрузки является отдельной транзакцией. Правила разделения следующие:

   - FE вычисляет фактическое количество параллельных подзадач загрузки в соответствии с желаемым количеством параллелизма `desired_concurrent_number`, количеством partition в topic Kafka и количеством активных узлов BE.
   - FE разделяет задачу на подзадачи загрузки на основе вычисленного фактического количества параллелизма и размещает подзадачи в очереди задач.

   Каждый topic Kafka состоит из нескольких partition. Связь между partition topic и подзадачей загрузки следующая:

   - Partition однозначно назначается подзадаче загрузки, и все сообщения из partition потребляются подзадачей загрузки.
   - Подзадача загрузки может потреблять сообщения из одной или нескольких partition.
   - Все partition распределяются равномерно между подзадачами загрузки.

3. **Несколько подзадач загрузки выполняются параллельно для потребления сообщений из нескольких partition topic Kafka и загрузки данных в Selena**

   i. **FE планирует и отправляет подзадачи загрузки**: FE планирует подзадачи загрузки в очереди на регулярной основе и назначает их выбранным узлам Coordinator BE. Интервал между подзадачами загрузки определяется параметром конфигурации `max_batch_interval`. FE равномерно распределяет подзадачи загрузки по всем узлам BE. См. CREATE ROUTINE LOAD для получения дополнительной информации о `max_batch_interval`.

ii. Coordinator BE запускает подзадачу загрузки, потребляет сообщения в partition, анализирует и фильтрует данные. Подзадача загрузки продолжается до тех пор, пока не будет потреблено заранее определённое количество сообщений или не будет достигнут заранее определённый лимит времени. Размер пакета сообщений и лимит времени определяются в конфигурациях FE `max_routine_load_batch_size` и `routine_load_task_consume_second`. Для получения подробной информации см. Конфигурация FE. Затем Coordinator BE распределяет сообщения на Executor BE. Executor BE записывают сообщения на диски.

> **ПРИМЕЧАНИЕ**
>
> Selena поддерживает доступ к Kafka через протоколы безопасности, включая SASL_SSL, SAS_PLAINTEXT, SSL и PLAINTEXT. В этом разделе в качестве примера используется подключение к Kafka через PLAINTEXT. Если вам нужно подключиться к Kafka через другие протоколы безопасности, см. CREATE ROUTINE LOAD.

4. **FE генерирует новые подзадачи загрузки для непрерывной загрузки данных.** После того как Executor BE записали данные на диски, Coordinator BE сообщает результат подзадачи загрузки в FE. На основе результата FE затем генерирует новые подзадачи загрузки для непрерывной загрузки данных. Или FE повторяет неудачные подзадачи, чтобы гарантировать, что данные, загруженные в Selena, не будут потеряны или дублированы.

Создание задачи Routine Load

Следующие три примера описывают, как потреблять данные в форматах CSV, JSON и Avro из Kafka и загружать данные в Selena путём создания задачи Routine Load. Для получения подробного синтаксиса и описания параметров см. CREATE ROUTINE LOAD.

Загрузка данных в формате CSV

В этом разделе описывается, как создать задачу Routine Load для потребления данных в формате CSV из cluster Kafka и загрузки данных в Selena.

Подготовка набора данных

Предположим, что в topic `ordertest1` в cluster Kafka есть набор данных в формате CSV. Каждое сообщение в наборе данных включает шесть полей: идентификатор заказа, дата

оплаты, имя клиента, национальность, пол и цена.

```
2020050802,2020-05-08,Johann Georg Faust,Deutschland,male,895
2020050802,2020-05-08,Julien Sorel,France,male,893
2020050803,2020-05-08,Dorian Grey,UK,male,1262
2020050901,2020-05-09,Anna Karenina",Russia,female,175
2020051001,2020-05-10,Tess Durbeyfield,US,female,986
2020051101,2020-05-11,Edogawa Conan,japan,male,8924
```

Создание таблицы

В соответствии с полями данных в формате CSV создайте таблицу `example_tbl1` в базе данных `example_db`. Следующий пример создаёт таблицу с 5 полями, исключая поле пола клиента из данных в формате CSV.

```
CREATE TABLE example_db.example_tbl1 (
    `order_id` bigint NOT NULL COMMENT "Order ID",
    `pay_dt` date NOT NULL COMMENT "Payment date",
    `customer_name` varchar(26) NULL COMMENT "Customer name",
    `nationality` varchar(26) NULL COMMENT "Nationality",
    `price`double NULL COMMENT "Price"
)
ENGINE=OLAP
DUPLICATE KEY (order_id,pay_dt)
DISTRIBUTED BY HASH(`order_id`);
```

> **ВНИМАНИЕ**
>
> Начиная с версии v1.5.2, Selena может автоматически устанавливать количество bucket (BUCKETS) при создании таблицы или добавлении partition. Вам больше не нужно вручную устанавливать количество bucket. Для получения подробной информации см. установка количества bucket.

Отправка задачи Routine Load

Выполните следующий оператор, чтобы отправить задачу Routine Load с именем `example_tbl1_ordertest1` для потребления сообщений из topic `ordertest1` и загрузки данных в таблицу `example_tbl1`. Подзадача загрузки потребляет сообщения с начального смещения в указанных partition topic.

```
CREATE ROUTINE LOAD example_db.example_tbl1_ordertest1 ON example_tbl1
COLUMNS TERMINATED BY ",",
COLUMNS (order_id, pay_dt, customer_name, nationality, temp_gender, price)
PROPERTIES
(
    "desired_concurrent_number" = "5"
)
FROM KAFKA
(
    "kafka_broker_list" = "<kafka_broker1_ip>:<kafka_broker1_port>,
<kafka_broker2_ip>:<kafka_broker2_port>",
    "kafka_topic" = "ordertest1",
    "kafka_partitions" = "0,1,2,3,4",
    "property.kafka_default_offsets" = "OFFSET_BEGINNING"
);
```

После отправки задачи загрузки вы можете выполнить оператор SHOW ROUTINE LOAD, чтобы проверить статус задачи загрузки.

- **Имя задачи загрузки**

  На одной таблице может быть несколько задач загрузки. Поэтому мы рекомендуем называть задачу загрузки с соответствующим topic Kafka и временем отправки задачи загрузки. Это помогает различать задачи загрузки на каждой таблице.

- **Разделитель столбцов**

  Свойство `COLUMN TERMINATED BY` определяет разделитель столбцов данных в формате CSV. По умолчанию это `\t`.

- **Partition и смещение topic Kafka**

  Вы можете указать свойства `kafka_partitions` и `kafka_offsets`, чтобы указать partition и смещения для потребления сообщений. Например, если вы хотите, чтобы задача загрузки потребляла сообщения из partition Kafka `"0,1,2,3,4"` topic `ordertest1` все с начальными смещениями, вы можете указать свойства следующим образом: Если вы хотите, чтобы задача загрузки потребляла сообщения из partition Kafka `"0,1,2,3,4"` и вам нужно указать отдельное начальное смещение для каждой partition, вы можете настроить следующим образом:

```
"kafka_partitions" ="0,1,2,3,4",
"kafka_offsets" = "OFFSET_BEGINNING, OFFSET_END, 1000, 2000, 3000"
```

Вы также можете установить смещения по умолчанию для всех partition с помощью свойства `property.kafka_default_offsets`.

```
"kafka_partitions" ="0,1,2,3,4",
"property.kafka_default_offsets" = "OFFSET_BEGINNING"
```

Для получения подробной информации см. CREATE ROUTINE LOAD.

- **Сопоставление и преобразование данных**

Чтобы указать связь сопоставления и преобразования между данными в формате CSV и таблицей Selena, вам нужно использовать параметр `COLUMNS`.

Сопоставление данных:

- Selena извлекает столбцы из данных в формате CSV и сопоставляет их **последовательно** с полями, объявленными в параметре `COLUMNS`.
- Selena извлекает поля, объявленные в параметре `COLUMNS`, и сопоставляет их **по имени** со столбцами таблицы Selena.

Преобразование данных:

И поскольку пример исключает столбец пола клиента из данных в формате CSV, поле `temp_gender` в параметре `COLUMNS` используется в качестве заполнителя для этого поля. Остальные поля сопоставляются со столбцами таблицы Selena `example_tbl1` напрямую.

Для получения дополнительной информации о преобразовании данных см. Преобразование данных при загрузке.

> **ПРИМЕЧАНИЕ**
>
> Вам не нужно указывать параметр `COLUMNS`, если имена, количество и порядок столбцов в данных в формате CSV полностью соответствуют таковым в таблице

> Selena.

- **Параллелизм задач**

  Когда есть много partition topic Kafka и достаточно узлов BE, вы можете ускорить загрузку, увеличив параллелизм задач.

  Чтобы увеличить фактический параллелизм подзадач загрузки, вы можете увеличить желаемый параллелизм подзадач загрузки `desired_concurrent_number` при создании задачи routine load. Вы также можете установить динамический параметр конфигурации FE `max_routine_load_task_concurrent_num` (максимальный параллелизм подзадач загрузки по умолчанию) на большее значение. Для получения дополнительной информации о `max_routine_load_task_concurrent_num` см. Параметры конфигурации FE.

  Фактический параллелизм задач определяется минимальным значением среди количества активных узлов BE, количества предварительно указанных partition topic Kafka и значений `desired_concurrent_number` и `max_routine_load_task_concurrent_num`.

  В примере количество активных узлов BE равно `5`, количество предварительно указанных partition topic Kafka равно `5`, а значение `max_routine_load_task_concurrent_num` равно `5`. Чтобы увеличить фактический параллелизм подзадач загрузки, вы можете увеличить `desired_concurrent_number` со значения по умолчанию `3` до `5`.

  Для получения дополнительной информации о свойствах см. CREATE ROUTINE LOAD.

Загрузка данных в формате JSON

В этом разделе описывается, как создать задачу Routine Load для потребления данных в формате JSON из cluster Kafka и загрузки данных в Selena.

Подготовка набора данных

Предположим, что в topic `ordertest2` в cluster Kafka есть набор данных в формате JSON. Набор данных включает шесть ключей: идентификатор товара, имя клиента, национальность, время оплаты и цена. Кроме того, вы хотите преобразовать столбец времени оплаты в тип DATE и загрузить его в столбец `pay_dt` в таблице Selena.

```
{"commodity_id": "1", "customer_name": "Mark Twain", "country":
"US","pay_time": 1589191487,"price": 875}
{"commodity_id": "2", "customer_name": "Oscar Wilde", "country":
"UK","pay_time": 1589191487,"price": 895}
{"commodity_id": "3", "customer_name": "Antoine de Saint-
Exupéry","country": "France","pay_time": 1589191487,"price": 895}
```

> **ВНИМАНИЕ** Каждый JSON-объект в строке должен находиться в одном сообщении Kafka, иначе будет возвращена ошибка парсинга JSON.

Создание таблицы

В соответствии с ключами данных в формате JSON создайте таблицу `example_tbl2` в базе данных `example_db`.

```
CREATE TABLE `example_tbl2` (
    `commodity_id` varchar(26) NULL COMMENT "Commodity ID",
    `customer_name` varchar(26) NULL COMMENT "Customer name",
    `country` varchar(26) NULL COMMENT "Country",
    `pay_time` bigint(20) NULL COMMENT "Payment time",
    `pay_dt` date NULL COMMENT "Payment date",
    `price`double SUM NULL COMMENT "Price"
)
ENGINE=OLAP
AGGREGATE KEY(`commodity_id`,`customer_name`,`country`,`pay_time`,`pay_dt`)
DISTRIBUTED BY HASH(`commodity_id`);
```

> **ВНИМАНИЕ**
>
> Начиная с версии v1.5.2, Selena может автоматически устанавливать количество bucket (BUCKETS) при создании таблицы или добавлении partition. Вам больше не нужно вручную устанавливать количество bucket. Для получения подробной информации см. установка количества bucket.

Отправка задачи Routine Load

Выполните следующий оператор, чтобы отправить задачу Routine Load с именем `example_tbl2_ordertest2` для потребления сообщений из topic `ordertest2` и загрузки данных в таблицу `example_tbl2`. Подзадача загрузки потребляет сообщения с начального смещения в указанных partition topic.

```
CREATE ROUTINE LOAD example_db.example_tbl2_ordertest2 ON example_tbl2
COLUMNS(commodity_id, customer_name, country, pay_time, price,
pay_dt=from_unixtime(pay_time, '%Y%m%d'))
PROPERTIES
(
    "desired_concurrent_number" = "5",
    "format" = "json",
    "jsonpaths" = "
[\"$.commodity_id\",\"$.customer_name\",\"$.country\",\"$.pay_time\",\"$.pric
 )
FROM KAFKA
(
    "kafka_broker_list" ="<kafka_broker1_ip>:<kafka_broker1_port>,
<kafka_broker2_ip>:<kafka_broker2_port>",
    "kafka_topic" = "ordertest2",
    "kafka_partitions" ="0,1,2,3,4",
    "property.kafka_default_offsets" = "OFFSET_BEGINNING"
);
```

После отправки задачи загрузки вы можете выполнить оператор SHOW ROUTINE LOAD, чтобы проверить статус задачи загрузки.

- **Формат данных**

  Вам нужно указать `"format" = "json"` в предложении `PROPERTIES`, чтобы определить, что формат данных — JSON.

- **Сопоставление и преобразование данных**

  Чтобы указать связь сопоставления и преобразования между данными в формате JSON и таблицей Selena, вам нужно указать параметр `COLUMNS` и свойство `jsonpaths`. Порядок полей, указанных в параметре `COLUMNS`, должен соответствовать порядку данных в формате JSON, а имена полей должны соответствовать именам в таблице Selena. Свойство `jsonpaths` используется для извлечения необходимых полей из данных JSON. Затем эти поля именуются свойством `COLUMNS`.

  Поскольку в примере необходимо преобразовать поле времени оплаты в тип данных DATE и загрузить данные в столбец `pay_dt` в таблице Selena, вам нужно использовать функцию from_unixtime. Остальные поля сопоставляются с полями таблицы `example_tbl2` напрямую.

Сопоставление данных:

- Selena извлекает ключи `name` и `code` из данных в формате JSON и сопоставляет их с ключами, объявленными в свойстве `jsonpaths`.
- Selena извлекает ключи, объявленные в свойстве `jsonpaths`, и сопоставляет их **последовательно** с полями, объявленными в параметре `COLUMNS`.
- Selena извлекает поля, объявленные в параметре `COLUMNS`, и сопоставляет их **по имени** со столбцами таблицы Selena.

Преобразование данных:

- Поскольку в примере необходимо преобразовать ключ `pay_time` в тип данных DATE и загрузить данные в столбец `pay_dt` в таблице Selena, вам нужно использовать функцию from_unixtime в параметре `COLUMNS`. Остальные поля сопоставляются с полями таблицы `example_tbl2` напрямую.
- И поскольку пример исключает столбец пола клиента из данных в формате JSON, поле `temp_gender` в параметре `COLUMNS` используется в качестве заполнителя для этого поля. Остальные поля сопоставляются со столбцами таблицы Selena `example_tbl1` напрямую.

  Для получения дополнительной информации о преобразовании данных см. Преобразование данных при загрузке.

  > **ПРИМЕЧАНИЕ**
  >
  > Вам не нужно указывать параметр `COLUMNS`, если имена и количество ключей в JSON-объекте полностью соответствуют именам и количеству полей в таблице Selena.

Загрузка данных в формате Avro

Начиная с версии v1.5.2, Selena поддерживает загрузку данных Avro с помощью Routine Load.

Подготовка набора данных

Схема Avro

1. Создайте следующий файл схемы Avro `avro_schema.avsc`:

```
{
    "type": "record",
    "name": "sensor_log",
    "fields" : [
        {"name": "id", "type": "long"},
        {"name": "name", "type": "string"},
        {"name": "checked", "type" : "boolean"},
        {"name": "data", "type": "double"},
        {"name": "sensor_type", "type": {"type": "enum", "name":
"sensor_type_enum", "symbols" : ["TEMPERATURE", "HUMIDITY", "AIR-
PRESSURE"]}}
    ]
}
```

2. Зарегистрируйте схему Avro в Schema Registry.

Данные Avro

Подготовьте данные Avro и отправьте их в topic Kafka `topic_0`.

Создание таблицы

В соответствии с полями данных Avro создайте таблицу `sensor_log` в целевой базе данных `example_db` в cluster Selena. Имена столбцов таблицы должны соответствовать именам полей в данных Avro. Для сопоставления типов данных между столбцами таблицы и полями данных Avro см. Сопоставление типов данных.

```
CREATE TABLE example_db.sensor_log (
    `id` bigint NOT NULL COMMENT "sensor id",
    `name` varchar(26) NOT NULL COMMENT "sensor name",
    `checked` boolean NOT NULL COMMENT "checked",
    `data` double NULL COMMENT "sensor data",
    `sensor_type` varchar(26) NOT NULL COMMENT "sensor type"
)
ENGINE=OLAP
DUPLICATE KEY (id)
DISTRIBUTED BY HASH(`id`);
```

> **ВНИМАНИЕ**
>
> Начиная с версии v1.5.2, Selena может автоматически устанавливать количество bucket (BUCKETS) при создании таблицы или добавлении partition. Вам больше не нужно вручную устанавливать количество bucket. Для получения подробной информации см. установка количества bucket.

Отправка задачи Routine Load

Выполните следующий оператор, чтобы отправить задачу Routine Load с именем `sensor_log_load_job` для потребления сообщений Avro из topic Kafka `topic_0` и загрузки данных в таблицу `sensor_log` в базе данных `sensor`. Задача загрузки потребляет сообщения с начального смещения в указанных partition topic.

```
CREATE ROUTINE LOAD example_db.sensor_log_load_job ON sensor_log
PROPERTIES
(
    "format" = "avro"
)
FROM KAFKA
(
    "kafka_broker_list" = "<kafka_broker1_ip>:<kafka_broker1_port>,
<kafka_broker2_ip>:<kafka_broker2_port>,...",
    "confluent.schema.registry.url" = "http://172.xx.xxx.xxx:8081",
    "kafka_topic" = "topic_0",
    "kafka_partitions" = "0,1,2,3,4,5",
    "property.kafka_default_offsets" = "OFFSET_BEGINNING"
);
```

- Формат данных

  Вам нужно указать `"format = "avro"` в предложении `PROPERTIES`, чтобы определить, что формат данных — Avro.

- Schema Registry

  Вам нужно настроить `confluent.schema.registry.url`, чтобы указать URL Schema Registry, где зарегистрирована схема Avro. Selena получает схему Avro, используя этот URL. Формат следующий:

```
confluent.schema.registry.url = http[s]://[<schema-registry-api-key>:
<schema-registry-api-secret>@]<hostname|ip address>[:<port>]
```

- Сопоставление и преобразование данных

  Чтобы указать связь сопоставления и преобразования между данными в формате Avro и таблицей Selena, вам нужно указать параметр `COLUMNS` и свойство `jsonpaths`. Порядок полей, указанных в параметре `COLUMNS`, должен соответствовать порядку полей в свойстве `jsonpaths`, а имена полей должны соответствовать именам в таблице Selena. Свойство `jsonpaths` используется для извлечения необходимых полей из данных Avro. Затем эти поля именуются свойством `COLUMNS`.

  Для получения дополнительной информации о преобразовании данных см. Преобразование данных при загрузке.

  > ПРИМЕЧАНИЕ
  >
  > Вам не нужно указывать параметр `COLUMNS`, если имена и количество полей в записи Avro полностью соответствуют именам и количеству столбцов в таблице Selena.

После отправки задачи загрузки вы можете выполнить оператор SHOW ROUTINE LOAD, чтобы проверить статус задачи загрузки.

Сопоставление типов данных

Сопоставление типов данных между полями данных Avro, которые вы хотите загрузить, и столбцами таблицы Selena следующее:

Примитивные типы

Avro	Selena
nul	NULL
boolean	BOOLEAN
int	INT
long	BIGINT
float	FLOAT
double	DOUBLE
bytes	STRING

Avro	Selena
string	STRING

Сложные типы

Avro	Selena
record	Загружает всю RECORD или её подполя в Selena как JSON.
enums	STRING
arrays	ARRAY
maps	JSON
union(T, null)	NULLABLE(T)
fixed	STRING

Ограничения

- В настоящее время Selena не поддерживает эволюцию схемы.
- Каждое сообщение Kafka должно содержать только одну запись данных Avro.

Проверка задачи загрузки и подзадачи

Проверка задачи загрузки

Выполните оператор SHOW ROUTINE LOAD, чтобы проверить статус задачи загрузки `example_tbl2_ordertest2`. Selena возвращает состояние выполнения `State`, статистическую информацию (включая общее количество потреблённых строк и общее количество загруженных строк) `Statistics` и прогресс задачи загрузки `progress`.

Если состояние задачи загрузки автоматически изменилось на **PAUSED**, это возможно потому, что количество ошибочных строк превысило порог. Для получения подробных инструкций по установке этого порога см. CREATE ROUTINE LOAD. Вы можете проверить файлы `ReasonOfStateChanged` и `ErrorLogUrls`, чтобы выявить и устранить проблему. Устранив проблему, вы можете затем выполнить оператор RESUME ROUTINE LOAD, чтобы возобновить приостановленную (**PAUSED**) задачу загрузки.

Если состояние задачи загрузки — **CANCELLED**, возможно, задача загрузки столкнулась с исключением (например, таблица была удалена). Вы можете проверить файлы `ReasonOfStateChanged` и `ErrorLogUrls`, чтобы выявить и устранить проблему. Однако вы не можете возобновить отменённую (**CANCELLED**) задачу загрузки.

```
MySQL [example_db]> SHOW ROUTINE LOAD FOR example_tbl2_ordertest2 \G
*************************** 1. row ***************************
                  Id: 63013
                Name: example_tbl2_ordertest2
          CreateTime: 2022-08-10 17:09:00
           PauseTime: NULL
             EndTime: NULL
              DbName: default_cluster:example_db
           TableName: example_tbl2
               State: RUNNING
      DataSourceType: KAFKA
      CurrentTaskNum: 3
       JobProperties: {"partitions":"*","partial_update":"false","columnToCol
'%Y%m%d'),price","maxBatchIntervalS":"20","whereExpr":"*","dataFormat":"json"
[\"$.commodity_id\",\"$.customer_name\",\"$.country\",\"$.pay_time\",\"$.pric
DataSourceProperties: {"topic":"ordertest2","currentKafkaPartitions":"0,1,2,3
    CustomProperties: {"kafka_default_offsets":"OFFSET_BEGINNING"}
           Statistic: {"receivedBytes":230,"errorRows":0,"committedTaskNum":1
            Progress: {"0":"1","1":"OFFSET_ZERO","2":"OFFSET_ZERO","3":"OFFSE
ReasonOfStateChanged:
        ErrorLogUrls:
            OtherMsg:
```

> **ВНИМАНИЕ**
>
> Вы не можете проверить задачу загрузки, которая остановлена или ещё не запущена.

Проверка подзадачи загрузки

Выполните оператор SHOW ROUTINE LOAD TASK, чтобы проверить подзадачи загрузки задачи загрузки `example_tbl2_ordertest2`, такие как количество текущих выполняемых подзадач, partition topic Kafka, которые потребляются, и прогресс потребления `DataSourceProperties`, а также соответствующий узел Coordinator BE `BeId`.

```
MySQL [example_db]> SHOW ROUTINE LOAD TASK WHERE JobName =
"example_tbl2_ordertest2" \G
*************************** 1. row ***************************
              TaskId: 18c3a823-d73e-4a64-b9cb-b9eced026753
               TxnId: -1
           TxnStatus: UNKNOWN
               JobId: 63013
          CreateTime: 2022-08-10 17:09:05
   LastScheduledTime: 2022-08-10 17:47:27
    ExecuteStartTime: NULL
             Timeout: 60
                BeId: -1
DataSourceProperties: {"1":0,"4":0}
             Message: there is no new data in kafka, wait for 20 seconds to
schedule again
*************************** 2. row ***************************
              TaskId: f76c97ac-26aa-4b41-8194-a8ba2063eb00
               TxnId: -1
           TxnStatus: UNKNOWN
               JobId: 63013
          CreateTime: 2022-08-10 17:09:05
   LastScheduledTime: 2022-08-10 17:47:26
    ExecuteStartTime: NULL
             Timeout: 60
                BeId: -1
DataSourceProperties: {"2":0}
             Message: there is no new data in kafka, wait for 20 seconds to
schedule again
*************************** 3. row ***************************
              TaskId: 1a327a34-99f4-4f8d-8014-3cd38db99ec6
               TxnId: -1
           TxnStatus: UNKNOWN
               JobId: 63013
          CreateTime: 2022-08-10 17:09:26
   LastScheduledTime: 2022-08-10 17:47:27
    ExecuteStartTime: NULL
             Timeout: 60
                BeId: -1
DataSourceProperties: {"0":2,"3":0}
             Message: there is no new data in kafka, wait for 20 seconds to
schedule again
```

Приостановка задачи загрузки

Вы можете выполнить оператор PAUSE ROUTINE LOAD, чтобы приостановить задачу загрузки. Состояние задачи загрузки станет **PAUSED** после выполнения оператора. Однако она не остановлена. Вы можете выполнить оператор RESUME ROUTINE LOAD, чтобы возобновить её. Вы также можете проверить её статус с помощью оператора SHOW ROUTINE LOAD.

Следующий пример приостанавливает задачу загрузки `example_tbl2_ordertest2`:

```
PAUSE ROUTINE LOAD FOR example_tbl2_ordertest2;
```

Возобновление задачи загрузки

Вы можете выполнить оператор RESUME ROUTINE LOAD, чтобы возобновить приостановленную задачу загрузки. Состояние задачи загрузки временно станет **NEED_SCHEDULE** (поскольку задача загрузки перепланируется), а затем станет **RUNNING**. Вы можете проверить её статус с помощью оператора SHOW ROUTINE LOAD.

Следующий пример возобновляет приостановленную задачу загрузки `example_tbl2_ordertest2`:

```
RESUME ROUTINE LOAD FOR example_tbl2_ordertest2;
```

Изменение задачи загрузки

Перед изменением задачи загрузки вы должны приостановить её с помощью оператора PAUSE ROUTINE LOAD. Затем вы можете выполнить ALTER ROUTINE LOAD. После изменения вы можете выполнить оператор RESUME ROUTINE LOAD, чтобы возобновить её, и проверить её статус с помощью оператора SHOW ROUTINE LOAD.

Предположим, что количество активных узлов BE увеличилось до `6`, а partition topic Kafka для потребления — `"0,1,2,3,4,5,6,7"`. Если вы хотите увеличить фактический параллелизм подзадач загрузки, вы можете выполнить следующий оператор, чтобы увеличить количество желаемого параллелизма задач `desired_concurrent_number` до `6` (больше или равно количеству активных узлов BE) и указать partition topic Kafka и начальные смещения.

> **ПРИМЕЧАНИЕ**
>
> Поскольку фактический параллелизм задач определяется минимальным значением нескольких параметров, вы должны убедиться, что значение динамического параметра FE `max_routine_load_task_concurrent_num` больше или равно `6`.

```
ALTER ROUTINE LOAD FOR example_tbl2_ordertest2
PROPERTIES
(
    "desired_concurrent_number" = "6"
)
FROM kafka
(
    "kafka_partitions" = "0,1,2,3,4,5,6,7",
    "kafka_offsets" =
"OFFSET_BEGINNING,OFFSET_BEGINNING,OFFSET_BEGINNING,OFFSET_BEGINNING,OFFSET_E
);
```

Остановка задачи загрузки

Вы можете выполнить оператор STOP ROUTINE LOAD, чтобы остановить задачу загрузки. Состояние задачи загрузки станет **STOPPED** после выполнения оператора, и вы не сможете возобновить остановленную задачу загрузки. Вы не можете проверить статус остановленной задачи загрузки с помощью оператора SHOW ROUTINE LOAD.

Следующий пример останавливает задачу загрузки `example_tbl2_ordertest2`:

```
STOP ROUTINE LOAD FOR example_tbl2_ordertest2;
```

AutoMQ Kafka

AutoMQ for Kafka — это облачная версия Kafka, переработанная для облачных сред. AutoMQ Kafka является открытым исходным кодом и полностью совместима с протоколом Kafka, полностью используя преимущества облака. По сравнению с самостоятельно управляемым Apache Kafka, AutoMQ Kafka с её облачной архитектурой предлагает такие функции, как автоматическое масштабирование ёмкости, самобалансировка сетевого трафика, перемещение партиции за секунды. Эти функции способствуют значительно более низкой совокупной стоимости владения (TCO) для пользователей.

Эта статья поможет вам импортировать данные в AutoMQ Kafka с помощью Selena Routine Load. Для понимания основных принципов Routine Load обратитесь к разделу об основах Routine Load.

Подготовка окружения

Подготовка Selena и тестовых данных

Убедитесь, что у вас есть работающий cluster Selena.

Создание базы данных и таблицы с первичным ключом для тестирования:

```
create database automq_db;
create table users (
  id bigint NOT NULL,
  name string NOT NULL,
  timestamp string NULL,
  status string NULL
) PRIMARY KEY (id)
DISTRIBUTED BY HASH(id)
PROPERTIES (
  "enable_persistent_index" = "true"
);
```

> **ПРИМЕЧАНИЕ**
>
> Если cluster Selena в тестовой среде содержит только один BE, количество replica можно установить равным `1` в предложении `PROPERTIES`, например `PROPERTIES( "replication_num" = "1" )`. Количество replica по умолчанию равно 3, что также является рекомендуемым значением для production-кластеров Selena. Если вы хотите использовать значение по умолчанию, настраивать параметр `replication_num` не нужно.

Подготовка AutoMQ Kafka и тестовых данных

Для подготовки вашего окружения AutoMQ Kafka и тестовых данных следуйте руководству Быстрый старт AutoMQ для развёртывания вашего cluster AutoMQ Kafka. Убедитесь, что Selena может напрямую подключаться к вашему серверу AutoMQ Kafka.

Чтобы быстро создать топик с именем `example_topic` в AutoMQ Kafka и записать в него тестовые JSON-данные, выполните следующие шаги:

Создание топика

Используйте инструменты командной строки Kafka для создания топика. Убедитесь, что у вас есть доступ к окружению Kafka и служба Kafka запущена. Вот команда для создания топика:

```
./kafka-topics.sh --create --topic example_topic --bootstrap-server
10.0.96.4:9092 --partitions 1 --replication-factor 1
```

> Примечание: Замените `topic` и `bootstrap-server` на адрес вашего сервера Kafka.

Чтобы проверить результат создания топика, используйте эту команду:

```
./kafka-topics.sh --describe example_topic --bootstrap-server
10.0.96.4:9092
```

Генерация тестовых данных

Сгенерируйте простые тестовые данные в формате JSON

```
{
  "id": 1,
  "name": "testuser",
  "timestamp": "2023-11-10T12:00:00",
  "status": "active"
}
```

Запись тестовых данных

Используйте инструменты командной строки Kafka или программные методы для записи тестовых данных в example_topic. Вот пример с использованием инструментов командной строки:

```
echo '{"id": 1, "name": "testuser", "timestamp": "2023-11-10T12:00:00",
"status": "active"}' | sh kafka-console-producer.sh --broker-list
10.0.96.4:9092 --topic example_topic
```

> Примечание: Замените `topic` и `bootstrap-server` на адрес вашего сервера Kafka.

Чтобы просмотреть недавно записанные данные топика, используйте следующую команду:

```
sh kafka-console-consumer.sh --bootstrap-server 10.0.96.4:9092 --topic
example_topic --from-beginning
```

Создание задания Routine Load

В командной строке Selena создайте задание Routine Load для непрерывного импорта данных из топика AutoMQ Kafka:

```
CREATE ROUTINE LOAD automq_example_load ON users
COLUMNS(id, name, timestamp, status)
PROPERTIES
(
  "desired_concurrent_number" = "5",
  "format" = "json",
  "jsonpaths" = "[\"$.id\",\"$.name\",\"$.timestamp\",\"$.status\"]"
)
FROM KAFKA
(
  "kafka_broker_list" = "10.0.96.4:9092",
  "kafka_topic" = "example_topic",
  "kafka_partitions" = "0",
  "property.kafka_default_offsets" = "OFFSET_BEGINNING"
);
```

> Примечание: Замените `kafka_broker_list` на адрес вашего сервера Kafka.

Описание параметров

Формат данных

Укажите формат данных как JSON в `"format" = "json"` в предложении PROPERTIES.

Извлечение и преобразование данных

Чтобы указать отношение сопоставления и преобразования между исходными данными и целевой таблицей, настройте параметры COLUMNS и jsonpaths. Имена столбцов в COLUMNS соответствуют именам столбцов целевой таблицы, а их порядок соответствует порядку столбцов в исходных данных. Параметр jsonpaths используется для извлечения необходимых данных полей из JSON-данных, подобно вновь сгенерированным CSV-

данным. Затем параметр COLUMNS временно именует поля в jsonpaths по порядку. Для получения дополнительной информации о преобразовании данных см. Преобразование данных при импорте.

> Примечание: Если каждый JSON-объект на строке имеет имена ключей и количество (порядок не требуется), соответствующие столбцам целевой таблицы, нет необходимости настраивать COLUMNS.

Проверка импорта данных

Сначала мы проверяем задание импорта Routine Load и убеждаемся, что статус задания импорта Routine Load находится в состоянии RUNNING.

```
show routine load\G
```

Затем, выполняя запрос к соответствующей таблице в базе данных Selena, мы можем наблюдать, что данные были успешно импортированы.

```
Selena > select * from users;
+------+--------------+---------------------+--------+
| id   | name         | timestamp           | status |
+------+--------------+---------------------+--------+
|    1 | testuser     | 2023-11-10T12:00:00 | active |
|    2 | testuser     | 2023-11-10T12:00:00 | active |
+------+--------------+---------------------+--------+
2 rows in set (0.01 sec)
```

1.3.4 Потоковая загрузка с Selena Pipe

Преимущества Pipe

Pipe идеально подходит для непрерывной загрузки данных и крупномасштабной загрузки данных:

- **Крупномасштабная загрузка данных микро-пакетами помогает снизить затраты на повторные попытки, вызванные ошибками данных.**

  С помощью Pipe Selena обеспечивает эффективную загрузку большого количества файлов данных со значительным общим объёмом данных. Pipe автоматически

разбивает файлы на основе их количества или размера, разделяя задание загрузки на более мелкие последовательные задачи. Такой подход гарантирует, что ошибки в одном файле не повлияют на всё задание загрузки. Статус загрузки каждого файла записывается Pipe, что позволяет легко идентифицировать и исправлять файлы, содержащие ошибки. Минимизируя необходимость повторных попыток из-за ошибок данных, этот подход помогает снизить затраты.

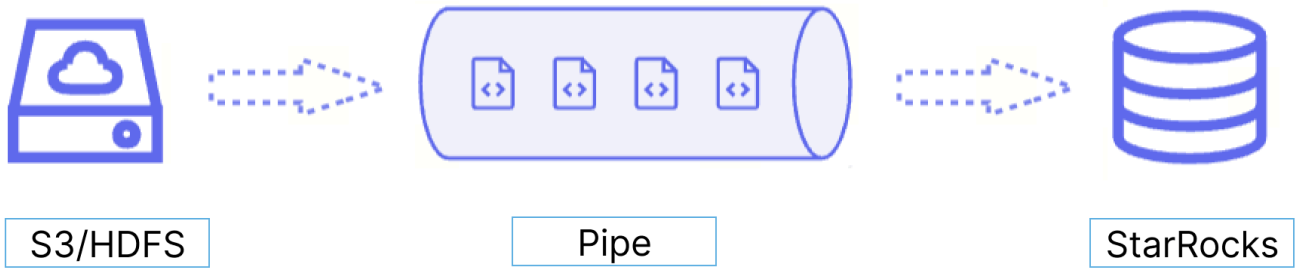
- **Непрерывная загрузка данных помогает сократить трудозатраты.**

  Pipe помогает записывать новые или обновлённые файлы данных в определённое место и непрерывно загружать новые данные из этих файлов в Selena. После создания задания Pipe с указанием `"AUTO_INGEST" = "TRUE"` оно будет постоянно отслеживать изменения файлов данных, хранящихся по указанному пути, и автоматически загружать новые или обновлённые данные из файлов данных в целевую таблицу Selena.

Кроме того, Pipe выполняет проверку уникальности файлов, чтобы предотвратить повторную загрузку данных. В процессе загрузки Pipe проверяет уникальность каждого файла данных на основе имени файла и дайджеста. Если файл с определённым именем и дайджестом уже был обработан заданием Pipe, это задание Pipe пропустит все последующие файлы с тем же именем и дайджестом. Обратите внимание, что объектное хранилище, такое как AWS S3, использует ETag в качестве дайджеста файла, а HDFS использует LastModifiedTime используется как дайджест файла.

Статус загрузки каждого файла данных записывается и сохраняется в представлении `information_schema.pipe_files`. После удаления задания Pipe, связанного с этим представлением, записи о файлах, загруженных в этом задании, также будут удалены.

Поток данных



Загрузка из HDFS

Загрузка данных из HDFS

Selena предоставляет следующие варианты загрузки данных из HDFS:

- Синхронная загрузка с использованием INSERT+`FILES()`
- Асинхронная загрузка с использованием Broker Load
- Непрерывная асинхронная загрузка с использованием Pipe

Каждый из этих вариантов имеет свои преимущества, которые подробно описаны в следующих разделах.

В большинстве случаев мы рекомендуем использовать метод INSERT+`FILES()`, который намного проще в использовании.

Однако метод INSERT+`FILES()` в настоящее время поддерживает только форматы файлов Parquet, ORC и CSV. Поэтому, если вам нужно загружать данные других форматов файлов, таких как JSON, или выполнять изменения данных, такие как DELETE, во время загрузки данных, вы можете использовать Broker Load.

Если вам нужно загрузить большое количество файлов данных со значительным общим объёмом данных (например, более 100 ГБ или даже 1 ТБ), мы рекомендуем использовать метод Pipe. Pipe может разбивать файлы по их количеству или размеру, разделяя задание загрузки на более мелкие последовательные задачи. Такой подход гарантирует, что ошибки в одном файле не повлияют на всё задание загрузки, и минимизирует необходимость повторных попыток из-за ошибок в данных.

Перед началом работы

Подготовка исходных данных

Убедитесь, что исходные данные, которые вы хотите загрузить в Selena, правильно хранятся в вашем cluster HDFS. В этой теме предполагается, что вы хотите загрузить `/user/amber/user_behavior_ten_million_rows.parquet` из HDFS в Selena.

Проверка привилегий

Вы можете загружать данные в таблицы Selena только как пользователь, имеющий привилегию INSERT на эти таблицы Selena. Если у вас нет привилегии INSERT, следуйте инструкциям в разделе GRANT, чтобы предоставить привилегию INSERT пользователю,

которого вы используете для подключения к вашему cluster Selena. Синтаксис: `GRANT INSERT ON TABLE <table_name> IN DATABASE <database_name> TO { ROLE <role_name> | USER <user_identity>}`.

Сбор данных аутентификации

Вы можете использовать простой метод аутентификации для установления соединений с вашим cluster HDFS. Чтобы использовать простую аутентификацию, вам нужно собрать имя пользователя и пароль учётной записи, которую вы можете использовать для доступа к NameNode cluster HDFS.

Использование INSERT+FILES()

Этот метод доступен начиная с версии v1.5.2 и в настоящее время поддерживает только форматы файлов Parquet, ORC и CSV (начиная с версии v1.5.2).

Преимущества INSERT+FILES()

`FILES()` может читать файл, хранящийся в облачном хранилище, на основе указанных вами свойств, связанных с путём, выводить схему таблицы данных в файле, а затем возвращать данные из файла в виде строк данных.

С помощью `FILES()` вы можете:

- Запрашивать данные напрямую из HDFS с помощью SELECT.
- Создавать и загружать таблицу с помощью CREATE TABLE AS SELECT (CTAS).
- Загружать данные в существующую таблицу с помощью INSERT.

Типичные примеры

Прямой запрос из HDFS с помощью SELECT

Прямой запрос из HDFS с помощью SELECT+`FILES()` позволяет получить хороший предварительный просмотр содержимого набора данных перед созданием таблицы. Например:

- Получить предварительный просмотр набора данных без сохранения данных.
- Запросить минимальные и максимальные значения и решить, какие типы данных использовать.
- Проверить наличие значений `NULL`.

Следующий пример запрашивает файл данных `/user/amber/user_behavior_ten_million_rows.parquet`, хранящийся в cluster HDFS:

```
SELECT * FROM FILES
(
    "path" = "hdfs://<hdfs_ip>:
<hdfs_port>/user/amber/user_behavior_ten_million_rows.parquet",
    "format" = "parquet",
    "hadoop.security.authentication" = "simple",
    "username" = "<hdfs_username>",
    "password" = "<hdfs_password>"
)
LIMIT 3;
```

Система возвращает следующий результат запроса:

```
+--------+---------+------------+--------------+---------------------+
| UserID | ItemID  | CategoryID | BehaviorType | Timestamp           |
+--------+---------+------------+--------------+---------------------+
| 543711 |  829192 |    2355072 | pv           | 2017-11-27 08:22:37 |
| 543711 | 2056618 |    3645362 | pv           | 2017-11-27 10:16:46 |
| 543711 | 1165492 |    3645362 | pv           | 2017-11-27 10:17:00 |
+--------+---------+------------+--------------+---------------------+
```

> **ПРИМЕЧАНИЕ**
>
> Обратите внимание, что имена столбцов, возвращённые выше, предоставлены файлом Parquet.

Создание и загрузка таблицы с помощью CTAS

Это продолжение предыдущего примера. Предыдущий запрос обёрнут в CREATE TABLE AS SELECT (CTAS) для автоматизации создания таблицы с использованием вывода схемы. Это означает, что Selena выведет схему таблицы, создаст нужную вам таблицу, а затем загрузит данные в таблицу. Имена и типы столбцов не требуются для создания таблицы при использовании табличной функции `FILES()` с файлами Parquet, поскольку формат Parquet включает имена столбцов.

> **ПРИМЕЧАНИЕ**

> Синтаксис CREATE TABLE при использовании вывода схемы не позволяет устанавливать количество replica, поэтому установите его перед созданием таблицы. Пример ниже для системы с тремя replica:
>
> ```
> ADMIN SET FRONTEND CONFIG ('default_replication_num' = "3");
> ```

Создайте базу данных и переключитесь на неё:

```
CREATE DATABASE IF NOT EXISTS mydatabase;
USE mydatabase;
```

Используйте CTAS для создания таблицы и загрузки данных из файла данных `/user/amber/user_behavior_ten_million_rows.parquet` в таблицу:

```
CREATE TABLE user_behavior_inferred AS
SELECT * FROM FILES
(
    "path" = "hdfs://<hdfs_ip>:
<hdfs_port>/user/amber/user_behavior_ten_million_rows.parquet",
    "format" = "parquet",
    "hadoop.security.authentication" = "simple",
    "username" = "<hdfs_username>",
    "password" = "<hdfs_password>"
);
```

После создания таблицы вы можете просмотреть её схему с помощью DESCRIBE:

```
DESCRIBE user_behavior_inferred;
```

Система возвращает следующий результат запроса:

```
+--------------+-----------+------+-------+---------+-------+
| Field        | Type      | Null | Key   | Default | Extra |
+--------------+-----------+------+-------+---------+-------+
| UserID       | bigint    | YES  | true  | NULL    |       |
| ItemID       | bigint    | YES  | true  | NULL    |       |
| CategoryID   | bigint    | YES  | true  | NULL    |       |
| BehaviorType | varbinary | YES  | false | NULL    |       |
| Timestamp    | varbinary | YES  | false | NULL    |       |
+--------------+-----------+------+-------+---------+-------+
```

Запросите таблицу, чтобы убедиться, что данные были загружены в неё. Пример:

```
SELECT * from user_behavior_inferred LIMIT 3;
```

Возвращается следующий результат запроса, указывающий, что данные были успешно загружены:

```
+--------+--------+------------+--------------+---------------------+
| UserID | ItemID | CategoryID | BehaviorType | Timestamp           |
+--------+--------+------------+--------------+---------------------+
|     84 |  56257 |    1879194 | pv           | 2017-11-26 05:56:23 |
|     84 | 108021 |    2982027 | pv           | 2017-12-02 05:43:00 |
|     84 | 390657 |    1879194 | pv           | 2017-11-28 11:20:30 |
+--------+--------+------------+--------------+---------------------+
```

Загрузка в существующую таблицу с помощью INSERT

Вы можете захотеть настроить таблицу, в которую вставляете данные, например:

- тип данных столбца, настройку nullable или значения по умолчанию
- типы ключей и столбцы
- разбиение данных на partition и bucket

> **ПРИМЕЧАНИЕ**
>
> Создание наиболее эффективной структуры таблицы требует знания того, как данные будут использоваться и содержимого столбцов. Эта тема не охватывает проектирование таблиц. Для получения информации о проектировании таблиц см. Типы таблиц.

В этом примере мы создаём таблицу на основе знаний о том, как таблица будет запрашиваться, и данных в файле Parquet. Знание данных в файле Parquet можно получить, запросив файл напрямую в HDFS.

- Поскольку запрос набора данных в HDFS указывает, что столбец `Timestamp` содержит данные, соответствующие типу данных VARBINARY, тип столбца указан в следующем DDL.
- Запрашивая данные в HDFS, вы можете обнаружить, что в наборе данных нет значений `NULL`, поэтому DDL не устанавливает какие-либо столбцы как nullable.
- На основе знаний об ожидаемых типах запросов sort key и столбец для bucket установлены на столбец `UserID`. Ваш случай использования может отличаться для этих данных, поэтому вы можете решить использовать `ItemID` в дополнение к `UserID` или вместо него для sort key.

Создайте базу данных и переключитесь на неё:

```
CREATE DATABASE IF NOT EXISTS mydatabase;
USE mydatabase;
```

Создайте таблицу вручную (мы рекомендуем, чтобы таблица имела ту же схему, что и файл Parquet, который вы хотите загрузить из HDFS):

```
CREATE TABLE user_behavior_declared
(
    UserID int(11),
    ItemID int(11),
    CategoryID int(11),
    BehaviorType varchar(65533),
    Timestamp varbinary
)
ENGINE = OLAP
DUPLICATE KEY(UserID)
DISTRIBUTED BY HASH(UserID);
```

Отобразите схему, чтобы вы могли сравнить её с выведенной схемой, созданной табличной функцией `FILES()`:

```
DESCRIBE user_behavior_declared;
```

```
+--------------+----------------+------+-------+---------+-------+
| Field        | Type           | Null | Key   | Default | Extra |
+--------------+----------------+------+-------+---------+-------+
| UserID       | int            | NO   | true  | NULL    |       |
| ItemID       | int            | NO   | false | NULL    |       |
| CategoryID   | int            | NO   | false | NULL    |       |
| BehaviorType | varchar(65533) | NO   | false | NULL    |       |
| Timestamp    | varbinary      | NO   | false | NULL    |       |
+--------------+----------------+------+-------+---------+-------+
5 rows in set (0.00 sec)
```

ПОДСКАЗКА

Сравните схему, которую вы только что создали, со схемой, выведенной ранее с помощью табличной функции `FILES()`. Посмотрите на:

- типы данных
- nullable
- ключевые поля

Для лучшего контроля схемы целевой таблицы и лучшей производительности запросов мы рекомендуем указывать схему таблицы вручную в производственных средах.

После создания таблицы вы можете загрузить её с помощью INSERT INTO SELECT FROM FILES():

```
INSERT INTO user_behavior_declared
SELECT * FROM FILES
(
    "path" = "hdfs://<hdfs_ip>:
<hdfs_port>/user/amber/user_behavior_ten_million_rows.parquet",
    "format" = "parquet",
    "hadoop.security.authentication" = "simple",
    "username" = "<hdfs_username>",
    "password" = "<hdfs_password>"
);
```

После завершения загрузки вы можете запросить таблицу, чтобы убедиться, что данные были загружены в неё. Пример:

```
SELECT * from user_behavior_declared LIMIT 3;
```

Возвращается следующий результат запроса, указывающий, что данные были успешно загружены:

```
+--------+---------+------------+--------------+---------------------+
| UserID | ItemID  | CategoryID | BehaviorType | Timestamp           |
+--------+---------+------------+--------------+---------------------+
|    107 | 1568743 |    4476428 | pv           | 2017-11-25 14:29:53 |
|    107 |  470767 |    1020087 | pv           | 2017-11-25 14:32:31 |
|    107 |  358238 |    1817004 | pv           | 2017-11-25 14:43:23 |
+--------+---------+------------+--------------+---------------------+
```

Проверка прогресса загрузки

Вы можете запросить прогресс заданий INSERT из представления `loads` в Information Schema Selena. Эта функция поддерживается начиная с версии v1.5.2. Пример:

```
SELECT * FROM information_schema.loads ORDER BY JOB_ID DESC;
```

Для получения информации о полях, предоставляемых в представлении `loads`, см. `loads`.

Если вы отправили несколько заданий загрузки, вы можете фильтровать по `LABEL`, связанному с заданием. Пример:

```
SELECT * FROM information_schema.loads WHERE LABEL = 'insert_0d86c3f9-851f-11
*************************** 1. row ***************************
              JOB_ID: 10214
               LABEL: insert_0d86c3f9-851f-11ee-9c3e-00163e044958
       DATABASE_NAME: mydatabase
               STATE: FINISHED
            PROGRESS: ETL:100%; LOAD:100%
                TYPE: INSERT
            PRIORITY: NORMAL
           SCAN_ROWS: 10000000
       FILTERED_ROWS: 0
     UNSELECTED_ROWS: 0
           SINK_ROWS: 10000000
            ETL_INFO:
           TASK_INFO: resource:N/A; timeout(s):300; max_filter_ratio:0.0
         CREATE_TIME: 2023-11-17 15:58:14
      ETL_START_TIME: 2023-11-17 15:58:14
     ETL_FINISH_TIME: 2023-11-17 15:58:14
     LOAD_START_TIME: 2023-11-17 15:58:14
    LOAD_FINISH_TIME: 2023-11-17 15:58:18
         JOB_DETAILS: {"All backends":{"0d86c3f9-851f-11ee-9c3e-00163e044958"
[10120]},"FileNumber":0,"FileSize":0,"InternalTableLoadBytes":311710786,"Inte
backends":{"0d86c3f9-851f-11ee-9c3e-00163e044958":[]}}
           ERROR_MSG: NULL
        TRACKING_URL: NULL
        TRACKING_SQL: NULL
REJECTED_RECORD_PATH: NULL
```

> **ПРИМЕЧАНИЕ**
>
> INSERT — это синхронная команда. Если задание INSERT всё ещё выполняется, вам нужно открыть другую сессию, чтобы проверить его статус выполнения.

Использование Broker Load

Асинхронный процесс Broker Load обрабатывает подключение к HDFS, извлечение данных и сохранение данных в Selena.

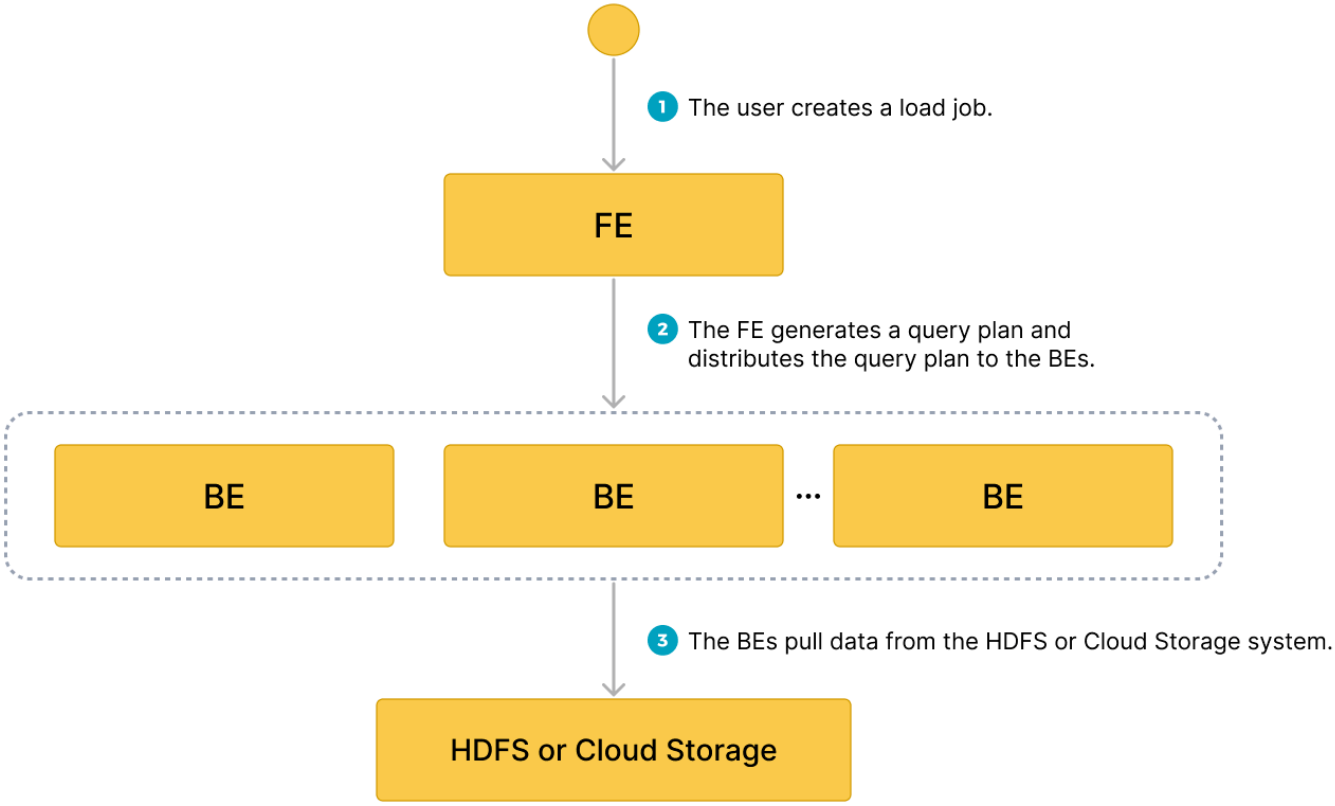
Этот метод поддерживает следующие форматы файлов:

- Parquet
- ORC
- CSV
- JSON (поддерживается начиная с версии v1.5.2)

Преимущества Broker Load

- Broker Load выполняется в фоновом режиме, и клиентам не нужно оставаться подключёнными для продолжения задания.
- Broker Load предпочтителен для длительных заданий, тайм-аут по умолчанию составляет 4 часа.
- В дополнение к форматам файлов Parquet и ORC, Broker Load поддерживает формат файлов CSV и формат файлов JSON (формат файлов JSON поддерживается начиная с версии v1.5.2).

Поток данных



1. Пользователь создаёт задание загрузки.
2. Frontend (FE) создаёт план запроса и распределяет план на backend-узлы (BE) или compute-узлы (CN).
3. BE или CN извлекают данные из источника и загружают данные в Selena.

Типичный пример

Создайте таблицу, запустите процесс загрузки, который извлекает файл данных `/user/amber/user_behavior_ten_million_rows.parquet` из HDFS, и проверьте прогресс и успешность загрузки данных.

Создание базы данных и таблицы

Создайте базу данных и переключитесь на неё:

```
CREATE DATABASE IF NOT EXISTS mydatabase;
USE mydatabase;
```

Создайте таблицу вручную (мы рекомендуем, чтобы таблица имела ту же схему, что и файл Parquet, который вы хотите загрузить из HDFS):

```
CREATE TABLE user_behavior
(
    UserID int(11),
    ItemID int(11),
    CategoryID int(11),
    BehaviorType varchar(65533),
    Timestamp varbinary
)
ENGINE = OLAP
DUPLICATE KEY(UserID)
DISTRIBUTED BY HASH(UserID);
```

Запуск Broker Load

Выполните следующую команду, чтобы запустить задание Broker Load, которое загружает данные из файла данных `/user/amber/user_behavior_ten_million_rows.parquet` в таблицу `user_behavior`:

```
LOAD LABEL user_behavior
(
    DATA INFILE("hdfs://<hdfs_ip>:
<hdfs_port>/user/amber/user_behavior_ten_million_rows.parquet")
    INTO TABLE user_behavior
    FORMAT AS "parquet"
 )
 WITH BROKER
(
    "hadoop.security.authentication" = "simple",
    "username" = "<hdfs_username>",
    "password" = "<hdfs_password>"
)
PROPERTIES
(
    "timeout" = "72000"
);
```

Это задание имеет четыре основных раздела:

- `LABEL`: Строка, используемая при запросе состояния задания загрузки.
- `LOAD` declaration: URI источника, формат исходных данных и имя целевой таблицы.
- `BROKER`: Данные подключения к источнику.
- `PROPERTIES`: Значение тайм-аута и любые другие свойства, применяемые к заданию загрузки.

Подробный синтаксис и описание параметров см. в BROKER LOAD.

Проверка прогресса загрузки

Вы можете запросить прогресс заданий Broker Load из представления `information_schema.loads`. Эта функция поддерживается начиная с версии v1.5.2.

```
SELECT * FROM information_schema.loads;
```

Для получения информации о полях, предоставляемых в представлении `loads`, см. Information Schema).

Если вы отправили несколько заданий загрузки, вы можете фильтровать по `LABEL`, связанному с заданием. Пример:

```
SELECT * FROM information_schema.loads WHERE LABEL = 'user_behavior';
```

В выводе ниже есть две записи для задания загрузки `user_behavior`:

- Первая запись показывает состояние `CANCELLED`. Прокрутите до `ERROR_MSG`, и вы увидите, что задание завершилось неудачей из-за `listPath failed`.
- Вторая запись показывает состояние `FINISHED`, что означает, что задание выполнено успешно.

```
JOB_ID|LABEL                                       |DATABASE_NAME|STATE    |PR
|CREATE_TIME        |ETL_START_TIME     |ETL_FINISH_TIME    |LOAD_START_TIME
|ERROR_MSG                              |TRACKING_URL|TRACKING_SQL|REJECTED_RE
------+----------------------------------------------+-------------+---------+--
--------------------------------------+-------------------+------------------
-----------------------------------------------------------------------------
--------------------------+---------------------------------------+-----------
 10121|user_behavior                                 |mydatabase   |CANCELLED|ET
timeout(s):72000; max_filter_ratio:0.0|2023-08-10 14:59:30|
{},"FileNumber":0,"FileSize":0,"InternalTableLoadBytes":0,"InternalTableLoadR
|type:ETL_RUN_FAIL; msg:listPath failed|             |             |
 10106|user_behavior                                 |mydatabase   |FINISHED |ET
timeout(s):72000; max_filter_ratio:0.0|2023-08-10 14:50:15|2023-08-10 14:50:1
c7348f9a5f2f":
[10004]},"FileNumber":1,"FileSize":1225637388,"InternalTableLoadBytes":271060
backends":{"a5|                                       |             |
```

После того как вы убедитесь, что задание загрузки завершено, вы можете проверить подмножество целевой таблицы, чтобы увидеть, были ли данные успешно загружены. Пример:

```
SELECT * from user_behavior LIMIT 3;
```

Возвращается следующий результат запроса, указывающий, что данные были успешно загружены:

```
+--------+---------+------------+--------------+---------------------+
| UserID | ItemID  | CategoryID | BehaviorType | Timestamp           |
+--------+---------+------------+--------------+---------------------+
|    142 | 2869980 |    2939262 | pv           | 2017-11-25 03:43:22 |
|    142 | 2522236 |    1669167 | pv           | 2017-11-25 15:14:12 |
|    142 | 3031639 |    3607361 | pv           | 2017-11-25 15:19:25 |
+--------+---------+------------+--------------+---------------------+
```

Использование Pipe

Начиная с версии v1.5.2, Selena предоставляет метод загрузки Pipe, который в настоящее время поддерживает только форматы файлов Parquet и ORC.

Преимущества Pipe

Pipe идеально подходит для непрерывной загрузки данных и крупномасштабной загрузки данных:

- **Крупномасштабная загрузка данных микро-пакетами помогает снизить затраты на повторные попытки, вызванные ошибками данных.**

  С помощью Pipe Selena обеспечивает эффективную загрузку большого количества файлов данных со значительным общим объёмом данных. Pipe автоматически разбивает файлы на основе их количества или размера, разделяя задание загрузки на более мелкие последовательные задачи. Такой подход гарантирует, что ошибки в одном файле не повлияют на всё задание загрузки. Статус загрузки каждого файла записывается Pipe, что позволяет легко идентифицировать и исправлять файлы, содержащие ошибки. Минимизируя необходимость повторных попыток из-за ошибок данных, этот подход помогает снизить затраты.

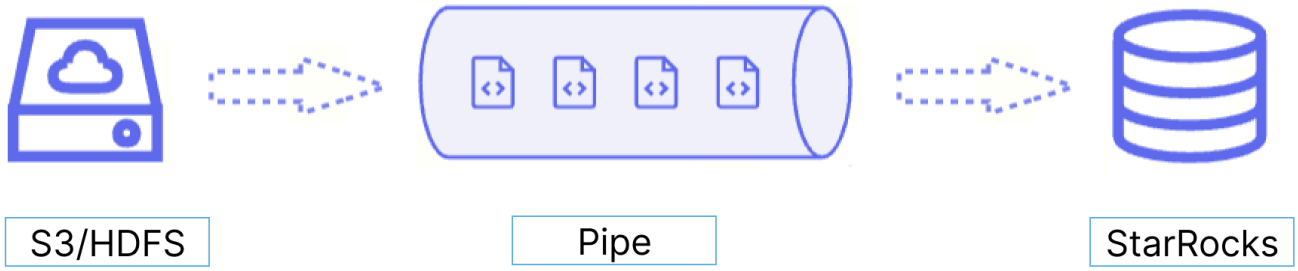
- **Непрерывная загрузка данных помогает сократить трудозатраты.**

  Pipe помогает записывать новые или обновлённые файлы данных в определённое место и непрерывно загружать новые данные из этих файлов в Selena. После создания задания Pipe с указанием `"AUTO_INGEST" = "TRUE"` оно будет постоянно отслеживать изменения файлов данных, хранящихся по указанному пути, и автоматически загружать новые или обновлённые данные из файлов данных в целевую таблицу Selena.

Кроме того, Pipe выполняет проверку уникальности файлов, чтобы предотвратить повторную загрузку данных. В процессе загрузки Pipe проверяет уникальность каждого файла данных на основе имени файла и дайджеста. Если файл с определённым именем и дайджестом уже был обработан заданием Pipe, это задание Pipe пропустит все последующие файлы с тем же именем и дайджестом. Обратите внимание, что HDFS uses LastModifiedTime используется как дайджест файла.

Статус загрузки каждого файла данных записывается и сохраняется в представлении `information_schema.pipe_files`. После удаления задания Pipe, связанного с этим представлением, записи о файлах, загруженных в этом задании, также будут удалены.

Поток данных



Pipe идеально подходит для непрерывной загрузки данных и крупномасштабной загрузки данных:

- **Крупномасштабная загрузка данных микропакетами помогает снизить затраты на повторные попытки, вызванные ошибками данных.**

  С помощью Pipe Selena обеспечивает эффективную загрузку большого количества файлов данных со значительным общим объёмом данных. Pipe автоматически разбивает файлы на основе их количества или размера, разделяя задание загрузки на меньшие последовательные задачи. Этот подход гарантирует, что ошибки в одном файле не повлияют на всё задание загрузки. Статус загрузки каждого файла записывается Pipe, что позволяет легко идентифицировать и исправлять файлы, содержащие ошибки. Минимизируя необходимость повторных попыток из-за ошибок данных, этот подход помогает снизить затраты.

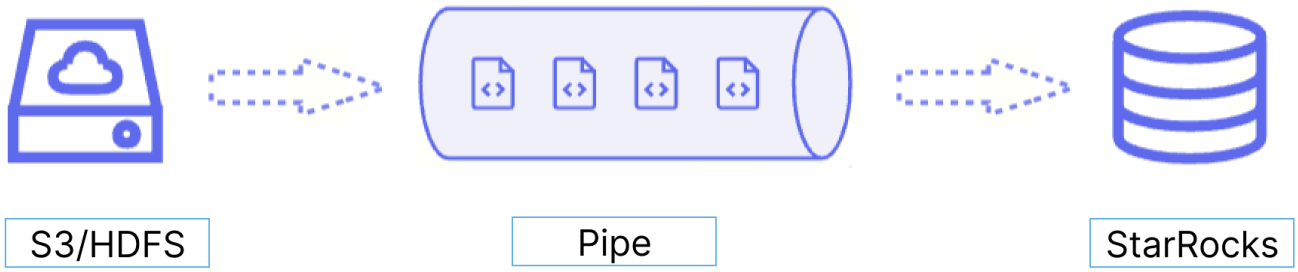
- **Непрерывная загрузка данных помогает сократить трудозатраты.**

Pipe помогает записывать новые или обновлённые файлы данных в определённое место и непрерывно загружать новые данные из этих файлов в Selena. После создания задания Pipe с указанием `"AUTO_INGEST" = "TRUE"` оно будет постоянно отслеживать изменения в файлах данных, хранящихся по указанному пути, и автоматически загружать новые или обновлённые данные из файлов данных в целевую таблицу Selena.

Кроме того, Pipe выполняет проверки уникальности файлов, чтобы предотвратить дублирование загрузки данных. В процессе загрузки Pipe проверяет уникальность каждого файла данных на основе имени файла и дайджеста. Если файл с определённым именем и дайджестом уже был обработан заданием Pipe, задание Pipe пропустит все последующие файлы с тем же именем и дайджестом. Обратите внимание, что HDFS использует `LastModifiedTime` в качестве дайджеста файла.

Статус загрузки каждого файла данных записывается и сохраняется в представлении `information_schema.pipe_files`. После удаления задания Pipe, связанного с представлением, записи о файлах, загруженных в этом задании, также будут удалены.

Поток данных



Различия между Pipe и INSERT+FILES()

Задание Pipe разделяется на одну или несколько транзакций в зависимости от размера и количества строк в каждом файле данных. Пользователи могут запрашивать промежуточные результаты во время процесса загрузки. Напротив, задание INSERT+`FILES()` обрабатывается как одна транзакция, и пользователи не могут просматривать данные во время процесса загрузки.

Последовательность загрузки файлов

Для каждого задания Pipe Selena поддерживает очередь файлов, из которой она извлекает и загружает файлы данных в виде микропакетов. Pipe не гарантирует, что

файлы данных загружаются в том же порядке, в котором они были загружены. Поэтому более новые данные могут быть загружены раньше более старых данных.

Типичный пример

Создание базы данных и таблицы

Создайте базу данных и переключитесь на неё:

```
CREATE DATABASE IF NOT EXISTS mydatabase;
USE mydatabase;
```

Создайте таблицу вручную (мы рекомендуем, чтобы таблица имела ту же схему, что и файл Parquet, который вы хотите загрузить из HDFS):

```
CREATE TABLE user_behavior_replica
(
    UserID int(11),
    ItemID int(11),
    CategoryID int(11),
    BehaviorType varchar(65533),
    Timestamp varbinary
)
ENGINE = OLAP
DUPLICATE KEY(UserID)
DISTRIBUTED BY HASH(UserID);
```

Запуск задания Pipe

Выполните следующую команду, чтобы запустить задание Pipe, которое загружает данные из файла данных `/user/amber/user_behavior_ten_million_rows.parquet` в таблицу `user_behavior_replica`:

```
CREATE PIPE user_behavior_replica
PROPERTIES
(
    "AUTO_INGEST" = "TRUE"
)
AS
INSERT INTO user_behavior_replica
SELECT * FROM FILES
(
    "path" = "hdfs://<hdfs_ip>:
<hdfs_port>/user/amber/user_behavior_ten_million_rows.parquet",
    "format" = "parquet",
    "hadoop.security.authentication" = "simple",
    "username" = "<hdfs_username>",
    "password" = "<hdfs_password>"
);
```

Это задание имеет четыре основных раздела:

- `pipe_name`: Имя pipe. Имя pipe должно быть уникальным в базе данных, к которой принадлежит pipe.
- `INSERT_SQL`: Оператор INSERT INTO SELECT FROM FILES, который используется для загрузки данных из указанного исходного файла данных в целевую таблицу.
- `PROPERTIES`: Набор необязательных параметров, которые определяют, как выполнять pipe. Они включают `AUTO_INGEST`, `POLL_INTERVAL`, `BATCH_SIZE` и `BATCH_FILES`. Укажите эти свойства в формате `"key" = "value"`.

Подробный синтаксис и описание параметров см. в CREATE PIPE.

Проверка прогресса загрузки

- Запросите прогресс заданий Pipe с помощью SHOW PIPES.

  ```
  SHOW PIPES;
  ```

  Если вы отправили несколько заданий загрузки, вы можете фильтровать по `NAME`, связанному с заданием. Пример:

```
SHOW PIPES WHERE NAME = 'user_behavior_replica' \G
*************************** 1. row ***************************
DATABASE_NAME: mydatabase
      PIPE_ID: 10252
    PIPE_NAME: user_behavior_replica
        STATE: RUNNING
   TABLE_NAME: mydatabase.user_behavior_replica
  LOAD_STATUS:
{"loadedFiles":1,"loadedBytes":132251298,"loadingFiles":0,"lastLoadedTime
11-17 16:13:22"}
   LAST_ERROR: NULL
 CREATED_TIME: 2023-11-17 16:13:15
1 row in set (0.00 sec)
```

- Запросите прогресс заданий Pipe из представления `pipes` в Information Schema Selena.

```
SELECT * FROM information_schema.pipes;
```

Если вы отправили несколько заданий загрузки, вы можете фильтровать по `PIPE_NAME`, связанному с заданием. Пример:

```
SELECT * FROM information_schema.pipes WHERE pipe_name = 'user_behavior_r
\G
*************************** 1. row ***************************
DATABASE_NAME: mydatabase
      PIPE_ID: 10252
    PIPE_NAME: user_behavior_replica
        STATE: RUNNING
   TABLE_NAME: mydatabase.user_behavior_replica
  LOAD_STATUS:
{"loadedFiles":1,"loadedBytes":132251298,"loadingFiles":0,"lastLoadedTime
11-17 16:13:22"}
   LAST_ERROR:
 CREATED_TIME: 2023-11-17 16:13:15
1 row in set (0.00 sec)
```

Проверка статуса файлов

Вы можете запросить статус загрузки файлов, загруженных из представления `pipe_files` в Information Schema Selena.

```
SELECT * FROM information_schema.pipe_files;
```

Если вы отправили несколько заданий загрузки, вы можете фильтровать по `PIPE_NAME`, связанному с заданием. Пример:

```
SELECT * FROM information_schema.pipe_files WHERE pipe_name =
'user_behavior_replica' \G
*************************** 1. row ***************************
   DATABASE_NAME: mydatabase
         PIPE_ID: 10252
       PIPE_NAME: user_behavior_replica
       FILE_NAME:
hdfs://172.26.195.67:9000/user/amber/user_behavior_ten_million_rows.parquet
    FILE_VERSION: 1700035418838
       FILE_SIZE: 132251298
   LAST_MODIFIED: 2023-11-15 08:03:38
      LOAD_STATE: FINISHED
     STAGED_TIME: 2023-11-17 16:13:16
 START_LOAD_TIME: 2023-11-17 16:13:17
FINISH_LOAD_TIME: 2023-11-17 16:13:22
       ERROR_MSG:
1 row in set (0.02 sec)
```

Управление Pipe

Вы можете изменять, приостанавливать или возобновлять, удалять или запрашивать созданные вами pipe, а также повторять загрузку определённых файлов данных. Для получения дополнительной информации см. ALTER PIPE, SUSPEND or RESUME PIPE, DROP PIPE, SHOW PIPES и RETRY FILE.

Загрузка из S3

Загрузка данных из AWS S3

Selena предоставляет следующие варианты загрузки данных из AWS S3:

- Синхронная загрузка с использованием INSERT+`FILES()`

- Асинхронная загрузка с использованием Broker Load
- Непрерывная асинхронная загрузка с использованием Pipe

Каждый из этих вариантов имеет свои преимущества, которые подробно описаны в следующих разделах.

В большинстве случаев мы рекомендуем использовать метод INSERT+`FILES()`, который намного проще в использовании.

Однако метод INSERT+`FILES()` в настоящее время поддерживает только форматы файлов Parquet, ORC и CSV. Поэтому, если вам нужно загружать данные других форматов файлов, таких как JSON, или выполнять изменения данных, такие как DELETE, во время загрузки данных, вы можете использовать Broker Load.

Если вам нужно загрузить большое количество файлов данных со значительным общим объёмом данных (например, более 100 ГБ или даже 1 ТБ), мы рекомендуем использовать метод Pipe. Pipe может разбивать файлы по их количеству или размеру, разделяя задание загрузки на более мелкие последовательные задачи. Такой подход гарантирует, что ошибки в одном файле не повлияют на всё задание загрузки, и минимизирует необходимость повторных попыток из-за ошибок в данных.

Перед началом работы

Подготовка исходных данных

Убедитесь, что исходные данные, которые вы хотите загрузить в Selena, правильно хранятся в bucket S3. Вы также можете рассмотреть, где находятся данные и база данных, потому что затраты на передачу данных значительно ниже, когда ваш bucket и ваш cluster Selena находятся в одном регионе.

В этой теме мы предоставляем вам пример набора данных в bucket S3, `s3://selena-examples/user-behavior-10-million-rows.parquet`. Вы можете получить доступ к этому набору данных с любыми действительными учётными данными, так как объект доступен для чтения любому аутентифицированному пользователю AWS.

Проверка привилегий

Вы можете загружать данные в таблицы Selena только как пользователь, имеющий привилегию INSERT на эти таблицы Selena. Если у вас нет привилегии INSERT, следуйте инструкциям в разделе GRANT, чтобы предоставить привилегию INSERT пользователю,

которого вы используете для подключения к вашему cluster Selena. Синтаксис: `GRANT INSERT ON TABLE <table_name> IN DATABASE <database_name> TO { ROLE <role_name> | USER <user_identity>}`.

Сбор данных аутентификации

Примеры в этой теме используют аутентификацию на основе IAM-пользователя. Чтобы убедиться, что у вас есть разрешение на чтение данных из AWS S3, мы рекомендуем прочитать Подготовка к аутентификации на основе IAM-пользователя и следовать инструкциям для создания IAM-пользователя с правильно настроенными политиками IAM.

Вкратце, если вы практикуете аутентификацию на основе IAM-пользователя, вам нужно собрать информацию о следующих ресурсах AWS:

- Bucket S3, в котором хранятся ваши данные.
- Ключ объекта S3 (имя объекта), если вы обращаетесь к определённому объекту в bucket. Обратите внимание, что ключ объекта может включать префикс, если ваши объекты S3 хранятся в подпапках.
- Регион AWS, к которому принадлежит bucket S3.
- Access key и secret key, используемые в качестве учётных данных доступа.

Для получения информации обо всех доступных методах аутентификации см. Аутентификация в ресурсах AWS.

Использование INSERT+FILES()

Этот метод доступен начиная с версии v1.5.2 и в настоящее время поддерживает только форматы файлов Parquet, ORC и CSV (начиная с версии v1.5.2).

Преимущества INSERT+FILES()

`FILES()` может читать файл, хранящийся в облачном хранилище, на основе указанных вами свойств, связанных с путём, выводить схему таблицы данных в файле, а затем возвращать данные из файла в виде строк данных.

С помощью `FILES()` вы можете:

- Запрашивать данные напрямую из S3 с помощью SELECT.
- Создавать и загружать таблицу с помощью CREATE TABLE AS SELECT (CTAS).
- Загружать данные в существующую таблицу с помощью INSERT.

Типичные примеры

Прямой запрос из S3 с помощью SELECT

Прямой запрос из S3 с помощью SELECT+`FILES()` позволяет получить хороший предварительный просмотр содержимого набора данных перед созданием таблицы. Например:

- Получить предварительный просмотр набора данных без сохранения данных.
- Запросить минимальные и максимальные значения и решить, какие типы данных использовать.
- Проверить наличие значений `NULL`.

Следующий пример запрашивает пример набора данных `s3://selena-examples/user-behavior-10-million-rows.parquet`:

```
SELECT * FROM FILES
(
    "path" = "s3://selena-examples/user-behavior-10-million-rows.parquet",
    "format" = "parquet",
    "aws.s3.region" = "us-east-1",
    "aws.s3.access_key" = "AAAAAAAAAAAAAAAAAAAA",
    "aws.s3.secret_key" = "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB"
)
LIMIT 3;
```

> **ПРИМЕЧАНИЕ**
>
> Замените `AAA` и `BBB` на свои учётные данные в приведённой выше команде. Можно использовать любой действительный `aws.s3.access_key` и `aws.s3.secret_key`, так как объект доступен для чтения любому аутентифицированному пользователю AWS.

Система возвращает следующий результат запроса:

```
+--------+---------+------------+--------------+---------------------+
| UserID | ItemID  | CategoryID | BehaviorType | Timestamp           |
+--------+---------+------------+--------------+---------------------+
|      1 | 2576651 |     149192 | pv           | 2017-11-25 01:21:25 |
|      1 | 3830808 |    4181361 | pv           | 2017-11-25 07:04:53 |
|      1 | 4365585 |    2520377 | pv           | 2017-11-25 07:49:06 |
+--------+---------+------------+--------------+---------------------+
```

> **ПРИМЕЧАНИЕ**
>
> Обратите внимание, что имена столбцов, возвращённые выше, предоставлены файлом Parquet.

Создание и загрузка таблицы с помощью CTAS

Это продолжение предыдущего примера. Предыдущий запрос обёрнут в CREATE TABLE AS SELECT (CTAS) для автоматизации создания таблицы с использованием вывода схемы. Это означает, что Selena выведет схему таблицы, создаст нужную вам таблицу, а затем загрузит данные в таблицу. Имена и типы столбцов не требуются для создания таблицы при использовании табличной функции `FILES()` с файлами Parquet, поскольку формат Parquet включает имена столбцов.

> **ПРИМЕЧАНИЕ**
>
> Синтаксис CREATE TABLE при использовании вывода схемы не позволяет устанавливать количество replica, поэтому установите его перед созданием таблицы. Пример ниже для системы с одной replica:
>
> ```
> ADMIN SET FRONTEND CONFIG ('default_replication_num' = "1");
> ```

Создайте базу данных и переключитесь на неё:

```
CREATE DATABASE IF NOT EXISTS mydatabase;
USE mydatabase;
```

Используйте CTAS для создания таблицы и загрузки данных из примера набора данных `s3://selena-examples/user-behavior-10-million-rows.parquet` в таблицу:

```
CREATE TABLE user_behavior_inferred AS
SELECT * FROM FILES
(
    "path" = "s3://selena-examples/user-behavior-10-million-rows.parquet",
    "format" = "parquet",
    "aws.s3.region" = "us-east-1",
    "aws.s3.access_key" = "AAAAAAAAAAAAAAAAAAAA",
    "aws.s3.secret_key" = "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB"
);
```

> **ПРИМЕЧАНИЕ**
>
> Замените `AAA` и `BBB` на свои учётные данные в приведённой выше команде. Можно использовать любой действительный `aws.s3.access_key` и `aws.s3.secret_key`, так как объект доступен для чтения любому аутентифицированному пользователю AWS.

После создания таблицы вы можете просмотреть её схему с помощью DESCRIBE:

```
DESCRIBE user_behavior_inferred;
```

Система возвращает следующий результат запроса:

```
+--------------+------------------+------+-------+---------+-------+
| Field        | Type             | Null | Key   | Default | Extra |
+--------------+------------------+------+-------+---------+-------+
| UserID       | bigint           | YES  | true  | NULL    |       |
| ItemID       | bigint           | YES  | true  | NULL    |       |
| CategoryID   | bigint           | YES  | true  | NULL    |       |
| BehaviorType | varchar(1048576) | YES  | false | NULL    |       |
| Timestamp    | varchar(1048576) | YES  | false | NULL    |       |
+--------------+------------------+------+-------+---------+-------+
```

Запросите таблицу, чтобы убедиться, что данные были загружены в неё. Пример:

```
SELECT * from user_behavior_inferred LIMIT 3;
```

Возвращается следующий результат запроса, указывающий, что данные были успешно загружены:

```
+--------+---------+------------+--------------+---------------------+
| UserID | ItemID  | CategoryID | BehaviorType | Timestamp           |
+--------+---------+------------+--------------+---------------------+
| 225586 | 3694958 |    1040727 | pv           | 2017-12-01 00:58:40 |
| 225586 | 3726324 |     965809 | pv           | 2017-12-01 02:16:02 |
| 225586 | 3732495 |    1488813 | pv           | 2017-12-01 00:59:46 |
+--------+---------+------------+--------------+---------------------+
```

Загрузка в существующую таблицу с помощью INSERT

Вы можете захотеть настроить таблицу, в которую вставляете данные, например:

- тип данных столбца, настройку nullable или значения по умолчанию
- типы ключей и столбцы
- разбиение данных на partition и bucket

> **ПРИМЕЧАНИЕ**
>
> Создание наиболее эффективной структуры таблицы требует знания того, как данные будут использоваться и содержимого столбцов. Эта тема не охватывает проектирование таблиц. Для получения информации о проектировании таблиц см. Типы таблиц.

В этом примере мы создаём таблицу на основе знаний о том, как таблица будет запрашиваться, и данных в файле Parquet. Знание данных в файле Parquet можно получить, запросив файл напрямую в S3.

- Поскольку запрос набора данных в S3 указывает, что столбец `Timestamp` содержит данные, соответствующие типу данных VARCHAR, и Selena может преобразовать VARCHAR в DATETIME, тип данных изменён на DATETIME в следующем DDL.
- Запрашивая данные в S3, вы можете обнаружить, что в наборе данных нет значений `NULL`, поэтому DDL также может установить все столбцы как non-nullable.
- На основе знаний об ожидаемых типах запросов sort key и столбец для bucket установлены на столбец `UserID`. Ваш случай использования может отличаться для этих данных, поэтому вы можете решить использовать `ItemID` в дополнение к `UserID` или вместо него для sort key.

Создайте базу данных и переключитесь на неё:

```
CREATE DATABASE IF NOT EXISTS mydatabase;
USE mydatabase;
```

Создайте таблицу вручную:

```
CREATE TABLE user_behavior_declared
(
    UserID int(11),
    ItemID int(11),
    CategoryID int(11),
    BehaviorType varchar(65533),
    Timestamp datetime
)
ENGINE = OLAP
DUPLICATE KEY(UserID)
DISTRIBUTED BY HASH(UserID);
```

Отобразите схему, чтобы вы могли сравнить её с выведенной схемой, созданной табличной функцией `FILES()`:

```
DESCRIBE user_behavior_declared;
```

```
+--------------+----------------+------+-------+---------+-------+
| Field        | Type           | Null | Key   | Default | Extra |
+--------------+----------------+------+-------+---------+-------+
| UserID       | int            | YES  | true  | NULL    |       |
| ItemID       | int            | YES  | false | NULL    |       |
| CategoryID   | int            | YES  | false | NULL    |       |
| BehaviorType | varchar(65533) | YES  | false | NULL    |       |
| Timestamp    | datetime       | YES  | false | NULL    |       |
+--------------+----------------+------+-------+---------+-------+
```

ПОДСКАЗКА

Сравните схему, которую вы только что создали, со схемой, выведенной ранее с помощью табличной функции `FILES()`. Посмотрите на:

- типы данных
- nullable
- ключевые поля

Для лучшего контроля схемы целевой таблицы и лучшей производительности запросов мы рекомендуем указывать схему таблицы вручную в производственных средах.

После создания таблицы вы можете загрузить её с помощью INSERT INTO SELECT FROM FILES():

```
INSERT INTO user_behavior_declared
SELECT * FROM FILES
(
    "path" = "s3://selena-examples/user-behavior-10-million-rows.parquet",
    "format" = "parquet",
    "aws.s3.region" = "us-east-1",
    "aws.s3.access_key" = "AAAAAAAAAAAAAAAAAAAA",
    "aws.s3.secret_key" = "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB"
);
```

ПРИМЕЧАНИЕ

Замените `AAA` и `BBB` на свои учётные данные в приведённой выше команде. Можно использовать любой действительный `aws.s3.access_key` и `aws.s3.secret_key`, так как объект доступен для чтения любому аутентифицированному пользователю AWS.

После завершения загрузки вы можете запросить таблицу, чтобы убедиться, что данные были загружены в неё. Пример:

```
SELECT * from user_behavior_declared LIMIT 3;
```

Возвращается следующий результат запроса, указывающий, что данные были успешно загружены:

```
+--------+---------+------------+--------------+---------------------+
| UserID | ItemID  | CategoryID | BehaviorType | Timestamp           |
+--------+---------+------------+--------------+---------------------+
| 393529 | 3715112 |     883960 | pv           | 2017-12-02 02:45:44 |
| 393529 | 2650583 |     883960 | pv           | 2017-12-02 02:45:59 |
| 393529 | 3715112 |     883960 | pv           | 2017-12-02 03:00:56 |
+--------+---------+------------+--------------+---------------------+
```

Проверка прогресса загрузки

Вы можете запросить прогресс заданий INSERT из представления `loads` в Information Schema Selena. Эта функция поддерживается начиная с версии v1.5.2. Пример:

```
SELECT * FROM information_schema.loads ORDER BY JOB_ID DESC;
```

Для получения информации о полях, предоставляемых в представлении `loads`, см. `loads`.

Если вы отправили несколько заданий загрузки, вы можете фильтровать по `LABEL`, связанному с заданием. Пример:

```
SELECT * FROM information_schema.loads WHERE LABEL = 'insert_e3b882f5-7eb3-11
*************************** 1. row ***************************
              JOB_ID: 10243
               LABEL: insert_e3b882f5-7eb3-11ee-ae77-00163e267b60
       DATABASE_NAME: mydatabase
               STATE: FINISHED
            PROGRESS: ETL:100%; LOAD:100%
                TYPE: INSERT
            PRIORITY: NORMAL
           SCAN_ROWS: 10000000
       FILTERED_ROWS: 0
     UNSELECTED_ROWS: 0
           SINK_ROWS: 10000000
            ETL_INFO:
           TASK_INFO: resource:N/A; timeout(s):300; max_filter_ratio:0.0
         CREATE_TIME: 2023-11-09 11:56:01
      ETL_START_TIME: 2023-11-09 11:56:01
     ETL_FINISH_TIME: 2023-11-09 11:56:01
     LOAD_START_TIME: 2023-11-09 11:56:01
    LOAD_FINISH_TIME: 2023-11-09 11:56:44
         JOB_DETAILS: {"All backends":{"e3b882f5-7eb3-11ee-ae77-00163e267b60"
[10142]},"FileNumber":0,"FileSize":0,"InternalTableLoadBytes":311710786,"Inte
backends":{"e3b882f5-7eb3-11ee-ae77-00163e267b60":[]}}
           ERROR_MSG: NULL
        TRACKING_URL: NULL
        TRACKING_SQL: NULL
REJECTED_RECORD_PATH: NULL
```

> **ПРИМЕЧАНИЕ**
>
> INSERT — это синхронная команда. Если задание INSERT всё ещё выполняется, вам нужно открыть другую сессию, чтобы проверить его статус выполнения.

Использование Broker Load

Асинхронный процесс Broker Load обрабатывает подключение к S3, извлечение данных и сохранение данных в Selena.

Этот метод поддерживает следующие форматы файлов:

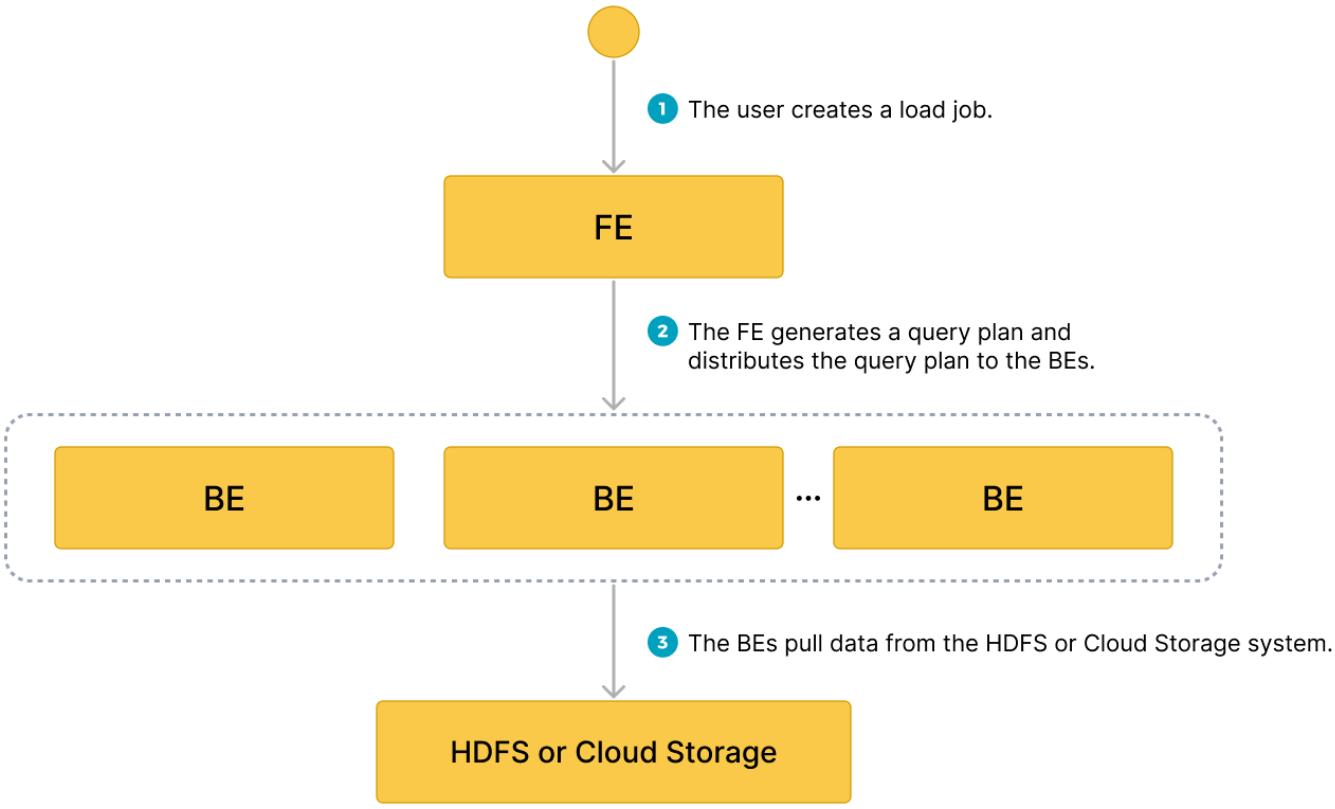
- Parquet
- ORC

- CSV
- JSON (поддерживается начиная с версии v1.5.2)

Преимущества Broker Load

- Broker Load выполняется в фоновом режиме, и клиентам не нужно оставаться подключёнными для продолжения задания.
- Broker Load предпочтителен для длительных заданий, тайм-аут по умолчанию составляет 4 часа.
- В дополнение к форматам файлов Parquet и ORC, Broker Load поддерживает формат файлов CSV и формат файлов JSON (формат файлов JSON поддерживается начиная с версии v1.5.2).

Поток данных



1. Пользователь создаёт задание загрузки.
2. Frontend (FE) создаёт план запроса и распределяет план на backend-узлы (BE) или compute-узлы (CN).
3. BE или CN извлекают данные из источника и загружают данные в Selena.

Типичный пример

Создайте таблицу, запустите процесс загрузки, который извлекает пример набора данных `s3://selena-examples/user-behavior-10-million-rows.parquet` из S3, и проверьте прогресс и успешность загрузки данных.

Создание базы данных и таблицы

Создайте базу данных и переключитесь на неё:

```
CREATE DATABASE IF NOT EXISTS mydatabase;
USE mydatabase;
```

Создайте таблицу вручную (мы рекомендуем, чтобы таблица имела ту же схему, что и файл Parquet, который вы хотите загрузить из AWS S3):

```
CREATE TABLE user_behavior
(
    UserID int(11),
    ItemID int(11),
    CategoryID int(11),
    BehaviorType varchar(65533),
    Timestamp datetime
)
ENGINE = OLAP
DUPLICATE KEY(UserID)
DISTRIBUTED BY HASH(UserID);
```

Запуск Broker Load

Выполните следующую команду, чтобы запустить задание Broker Load, которое загружает данные из примера набора данных `s3://selena-examples/user-behavior-10-million-rows.parquet` в таблицу `user_behavior`:

```
LOAD LABEL user_behavior
(
    DATA INFILE("s3://selena-examples/user-behavior-10-million-
rows.parquet")
    INTO TABLE user_behavior
    FORMAT AS "parquet"
 )
 WITH BROKER
 (
    "aws.s3.enable_ssl" = "true",
    "aws.s3.use_instance_profile" = "false",
    "aws.s3.region" = "us-east-1",
    "aws.s3.access_key" = "AAAAAAAAAAAAAAAAAAAA",
    "aws.s3.secret_key" = "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB"
 )
PROPERTIES
(
    "timeout" = "72000"
);
```

> **ПРИМЕЧАНИЕ**
>
> Замените `AAA` и `BBB` на свои учётные данные в приведённой выше команде. Можно использовать любой действительный `aws.s3.access_key` и `aws.s3.secret_key`, так как объект доступен для чтения любому аутентифицированному пользователю AWS.

Это задание имеет четыре основных раздела:

- `LABEL`: Строка, используемая при запросе состояния задания загрузки.
- `LOAD` declaration: URI источника, формат исходных данных и имя целевой таблицы.
- `BROKER`: Данные подключения к источнику.
- `PROPERTIES`: Значение тайм-аута и любые другие свойства, применяемые к заданию загрузки.

Подробный синтаксис и описание параметров см. в BROKER LOAD.

Проверка прогресса загрузки

Вы можете запросить прогресс задания Broker Load из представления `loads` в Information Schema Selena. Эта функция поддерживается начиная с версии v1.5.2.

```
SELECT * FROM information_schema.loads WHERE LABEL = 'user_behavior';
```

Для получения информации о полях, предоставляемых в представлении `loads`, см. `loads`.

Эта запись показывает состояние `LOADING` и прогресс 39%. Если вы видите что-то подобное, выполните команду ещё раз, пока не увидите состояние `FINISHED`.

```
               JOB_ID: 10466
                LABEL: user_behavior
        DATABASE_NAME: mydatabase
                STATE: LOADING
             PROGRESS: ETL:100%; LOAD:39%
                 TYPE: BROKER
             PRIORITY: NORMAL
            SCAN_ROWS: 4620288
        FILTERED_ROWS: 0
      UNSELECTED_ROWS: 0
            SINK_ROWS: 4620288
             ETL_INFO:
            TASK_INFO: resource:N/A; timeout(s):72000; max_filter_ratio:0.0
          CREATE_TIME: 2024-02-28 22:11:36
       ETL_START_TIME: 2024-02-28 22:11:41
      ETL_FINISH_TIME: 2024-02-28 22:11:41
      LOAD_START_TIME: 2024-02-28 22:11:41
     LOAD_FINISH_TIME: NULL
          JOB_DETAILS: {"All backends":{"2fb97223-b14c-404b-9be1-83aa9b3a7715"
[10004]},"FileNumber":1,"FileSize":136901706,"InternalTableLoadBytes":1440327
backends":{"2fb97223-b14c-404b-9be1-83aa9b3a7715":[10004]}}
            ERROR_MSG: NULL
         TRACKING_URL: NULL
         TRACKING_SQL: NULL
REJECTED_RECORD_PATH: NULL
```

После того как вы убедитесь, что задание загрузки завершено, вы можете проверить подмножество целевой таблицы, чтобы увидеть, были ли данные успешно загружены. Пример:

```
SELECT * from user_behavior LIMIT 3;
```

Возвращается следующий результат запроса, указывающий, что данные были успешно загружены:

```
+--------+---------+------------+--------------+---------------------+
| UserID | ItemID  | CategoryID | BehaviorType | Timestamp           |
+--------+---------+------------+--------------+---------------------+
|     34 |  856384 |    1029459 | pv           | 2017-11-27 14:43:27 |
|     34 | 5079705 |    1029459 | pv           | 2017-11-27 14:44:13 |
|     34 | 4451615 |    1029459 | pv           | 2017-11-27 14:45:52 |
+--------+---------+------------+--------------+---------------------+
```

Использование Pipe

Начиная с версии v1.5.2, Selena предоставляет метод загрузки Pipe, который в настоящее время поддерживает только форматы файлов Parquet и ORC.

Преимущества Pipe

Pipe идеально подходит для непрерывной загрузки данных и крупномасштабной загрузки данных:

- **Крупномасштабная загрузка данных микро-пакетами помогает снизить затраты на повторные попытки, вызванные ошибками данных.**

  С помощью Pipe Selena обеспечивает эффективную загрузку большого количества файлов данных со значительным общим объёмом данных. Pipe автоматически разбивает файлы на основе их количества или размера, разделяя задание загрузки на более мелкие последовательные задачи. Такой подход гарантирует, что ошибки в одном файле не повлияют на всё задание загрузки. Статус загрузки каждого файла записывается Pipe, что позволяет легко идентифицировать и исправлять файлы, содержащие ошибки. Минимизируя необходимость повторных попыток из-за ошибок данных, этот подход помогает снизить затраты.

- **Непрерывная загрузка данных помогает сократить трудозатраты.**

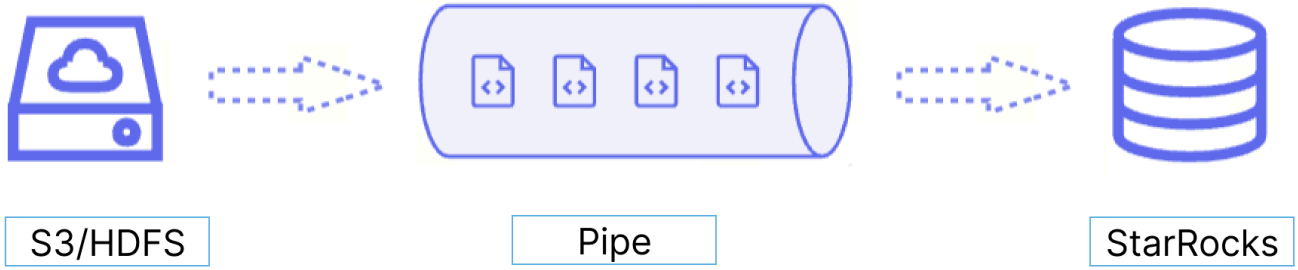
  Pipe помогает записывать новые или обновлённые файлы данных в определённое место и непрерывно загружать новые данные из этих файлов в Selena. После создания задания Pipe с указанием `"AUTO_INGEST" = "TRUE"` оно будет постоянно отслеживать изменения файлов данных, хранящихся по указанному пути, и

автоматически загружать новые или обновлённые данные из файлов данных в целевую таблицу Selena.

Кроме того, Pipe выполняет проверку уникальности файлов, чтобы предотвратить повторную загрузку данных. В процессе загрузки Pipe проверяет уникальность каждого файла данных на основе имени файла и дайджеста. Если файл с определённым именем и дайджестом уже был обработан заданием Pipe, это задание Pipe пропустит все последующие файлы с тем же именем и дайджестом. Обратите внимание, что объектное хранилище, такое как AWS S3, использует ETag используется как дайджест файла.

Статус загрузки каждого файла данных записывается и сохраняется в представлении `information_schema.pipe_files`. После удаления задания Pipe, связанного с этим представлением, записи о файлах, загруженных в этом задании, также будут удалены.

Поток данных



Pipe идеально подходит для непрерывной загрузки данных и крупномасштабной загрузки данных:

- **Крупномасштабная загрузка данных в микропакетах помогает снизить затраты на повторные попытки, вызванные ошибками данных.**

  С помощью Pipe Selena обеспечивает эффективную загрузку большого количества файлов данных со значительным общим объёмом данных. Pipe автоматически разделяет файлы по их количеству или размеру, разбивая задание загрузки на меньшие, последовательные задачи. Этот подход гарантирует, что ошибки в одном файле не влияют на всё задание загрузки. Статус загрузки каждого файла записывается Pipe, позволяя вам легко идентифицировать и исправлять файлы, содержащие ошибки. Минимизируя необходимость повторных попыток из-за ошибок данных, этот подход помогает снизить затраты.

- **Непрерывная загрузка данных помогает сократить трудозатраты.**

  Pipe помогает записывать новые или обновлённые файлы данных в определённое место и непрерывно загружать новые данные из этих файлов в Selena. После создания задания Pipe с указанным `"AUTO_INGEST" = "TRUE"` оно будет постоянно отслеживать изменения в файлах данных, хранящихся по указанному пути, и автоматически загружать новые или обновлённые данные из файлов данных в целевую таблицу Selena.

Кроме того, Pipe выполняет проверки уникальности файлов для предотвращения дублирования загрузки данных. Во время процесса загрузки Pipe проверяет уникальность каждого файла данных на основе имени файла и дайджеста. Если файл с определённым именем и дайджестом уже был обработан заданием Pipe, задание Pipe пропустит все последующие файлы с тем же именем и дайджестом. Обратите внимание, что объектное хранилище, такое как AWS S3, использует `ETag` в качестве дайджеста файла.

Статус загрузки каждого файла данных записывается и сохраняется в представлении `information_schema.pipe_files`. После удаления задания Pipe, связанного с представлением, записи о загруженных в этом задании файлах также будут удалены.

Различия между Pipe и INSERT+FILES()

Задание Pipe разделяется на одну или несколько транзакций в зависимости от размера и количества строк в каждом файле данных. Пользователи могут запрашивать промежуточные результаты во время процесса загрузки. В отличие от этого, задание INSERT+`FILES()` обрабатывается как одна транзакция, и пользователи не могут просматривать данные во время процесса загрузки.

Последовательность загрузки файлов

Для каждого задания Pipe Selena поддерживает очередь файлов, из которой извлекает и загружает файлы данных как микропакеты. Pipe не гарантирует, что файлы данных загружаются в том же порядке, в котором они были загружены. Поэтому более новые данные могут быть загружены раньше более старых данных.

Типичный пример

Создание базы данных и таблицы

Создайте базу данных и переключитесь на неё:

```
CREATE DATABASE IF NOT EXISTS mydatabase;
USE mydatabase;
```

Создайте таблицу вручную (мы рекомендуем, чтобы таблица имела ту же схему, что и файл Parquet, который вы хотите загрузить из AWS S3):

```
CREATE TABLE user_behavior_from_pipe
(
    UserID int(11),
    ItemID int(11),
    CategoryID int(11),
    BehaviorType varchar(65533),
    Timestamp datetime
)
ENGINE = OLAP
DUPLICATE KEY(UserID)
DISTRIBUTED BY HASH(UserID);
```

Запуск задания Pipe

Выполните следующую команду, чтобы запустить задание Pipe, которое загружает данные из примера набора данных `s3://selena-examples/user-behavior-10-million-rows/` в таблицу `user_behavior_from_pipe`. Это задание pipe использует как микропакеты, так и непрерывную загрузку (описанную выше) — специфические функции pipe.

В других примерах в этом руководстве загружается один файл Parquet с 10 миллионами строк. Для примера с pipe тот же набор данных разбит на 57 отдельных файлов, и все они хранятся в одной папке S3. Обратите внимание, что в команде `CREATE PIPE` ниже `path` — это URI для папки S3, и вместо указания имени файла URI заканчивается на `/*`. Установив `AUTO_INGEST` и указав папку вместо отдельного файла, задание pipe будет опрашивать папку S3 на наличие новых файлов и загружать их по мере их добавления в папку.

```
CREATE PIPE user_behavior_pipe
PROPERTIES
(
    "AUTO_INGEST" = "TRUE"
)
AS
INSERT INTO user_behavior_from_pipe
SELECT * FROM FILES
(
    "path" = "s3://selena-examples/user-behavior-10-million-rows/*",
    "format" = "parquet",
    "aws.s3.region" = "us-east-1",
    "aws.s3.access_key" = "AAAAAAAAAAAAAAAAAAAA",
    "aws.s3.secret_key" = "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB"
);
```

> **ПРИМЕЧАНИЕ**
>
> Замените `AAA` и `BBB` на свои учётные данные в приведённой выше команде. Можно использовать любой действительный `aws.s3.access_key` и `aws.s3.secret_key`, так как объект доступен для чтения любому аутентифицированному пользователю AWS.

Это задание имеет четыре основных раздела:

- `pipe_name`: Имя pipe. Имя pipe должно быть уникальным в пределах базы данных, к которой принадлежит pipe.
- `INSERT_SQL`: Оператор INSERT INTO SELECT FROM FILES, который используется для загрузки данных из указанного исходного файла данных в целевую таблицу.
- `PROPERTIES`: Набор необязательных параметров, которые указывают, как выполнять pipe. К ним относятся `AUTO_INGEST`, `POLL_INTERVAL`, `BATCH_SIZE` и `BATCH_FILES`. Указывайте эти свойства в формате `"key" = "value"`.

Подробный синтаксис и описание параметров см. в CREATE PIPE.

Проверка прогресса загрузки

- Запросите прогресс задания Pipe с помощью SHOW PIPES в текущей базе данных, к которой принадлежит задание Pipe.

```
SHOW PIPES WHERE NAME = 'user_behavior_pipe' \G
```

Возвращается следующий результат:

> **ПОДСКАЗКА**
>
> В выводе, показанном ниже, pipe находится в состоянии `RUNNING`. Pipe будет оставаться в состоянии `RUNNING`, пока вы вручную не остановите его. Вывод также показывает количество загруженных файлов (57) и время последней загрузки файла.

```
*************************** 1. row ***************************
DATABASE_NAME: mydatabase
      PIPE_ID: 10476
    PIPE_NAME: user_behavior_pipe
        STATE: RUNNING
   TABLE_NAME: mydatabase.user_behavior_from_pipe
  LOAD_STATUS:
{"loadedFiles":57,"loadedBytes":295345637,"loadingFiles":0,"lastLoadedTim
02-28 22:14:19"}
   LAST_ERROR: NULL
 CREATED_TIME: 2024-02-28 22:13:41
1 row in set (0.02 sec)
```

- Запросите прогресс задания Pipe из представления `pipes` в Information Schema Selena.

```
SELECT * FROM information_schema.pipes WHERE pipe_name =
'user_behavior_replica' \G
```

Возвращается следующий результат:

> **ПОДСКАЗКА**
>
> Некоторые запросы в этом руководстве заканчиваются на `\G` вместо точки с запятой (`;`). Это заставляет клиент MySQL выводить результаты в вертикальном формате. Если вы используете DBeaver или другой клиент, вам может потребоваться использовать точку с запятой (`;`) вместо `\G`.

```
*************************** 1. row ***************************
DATABASE_NAME: mydatabase
      PIPE_ID: 10217
    PIPE_NAME: user_behavior_replica
        STATE: RUNNING
   TABLE_NAME: mydatabase.user_behavior_replica
  LOAD_STATUS:
{"loadedFiles":1,"loadedBytes":132251298,"loadingFiles":0,"lastLoadedTime
11-09 15:35:42"}
   LAST_ERROR:
 CREATED_TIME: 9891-01-15 07:51:45
1 row in set (0.01 sec)
```

Проверка статуса файлов

Вы можете запросить статус загрузки файлов из представления `pipe_files` в Information Schema Selena.

```
SELECT * FROM information_schema.pipe_files WHERE pipe_name =
'user_behavior_replica' \G
```

Возвращается следующий результат:

```
*************************** 1. row ***************************
   DATABASE_NAME: mydatabase
         PIPE_ID: 10217
       PIPE_NAME: user_behavior_replica
       FILE_NAME: s3://selena-examples/user-behavior-10-million-
rows.parquet
    FILE_VERSION: e29daa86b1120fea58ad0d047e671787-8
       FILE_SIZE: 132251298
   LAST_MODIFIED: 2023-11-06 13:25:17
      LOAD_STATE: FINISHED
     STAGED_TIME: 2023-11-09 15:35:02
 START_LOAD_TIME: 2023-11-09 15:35:03
FINISH_LOAD_TIME: 2023-11-09 15:35:42
       ERROR_MSG:
1 row in set (0.03 sec)
```

Управление заданиями Pipe

Вы можете изменять, приостанавливать или возобновлять, удалять или запрашивать созданные вами pipe, а также повторять загрузку определённых файлов данных. Для получения дополнительной информации см. ALTER PIPE, SUSPEND or RESUME PIPE, DROP PIPE, SHOW PIPES и RETRY FILE.

1.3.5 Выгрузка данных из RisingWave в Selena

RisingWave — это распределенная SQL-база данных для потоковой обработки, которая обеспечивает простую, эффективную и надежную обработку потоковых данных. Чтобы быстро начать работу с RisingWave, см. Get started.

RisingWave предоставляет функцию выгрузки данных, которая позволяет пользователям напрямую выгружать данные в Selena без необходимости в каких-либо сторонних компонентах. Эта функция работает со всеми типами таблиц Selena: Duplicate Key, Primary Key, Aggregate и Unique Key.

Предварительные требования

- У вас есть работающий cluster RisingWave версии v1.7 или более поздней.
- У вас есть доступ к целевой таблице Selena, и версия Selena — v1.5.2 или более поздняя.
- Для выгрузки данных в таблицу Selena вы должны иметь привилегии SELECT и INSERT на целевой таблице. Для предоставления привилегий см. GRANT.

> **ПОДСКАЗКА**
>
> RisingWave поддерживает только семантику at-least-once для Selena Sink, что означает, что в случае сбоев могут быть записаны дубликаты данных. Рекомендуется использовать таблицы Selena с Primary Key, которые могут дедуплицировать данные и обеспечить сквозную идемпотентность записи.

Параметры

В следующей таблице описаны параметры, которые необходимо настроить при выгрузке данных из RisingWave в Selena. Все параметры обязательны, если не указано иное.

Параметры	Описание
connector	Установите значение `selena`.
selena.host	IP-адрес FE-ноды Selena.
selena.query_port	Порт запросов FE-ноды.
selena.http_port	HTTP-порт FE-ноды.
selena.user	Имя пользователя для доступа к cluster Selena.
selena.password	Пароль, связанный с именем пользователя.
selena.database	База данных Selena, в которой находится целевая таблица.
selena.table	Таблица Selena, в которую вы хотите выгрузить данные.
selena.partial_update	(Необязательно) Включить ли функцию частичного обновления Selena. Включение этой функции может повысить производительность Sink, когда нужно обновить только несколько столбцов.
type	Тип операции с данными во время выгрузки. • `append-only`: Выполняет только операции INSERT. • `upsert`: Выполняет операции Upsert. Если используется эта настройка, целевая таблица Selena должна быть таблицей с Primary Key.
force_append_only	(Необязательно) Когда для `type` установлено значение `append-only`, но в процессе выгрузки также есть операции Upsert и Delete, эта настройка может заставить задачу Sink генерировать данные только для добавления и отбрасывать данные Upsert и Delete.
primary_key	(Необязательно) Primary key таблицы Selena. Требуется, если `type` имеет значение `upsert`.

Сопоставление типов данных

В следующей таблице приведено сопоставление типов данных между RisingWave и Selena.

RisingWave	Selena
BOOLEAN	BOOLEAN
SMALLINT	SMALLINT

RisingWave	Selena
INTEGER	INT
BIGINT	BIGINT
REAL	FLOAT
DOUBLE	DOUBLE
DECIMAL	DECIMAL
DATE	DATE
VARCHAR	VARCHAR
TIME (Преобразуется в VARCHAR перед выгрузкой в Selena)	Не поддерживается
TIMESTAMP	DATETIME
TIMESTAMP WITH TIME ZONE (Преобразуется в TIMESTAMP перед выгрузкой в Selena)	Не поддерживается
INTERVAL (Преобразуется в VARCHAR перед выгрузкой в Selena)	Не поддерживается
STRUCT	JSON
ARRAY	ARRAY
BYTEA (Преобразуется в VARCHAR перед выгрузкой в Selena)	Не поддерживается
JSONB	JSON
SERIAL	BIGINT

Примеры

1. Создайте базу данных `demo` в Selena и создайте таблицу с Primary Key `score_board` в этой базе данных.

```
CREATE DATABASE demo;
USE demo;

CREATE TABLE demo.score_board(
    id int(11) NOT NULL COMMENT "",
    name varchar(65533) NULL DEFAULT "" COMMENT "",
    score int(11) NOT NULL DEFAULT "0" COMMENT ""
)
PRIMARY KEY(id)
DISTRIBUTED BY HASH(id);
```

2. Выгрузите данные из RisingWave в Selena.

```
-- Создайте таблицу в RisingWave.
CREATE TABLE score_board (
    id INT PRIMARY KEY,
    name VARCHAR,
    score INT
);

-- Вставьте данные в таблицу.
INSERT INTO score_board VALUES (1, 'selena', 100), (2, 'risingwave',
100);

-- Выгрузите данные из этой таблицы в таблицу Selena.
CREATE SINK score_board_sink
FROM score_board WITH (
    connector = 'selena',
    type = 'upsert',
    selena.host = 'selena-fe',
    selena.mysqlport = '9030',
    selena.httpport = '8030',
    selena.user = 'users',
    selena.password = '123456',
    selena.database = 'demo',
    selena.table = 'score_board',
      primary_key = 'id'
);
```

1.4

1.4.1 Selena Spark Connector

1.4.2 Загрузка данных с помощью Spark connector (рекомендуется)

Selena предоставляет самостоятельно разработанный коннектор под названием Selena Connector for Apache Spark™ (сокращённо Spark connector), который помогает загружать данные в таблицу Selena с помощью Spark. Основной принцип заключается в накоплении данных и последующей загрузке всех данных сразу в Selena через STREAM LOAD. Spark connector реализован на основе Spark DataSource V2. DataSource можно создать с помощью Spark DataFrames или Spark SQL. Поддерживаются как пакетный режим, так и режим Structured Streaming.

> **ВНИМАНИЕ**

> Только пользователи с привилегиями SELECT и INSERT на таблицу Selena могут загружать данные в эту таблицу. Вы можете следовать инструкциям в GRANT, чтобы предоставить эти привилегии пользователю.

Требования к версиям

Spark connector	Spark	Selena	Java	Scala
1.1.2	3.2, 3.3, 3.4, 3.5	2.5 и новее	8	2.12
1.1.1	3.2, 3.3 или 3.4	2.5 и новее	8	2.12
1.1.0	3.2, 3.3 или 3.4	2.5 и новее	8	2.12

> **ВНИМАНИЕ**
>
> - См. Обновление Spark connector для получения информации об изменениях поведения между различными версиями Spark connector.
> - Spark connector не предоставляет драйвер MySQL JDBC начиная с версии 1.1.1, и вам нужно вручную импортировать драйвер в classpath Spark. Вы можете найти драйвер на сайте MySQL или Maven Central.

Получение Spark connector

Вы можете получить JAR-файл Spark connector следующими способами:

- Напрямую загрузить скомпилированный JAR-файл Spark Connector.
- Добавить Spark connector как зависимость в ваш Maven-проект и затем загрузить JAR-файл.
- Скомпилировать исходный код Spark Connector в JAR-файл самостоятельно.

Формат именования JAR-файла Spark connector: `selena-spark-connector-${spark_version}_${scala_version}-${connector_version}.jar`.

Например, если вы установили Spark 3.2 и Scala 2.12 в вашей среде и хотите использовать Spark connector 1.1.0, вы можете использовать `selena-spark-connector-3.2_2.12-1.1.0.jar`.

> **ВНИМАНИЕ**
>
> В общем случае последняя версия Spark connector поддерживает совместимость только с тремя самыми последними версиями Spark.

Загрузка скомпилированного JAR-файла

Напрямую загрузите соответствующую версию JAR-файла Spark connector из Maven Central Repository.

Maven-зависимость

1. В файле `pom.xml` вашего Maven-проекта добавьте Spark connector как зависимость в следующем формате. Замените `spark_version`, `scala_version` и `connector_version` соответствующими версиями.

```
<dependency>
<groupId>com.selena</groupId>
<artifactId>selena-spark-connector-${spark_version}_${scala_version}
</artifactId>
<version>${connector_version}</version>
</dependency>
```

2. Например, если версия Spark в вашей среде — 3.2, версия Scala — 2.12, и вы выбираете Spark connector 1.1.0, вам нужно добавить следующую зависимость:

```
<dependency>
<groupId>com.selena</groupId>
<artifactId>selena-spark-connector-3.2_2.12</artifactId>
<version>1.1.0</version>
</dependency>
```

Компиляция самостоятельно

1. Загрузите пакет Spark connector.

2. Выполните следующую команду для компиляции исходного кода Spark connector в JAR-файл. Обратите внимание, что `spark_version` заменяется соответствующей версией Spark.

```
sh build.sh <spark_version>
```

   Например, если версия Spark в вашей среде — 3.2, вам нужно выполнить следующую команду:

```
sh build.sh 3.2
```

3. Перейдите в каталог `target/`, чтобы найти JAR-файл Spark connector, например `selena-spark-connector-3.2_2.12-1.1.0-SNAPSHOT.jar`, сгенерированный при компиляции.

> **ПРИМЕЧАНИЕ**
>
> Имя Spark connector, который не выпущен официально, содержит суффикс `SNAPSHOT`.

Параметры

selena.fe.http.url

Обязательный: ДА
Значение по умолчанию: Нет
Описание: HTTP URL FE в вашем cluster Selena. Вы можете указать несколько URL, которые должны быть разделены запятой (,). Формат: `<fe_host1>:<fe_http_port1>,<fe_host2>:<fe_http_port2>`. Начиная с версии 1.1.1, вы также можете добавить префикс `http://` к URL, например `http://<fe_host1>:<fe_http_port1>,http://<fe_host2>:<fe_http_port2>`.

selena.fe.jdbc.url

Обязательный: ДА
Значение по умолчанию: Нет
Описание: Адрес, используемый для подключения к MySQL-серверу FE. Формат: `jdbc:mysql://<fe_host>:<fe_query_port>`.

selena.table.identifier

Обязательный: ДА
Значение по умолчанию: Нет
Описание: Имя таблицы Selena. Формат: `<database_name>.<table_name>`.

selena.user

Обязательный: ДА
Значение по умолчанию: Нет

Описание: Имя пользователя вашей учётной записи cluster Selena. Пользователю необходимы привилегии SELECT и INSERT на таблицу Selena.

selena.password

Обязательный: ДА
Значение по умолчанию: Нет
Описание: Пароль вашей учётной записи cluster Selena.

selena.write.label.prefix

Обязательный: НЕТ
Значение по умолчанию: spark-
Описание: Префикс метки, используемый Stream Load.

selena.write.enable.transaction-stream-load

Обязательный: НЕТ
Значение по умолчанию: TRUE
Описание: Использовать ли Stream Load transaction interface для загрузки данных. Требуется Selena v1.5.2 или новее. Эта функция может загружать больше данных в транзакции с меньшим использованием памяти и улучшить производительность. **ВНИМАНИЕ:** Начиная с версии 1.1.1, этот параметр действует только когда значение `selena.write.max.retries` не положительное, поскольку Stream Load transaction interface не поддерживает повторы.

selena.write.buffer.size

Обязательный: НЕТ
Значение по умолчанию: 104857600
Описание: Максимальный размер данных, которые могут быть накоплены в памяти перед отправкой в Selena за один раз. Установка этого параметра на большее значение может улучшить производительность загрузки, но может увеличить задержку загрузки.

selena.write.buffer.rows

Обязательный: НЕТ
Значение по умолчанию: Integer.MAX_VALUE
Описание: Поддерживается начиная с версии 1.1.1. Максимальное количество строк, которые могут быть накоплены в памяти перед отправкой в Selena за один раз.

selena.write.flush.interval.ms

Обязательный: НЕТ
Значение по умолчанию: 300000
Описание: Интервал, с которым данные отправляются в Selena. Этот параметр используется для контроля задержки загрузки.

selena.write.max.retries

Обязательный: НЕТ
Значение по умолчанию: 3
Описание: Поддерживается начиная с версии 1.1.1. Количество повторов, которые коннектор выполняет для Stream Load для одного и того же пакета данных, если загрузка не удалась.
ВНИМАНИЕ: Поскольку Stream Load transaction interface не поддерживает повторы. Если этот параметр положительный, коннектор всегда использует интерфейс Stream Load и игнорирует значение `selena.write.enable.transaction-stream-load`.

selena.write.retry.interval.ms

Обязательный: НЕТ
Значение по умолчанию: 10000
Описание: Поддерживается начиная с версии 1.1.1. Интервал для повтора Stream Load для одного и того же пакета данных, если загрузка не удалась.

selena.columns

Обязательный: НЕТ
Значение по умолчанию: Нет
Описание: Столбец таблицы Selena, в который вы хотите загрузить данные. Вы можете указать несколько столбцов, которые должны быть разделены запятыми (,), например `"col0,col1,col2"`.

selena.column.types

Обязательный: НЕТ
Значение по умолчанию: Нет
Описание: Поддерживается начиная с версии 1.1.1. Настройка типов данных столбцов для Spark вместо использования значений по умолчанию, выведенных из таблицы Selena и сопоставления по умолчанию. Значение параметра — это схема в формате DDL, такая

же как вывод Spark StructType#toDDL, например `col0 INT, col1 STRING, col2 BIGINT`. Обратите внимание, что вам нужно указать только столбцы, которые требуют настройки. Один из вариантов использования — загрузка данных в столбцы типа BITMAP или HLL.

selena.write.properties.*

Обязательный: НЕТ
Значение по умолчанию: Нет
Описание: Параметры, используемые для управления поведением Stream Load. Например, параметр `selena.write.properties.format` указывает формат данных для загрузки, например CSV или JSON. Для получения списка поддерживаемых параметров и их описаний см. STREAM LOAD.

selena.write.properties.format

Обязательный: НЕТ
Значение по умолчанию: CSV
Описание: Формат файла, на основе которого Spark connector преобразует каждый пакет данных перед отправкой данных в Selena. Допустимые значения: CSV и JSON.

selena.write.properties.row_delimiter

Обязательный: НЕТ
Значение по умолчанию: \n
Описание: Разделитель строк для данных в формате CSV.

selena.write.properties.column_separator

Обязательный: НЕТ
Значение по умолчанию: \t
Описание: Разделитель столбцов для данных в формате CSV.

selena.write.properties.partial_update

Обязательный: НЕТ
Значение по умолчанию: `FALSE`
Описание: Использовать ли частичные обновления. Допустимые значения: `TRUE` и `FALSE`. Значение по умолчанию: `FALSE`, что означает отключение этой функции.

selena.write.properties.partial_update_mode

Обязательный: НЕТ
Значение по умолчанию: `row`
Описание: Указывает режим для частичных обновлений. Допустимые значения: `row` и `column`.

- Значение `row` (по умолчанию) означает частичные обновления в строчном режиме, который более подходит для обновлений в реальном времени с множеством столбцов и небольшими пакетами.
- Значение `column` означает частичные обновления в колоночном режиме, который более подходит для пакетных обновлений с небольшим количеством столбцов и множеством строк. В таких сценариях включение колоночного режима обеспечивает более высокую скорость обновления. Например, в таблице со 100 столбцами, если обновляются только 10 столбцов (10% от общего количества) для всех строк, скорость обновления в колоночном режиме в 10 раз выше.

selena.write.num.partitions

Обязательный: НЕТ
Значение по умолчанию: Нет
Описание: Количество partition, в которые Spark может записывать данные параллельно. Когда объём данных небольшой, вы можете уменьшить количество partition, чтобы снизить параллелизм и частоту загрузки. Значение по умолчанию для этого параметра определяется Spark. Однако этот метод может вызвать накладные расходы на Spark Shuffle.

selena.write.partition.columns

Обязательный: НЕТ
Значение по умолчанию: Нет
Описание: Столбцы partition в Spark. Параметр действует только когда указан `selena.write.num.partitions`. Если этот параметр не указан, все записываемые столбцы используются для партиционирования.

selena.timezone

Обязательный: НЕТ
Значение по умолчанию: Часовой пояс JVM по умолчанию
Описание: Поддерживается начиная с версии 1.1.1. Часовой пояс, используемый для

преобразования Spark `TimestampType` в Selena `DATETIME`. По умолчанию — часовой пояс JVM, возвращаемый `ZoneId#systemDefault()`. Формат может быть названием часового пояса, например `Asia/Shanghai`, или смещением зоны, например `+08:00`.

Сопоставление типов данных между Spark и Selena

- Сопоставление типов данных по умолчанию следующее:

Тип данных Spark	Тип данных Selena
BooleanType	BOOLEAN
ByteType	TINYINT
ShortType	SMALLINT
IntegerType	INT
LongType	BIGINT
StringType	LARGEINT
FloatType	FLOAT
DoubleType	DOUBLE
DecimalType	DECIMAL
StringType	CHAR
StringType	VARCHAR
StringType	STRING
StringType	JSON
DateType	DATE
TimestampType	DATETIME
ArrayType	ARRAY **ПРИМЕЧАНИЕ:** **Поддерживается начиная с версии 1.1.1**. Для получения подробных шагов см. Загрузка данных в столбцы типа ARRAY.

- Вы также можете настроить сопоставление типов данных.

  Например, таблица Selena содержит столбцы BITMAP и HLL, но Spark не поддерживает эти два типа данных. Вам нужно настроить соответствующие типы данных в Spark. Для получения подробных шагов см. загрузку данных в столбцы BITMAP и HLL. **BITMAP и HLL поддерживаются начиная с версии 1.1.1**.

Обновление Spark connector

Обновление с версии 1.1.0 до 1.1.1

- Начиная с версии 1.1.1, Spark connector не предоставляет `mysql-connector-java`, который является официальным драйвером JDBC для MySQL, из-за ограничений лицензии GPL, используемой `mysql-connector-java`. Однако Spark connector всё ещё нуждается в драйвере MySQL JDBC для подключения к Selena для получения метаданных таблицы, поэтому вам нужно вручную добавить драйвер в classpath Spark. Вы можете найти драйвер на сайте MySQL или Maven Central.
- Начиная с версии 1.1.1, коннектор по умолчанию использует интерфейс Stream Load, а не Stream Load transaction interface, как в версии 1.1.0. Если вы всё ещё хотите использовать Stream Load transaction interface, вы можете установить опцию `selena.write.max.retries` в `0`. См. описание `selena.write.enable.transaction-stream-load` и `selena.write.max.retries` для подробностей.

Примеры

Следующие примеры показывают, как использовать Spark connector для загрузки данных в таблицу Selena с помощью Spark DataFrames или Spark SQL. Spark DataFrames поддерживает как пакетный режим, так и режим Structured Streaming.

Для получения дополнительных примеров см. Примеры Spark Connector.

Подготовка

Создание таблицы Selena

Создайте базу данных `test` и создайте таблицу с Primary Key `score_board`.

```
CREATE DATABASE `test`;

CREATE TABLE `test`.`score_board`
(
    `id` int(11) NOT NULL COMMENT "",
    `name` varchar(65533) NULL DEFAULT "" COMMENT "",
    `score` int(11) NOT NULL DEFAULT "0" COMMENT ""
)
ENGINE=OLAP
PRIMARY KEY(`id`)
COMMENT "OLAP"
DISTRIBUTED BY HASH(`id`);
```

Сетевая конфигурация

Убедитесь, что машина, на которой находится Spark, может получить доступ к узлам FE cluster Selena через `http_port` (по умолчанию: `8030`) и `query_port` (по умолчанию: `9030`), а к узлам BE через `be_http_port` (по умолчанию: `8040`).

Настройка среды Spark

Обратите внимание, что следующие примеры выполняются в Spark 3.2.4 и используют `spark-shell`, `pyspark` и `spark-sql`. Перед запуском примеров убедитесь, что JAR-файл Spark connector размещён в каталоге `$SPARK_HOME/jars`.

Загрузка данных с помощью Spark DataFrames

Следующие два примера объясняют, как загружать данные с помощью пакетного режима Spark DataFrames или режима Structured Streaming.

Пакетный режим

Создайте данные в памяти и загрузите данные в таблицу Selena.

1. Вы можете написать приложение Spark, используя Scala или Python.

Для Scala выполните следующий фрагмент кода в `spark-shell`:

```
// 1. Создайте DataFrame из последовательности.
val data = Seq((1, "selena", 100), (2, "spark", 100))
val df = data.toDF("id", "name", "score")

// 2. Запишите в Selena, настроив формат как "selena" и следующие опции.
// Вам нужно изменить опции в соответствии с вашей средой.
df.write.format("selena")
    .option("selena.fe.http.url", "127.0.0.1:8030")
    .option("selena.fe.jdbc.url", "jdbc:mysql://127.0.0.1:9030")
    .option("selena.table.identifier", "test.score_board")
    .option("selena.user", "root")
    .option("selena.password", "")
    .mode("append")
    .save()
```

Для Python выполните следующий фрагмент кода в `pyspark`:

```python
from pyspark.sql import SparkSession

spark = SparkSession \
     .builder \
     .appName("Selena Example") \
     .getOrCreate()

 # 1. Создайте DataFrame из последовательности.
 data = [(1, "selena", 100), (2, "spark", 100)]
 df = spark.sparkContext.parallelize(data) \
         .toDF(["id", "name", "score"])

 # 2. Запишите в Selena, настроив формат как "selena" и следующие опции.
 # Вам нужно изменить опции в соответствии с вашей средой.
 df.write.format("selena") \
     .option("selena.fe.http.url", "127.0.0.1:8030") \
     .option("selena.fe.jdbc.url", "jdbc:mysql://127.0.0.1:9030") \
     .option("selena.table.identifier", "test.score_board") \
     .option("selena.user", "root") \
     .option("selena.password", "") \
     .mode("append") \
     .save()
```

2. Запросите данные в таблице Selena.

```
MySQL [test]> SELECT * FROM `score_board`;
+------+-----------+-------+
| id   | name      | score |
+------+-----------+-------+
|    1 | selena    |   100 |
|    2 | spark     |   100 |
+------+-----------+-------+
2 rows in set (0.00 sec)
```

Structured Streaming

Создайте потоковое чтение данных из CSV-файла и загрузите данные в таблицу Selena.

1. В каталоге `csv-data` создайте CSV-файл `test.csv` со следующими данными:

```
3,selena,100
4,spark,100
```

2. Вы можете написать приложение Spark, используя Scala или Python.

Для Scala выполните следующий фрагмент кода в `spark-shell`:

```
import org.apache.spark.sql.types.StructType

// 1. Создайте DataFrame из CSV.
val schema = (new StructType()
        .add("id", "integer")
        .add("name", "string")
        .add("score", "integer")
    )
val df = (spark.readStream
        .option("sep", ",")
        .schema(schema)
        .format("csv")
        // Замените на ваш путь к каталогу "csv-data".
        .load("/path/to/csv-data")
    )

// 2. Запишите в Selena, настроив формат как "selena" и следующие опции.
// Вам нужно изменить опции в соответствии с вашей средой.
val query = (df.writeStream.format("selena")
        .option("selena.fe.http.url", "127.0.0.1:8030")
        .option("selena.fe.jdbc.url", "jdbc:mysql://127.0.0.1:9030")
        .option("selena.table.identifier", "test.score_board")
        .option("selena.user", "root")
        .option("selena.password", "")
        // замените на ваш каталог контрольных точек
        .option("checkpointLocation", "/path/to/checkpoint")
        .outputMode("append")
        .start()
    )
```

Для Python выполните следующий фрагмент кода в `pyspark`:

```python
from pyspark.sql import SparkSession
from pyspark.sql.types import IntegerType, StringType, StructType,
StructField

spark = SparkSession \
     .builder \
     .appName("Selena SS Example") \
     .getOrCreate()

 # 1. Создайте DataFrame из CSV.
 schema = StructType([
         StructField("id", IntegerType()),
         StructField("name", StringType()),
         StructField("score", IntegerType())
     ])
 df = (
     spark.readStream
     .option("sep", ",")
     .schema(schema)
     .format("csv")
     # Замените на ваш путь к каталогу "csv-data".
     .load("/path/to/csv-data")
 )

 # 2. Запишите в Selena, настроив формат как "selena" и следующие опции.
 # Вам нужно изменить опции в соответствии с вашей средой.
 query = (
     df.writeStream.format("selena")
     .option("selena.fe.http.url", "127.0.0.1:8030")
     .option("selena.fe.jdbc.url", "jdbc:mysql://127.0.0.1:9030")
     .option("selena.table.identifier", "test.score_board")
     .option("selena.user", "root")
     .option("selena.password", "")
     # замените на ваш каталог контрольных точек
     .option("checkpointLocation", "/path/to/checkpoint")
     .outputMode("append")
     .start()
 )
```

3. Запросите данные в таблице Selena.

```
MySQL [test]> select * from score_board;
+------+-----------+-------+
| id   | name      | score |
+------+-----------+-------+
|    4 | spark     |   100 |
|    3 | selena    |   100 |
+------+-----------+-------+
2 rows in set (0.67 sec)
```

Загрузка данных с помощью Spark SQL

Следующий пример объясняет, как загружать данные с помощью Spark SQL, используя оператор `INSERT INTO` в Spark SQL CLI.

1. Выполните следующий SQL-оператор в `spark-sql`:

```
-- 1. Создайте таблицу, настроив источник данных как `selena` и
следующие опции.
-- Вам нужно изменить опции в соответствии с вашей средой.
CREATE TABLE `score_board`
USING selena
OPTIONS(
"selena.fe.http.url"="127.0.0.1:8030",
"selena.fe.jdbc.url"="jdbc:mysql://127.0.0.1:9030",
"selena.table.identifier"="test.score_board",
"selena.user"="root",
"selena.password"=""
);

-- 2. Вставьте две строки в таблицу.
INSERT INTO `score_board` VALUES (5, "selena", 100), (6, "spark", 100);
```

2. Запросите данные в таблице Selena.

```
MySQL [test]> select * from score_board;
+------+-----------+-------+
| id   | name      | score |
+------+-----------+-------+
|    6 | spark     |   100 |
|    5 | selena    |   100 |
+------+-----------+-------+
2 rows in set (0.00 sec)
```

Лучшие практики

Загрузка данных в таблицу с Primary Key

В этом разделе показано, как загружать данные в таблицу Selena с Primary Key для достижения частичных обновлений и условных обновлений. Вы можете посмотреть Изменение данных через загрузку для подробного введения в эти функции. Эти примеры используют Spark SQL.

Подготовка

Создайте базу данных `test` и создайте таблицу с Primary Key `score_board` в Selena.

```
CREATE DATABASE `test`;

CREATE TABLE `test`.`score_board`
(
    `id` int(11) NOT NULL COMMENT "",
    `name` varchar(65533) NULL DEFAULT "" COMMENT "",
    `score` int(11) NOT NULL DEFAULT "0" COMMENT ""
)
ENGINE=OLAP
PRIMARY KEY(`id`)
COMMENT "OLAP"
DISTRIBUTED BY HASH(`id`);
```

Частичные обновления

Этот пример покажет, как обновить данные только в столбце `name` через загрузку:

1. Вставьте начальные данные в таблицу Selena в клиенте MySQL.

```
mysql> INSERT INTO `score_board` VALUES (1, 'selena', 100), (2,
'spark', 100);

mysql> select * from score_board;
+------+-----------+-------+
| id   | name      | score |
+------+-----------+-------+
|    1 | selena    |   100 |
|    2 | spark     |   100 |
+------+-----------+-------+
2 rows in set (0.02 sec)
```

2. Создайте таблицу Spark `score_board` в клиенте Spark SQL.

   - Установите опцию `selena.write.properties.partial_update` в `true`, что указывает коннектору выполнять частичное обновление.
   - Установите опцию `selena.columns` в `"id,name"`, чтобы указать коннектору, какие столбцы записывать.

```
CREATE TABLE `score_board`
USING selena
OPTIONS(
    "selena.fe.http.url"="127.0.0.1:8030",
    "selena.fe.jdbc.url"="jdbc:mysql://127.0.0.1:9030",
    "selena.table.identifier"="test.score_board",
    "selena.user"="root",
    "selena.password"="",
    "selena.write.properties.partial_update"="true",
    "selena.columns"="id,name"
 );
```

3. Вставьте данные в таблицу в клиенте Spark SQL и обновите только столбец `name`.

```
INSERT INTO `score_board` VALUES (1, 'selena-update'), (2, 'spark-
update');
```

4. Запросите таблицу Selena в клиенте MySQL.

   Вы можете видеть, что изменились только значения для `name`, а значения для `score` не изменились.

```
mysql> select * from score_board;
+------+------------------+-------+
| id   | name             | score |
+------+------------------+-------+
|    1 | selena-update    |   100 |
|    2 | spark-update     |   100 |
+------+------------------+-------+
2 rows in set (0.02 sec)
```

Условные обновления

Этот пример покажет, как выполнять условные обновления в соответствии со значениями столбца `score`. Обновление для `id` вступает в силу только тогда, когда новое значение для `score` больше или равно старому значению.

1. Вставьте начальные данные в таблицу Selena в клиенте MySQL.

```
mysql> INSERT INTO `score_board` VALUES (1, 'selena', 100), (2,
'spark', 100);

mysql> select * from score_board;
+------+-----------+-------+
| id   | name      | score |
+------+-----------+-------+
|    1 | selena    |   100 |
|    2 | spark     |   100 |
+------+-----------+-------+
2 rows in set (0.02 sec)
```

2. Создайте таблицу Spark `score_board` следующим образом.

   - Установите опцию `selena.write.properties.merge_condition` в `score`, что указывает коннектору использовать столбец `score` в качестве условия.
   - Убедитесь, что Spark connector использует интерфейс Stream Load для загрузки данных, а не Stream Load transaction interface, поскольку последний не поддерживает эту функцию.

```
CREATE TABLE `score_board`
USING selena
OPTIONS(
    "selena.fe.http.url"="127.0.0.1:8030",
    "selena.fe.jdbc.url"="jdbc:mysql://127.0.0.1:9030",
    "selena.table.identifier"="test.score_board",
    "selena.user"="root",
    "selena.password"="",
    "selena.write.properties.merge_condition"="score"
 );
```

3. Вставьте данные в таблицу в клиенте Spark SQL и обновите строку с `id` равным 1 с меньшим значением score, а строку с `id` равным 2 с большим значением score.

```
INSERT INTO `score_board` VALUES (1, 'selena-update', 99), (2, 'spark-update', 101);
```

4. Запросите таблицу Selena в клиенте MySQL.

   Вы можете видеть, что изменилась только строка с `id` равным 2, а строка с `id` равным 1 не изменилась.

```
mysql> select * from score_board;
+------+--------------+-------+
| id   | name         | score |
+------+--------------+-------+
|    1 | selena       |   100 |
|    2 | spark-update |   101 |
+------+--------------+-------+
2 rows in set (0.03 sec)
```

Загрузка данных в столбцы типа BITMAP

`BITMAP` часто используется для ускорения подсчёта уникальных значений (count distinct), например подсчёта UV, см. Использование Bitmap для точного Count Distinct. Здесь мы возьмём подсчёт UV в качестве примера, чтобы показать, как загружать данные в столбцы типа `BITMAP`. **`BITMAP` поддерживается начиная с версии 1.1.1**.

1. Создайте Aggregate-таблицу Selena.

В базе данных `test` создайте Aggregate-таблицу `page_uv`, где столбец `visit_users` определён как тип `BITMAP` и настроен с агрегатной функцией `BITMAP_UNION`.

```
CREATE TABLE `test`.`page_uv` (
  `page_id` INT NOT NULL COMMENT 'page ID',
  `visit_date` datetime NOT NULL COMMENT 'access time',
  `visit_users` BITMAP BITMAP_UNION NOT NULL COMMENT 'user ID'
) ENGINE=OLAP
AGGREGATE KEY(`page_id`, `visit_date`)
DISTRIBUTED BY HASH(`page_id`);
```

2. Создайте таблицу Spark.

   Схема таблицы Spark выводится из таблицы Selena, и Spark не поддерживает тип `BITMAP`. Поэтому вам нужно настроить соответствующий тип данных столбца в Spark, например как `BIGINT`, настроив опцию `"selena.column.types"="visit_users BIGINT"`. При использовании Stream Load для загрузки данных коннектор использует функцию `to_bitmap` для преобразования данных типа `BIGINT` в тип `BITMAP`.

   Выполните следующий DDL в `spark-sql`:

```
CREATE TABLE `page_uv`
USING selena
OPTIONS(
   "selena.fe.http.url"="127.0.0.1:8030",
   "selena.fe.jdbc.url"="jdbc:mysql://127.0.0.1:9030",
   "selena.table.identifier"="test.page_uv",
   "selena.user"="root",
   "selena.password"="",
   "selena.column.types"="visit_users BIGINT"
);
```

3. Загрузите данные в таблицу Selena.

   Выполните следующий DML в `spark-sql`:

```
INSERT INTO `page_uv` VALUES
   (1, CAST('2020-06-23 01:30:30' AS TIMESTAMP), 13),
   (1, CAST('2020-06-23 01:30:30' AS TIMESTAMP), 23),
   (1, CAST('2020-06-23 01:30:30' AS TIMESTAMP), 33),
   (1, CAST('2020-06-23 02:30:30' AS TIMESTAMP), 13),
   (2, CAST('2020-06-23 01:30:30' AS TIMESTAMP), 23);
```

4. Вычислите UV страниц из таблицы Selena.

```
MySQL [test]> SELECT `page_id`, COUNT(DISTINCT `visit_users`) FROM
`page_uv` GROUP BY `page_id`;
+---------+-----------------------------+
| page_id | count(DISTINCT visit_users) |
+---------+-----------------------------+
|       2 |                           1 |
|       1 |                           3 |
+---------+-----------------------------+
2 rows in set (0.01 sec)
```

> **ВНИМАНИЕ:**
>
> Коннектор использует функцию `to_bitmap` для преобразования данных типов `TINYINT`, `SMALLINT`, `INTEGER` и `BIGINT` в Spark в тип `BITMAP` в Selena, и использует функцию `bitmap_hash` или `bitmap_hash64` для других типов данных Spark.

Загрузка данных в столбцы типа HLL

`HLL` можно использовать для приблизительного подсчёта уникальных значений (count distinct), см. Использование HLL для приблизительного count distinct.

Здесь мы возьмём подсчёт UV в качестве примера, чтобы показать, как загружать данные в столбцы типа `HLL`. **`HLL` поддерживается начиная с версии 1.1.1**.

1. Создайте Aggregate-таблицу Selena.

   В базе данных `test` создайте Aggregate-таблицу `hll_uv`, где столбец `visit_users` определён как тип `HLL` и настроен с агрегатной функцией `HLL_UNION`.

```
CREATE TABLE `hll_uv` (
`page_id` INT NOT NULL COMMENT 'page ID',
`visit_date` datetime NOT NULL COMMENT 'access time',
`visit_users` HLL HLL_UNION NOT NULL COMMENT 'user ID'
) ENGINE=OLAP
AGGREGATE KEY(`page_id`, `visit_date`)
DISTRIBUTED BY HASH(`page_id`);
```

2. Создайте таблицу Spark.

   Схема таблицы Spark выводится из таблицы Selena, и Spark не поддерживает тип `HLL`. Поэтому вам нужно настроить соответствующий тип данных столбца в Spark, например как `BIGINT`, настроив опцию `"selena.column.types"="visit_users BIGINT"`. При использовании Stream Load для загрузки данных коннектор использует функцию `hll_hash` для преобразования данных типа `BIGINT` в тип `HLL`.

   Выполните следующий DDL в `spark-sql`:

```
CREATE TABLE `hll_uv`
USING selena
OPTIONS(
   "selena.fe.http.url"="127.0.0.1:8030",
   "selena.fe.jdbc.url"="jdbc:mysql://127.0.0.1:9030",
   "selena.table.identifier"="test.hll_uv",
   "selena.user"="root",
   "selena.password"="",
   "selena.column.types"="visit_users BIGINT"
);
```

3. Загрузите данные в таблицу Selena.

   Выполните следующий DML в `spark-sql`:

```
INSERT INTO `hll_uv` VALUES
   (3, CAST('2023-07-24 12:00:00' AS TIMESTAMP), 78),
   (4, CAST('2023-07-24 13:20:10' AS TIMESTAMP), 2),
   (3, CAST('2023-07-24 12:30:00' AS TIMESTAMP), 674);
```

4. Вычислите UV страниц из таблицы Selena.

```
MySQL [test]> SELECT `page_id`, COUNT(DISTINCT `visit_users`) FROM
`hll_uv` GROUP BY `page_id`;
+---------+----------------------------+
| page_id | count(DISTINCT visit_users) |
+---------+----------------------------+
|       4 |                          1 |
|       3 |                          2 |
+---------+----------------------------+
2 rows in set (0.01 sec)
```

Загрузка данных в столбцы типа ARRAY

Следующий пример объясняет, как загружать данные в столбцы типа `ARRAY`.

1. Создайте таблицу Selena.

   В базе данных `test` создайте таблицу с Primary Key `array_tbl`, которая включает один столбец `INT` и два столбца `ARRAY`.

```
CREATE TABLE `array_tbl` (
    `id` INT NOT NULL,
    `a0` ARRAY<STRING>,
    `a1` ARRAY<ARRAY<INT>>
 )
 ENGINE=OLAP
 PRIMARY KEY(`id`)
 DISTRIBUTED BY HASH(`id`)
 ;
```

2. Запишите данные в Selena.

   Поскольку некоторые версии Selena не предоставляют метаданные столбца `ARRAY`, коннектор не может вывести соответствующий тип данных Spark для этого столбца. Однако вы можете явно указать соответствующий тип данных Spark столбца в опции `selena.column.types`. В этом примере вы можете настроить опцию как `a0 ARRAY<STRING>,a1 ARRAY<ARRAY<INT>>`.

   Выполните следующий код в `spark-shell`:

```
 val data = Seq(
    |  (1, Seq("hello", "selena"), Seq(Seq(1, 2), Seq(3, 4))),
    |  (2, Seq("hello", "spark"), Seq(Seq(5, 6, 7), Seq(8, 9, 10)))
    | )
 val df = data.toDF("id", "a0", "a1")
 df.write
      .format("selena")
      .option("selena.fe.http.url", "127.0.0.1:8030")
      .option("selena.fe.jdbc.url", "jdbc:mysql://127.0.0.1:9030")
      .option("selena.table.identifier", "test.array_tbl")
      .option("selena.user", "root")
      .option("selena.password", "")
      .option("selena.column.types", "a0 ARRAY<STRING>,a1
ARRAY<ARRAY<INT>>")
      .mode("append")
      .save()
```

3. Запросите данные в таблице Selena.

```
MySQL [test]> SELECT * FROM `array_tbl`;
+------+----------------------+--------------------+
| id   | a0                   | a1                 |
+------+----------------------+--------------------+
|    1 | ["hello","selena"]   | [[1,2],[3,4]]      |
|    2 | ["hello","spark"]    | [[5,6,7],[8,9,10]] |
+------+----------------------+--------------------+
2 rows in set (0.01 sec)
```

1.4.3 Apache Spark Load

1.4.4 Массовая загрузка данных с помощью Spark Load

Этот метод загрузки использует внешние ресурсы Apache Spark™ для предварительной обработки импортируемых данных, что улучшает производительность импорта и экономит вычислительные ресурсы. Он в основном используется для **начальной миграции** и **импорта больших объёмов данных** в Selena (объём данных до уровня ТБ).

Spark load — это **асинхронный** метод импорта, который требует от пользователей создания задач импорта типа Spark через протокол MySQL и просмотра результатов импорта с помощью `SHOW LOAD`.

> **ВНИМАНИЕ**
>
> - Только пользователи с привилегией INSERT на таблицу Selena могут загружать данные в эту таблицу. Вы можете следовать инструкциям в GRANT, чтобы предоставить необходимую привилегию.
> - Spark Load нельзя использовать для загрузки данных в таблицу с Primary Key.

Объяснение терминологии

- **Spark ETL**: Отвечает главным образом за ETL данных в процессе импорта, включая построение глобального словаря (тип BITMAP), партиционирование, сортировку, агрегацию и т.д.
- **Broker**: Broker — это независимый процесс без состояния. Он инкапсулирует интерфейс файловой системы и предоставляет Selena возможность читать файлы из удалённых систем хранения.
- **Глобальный словарь**: Сохраняет структуру данных, которая сопоставляет данные от исходного значения к закодированному значению. Исходное значение может быть любого типа данных, в то время как закодированное значение является целым числом. Глобальный словарь в основном используется в сценариях, где предварительно вычисляется точный count distinct.

Основы

Пользователь отправляет задачу импорта типа Spark через MySQL-клиент; FE записывает метаданные и возвращает результат отправки.

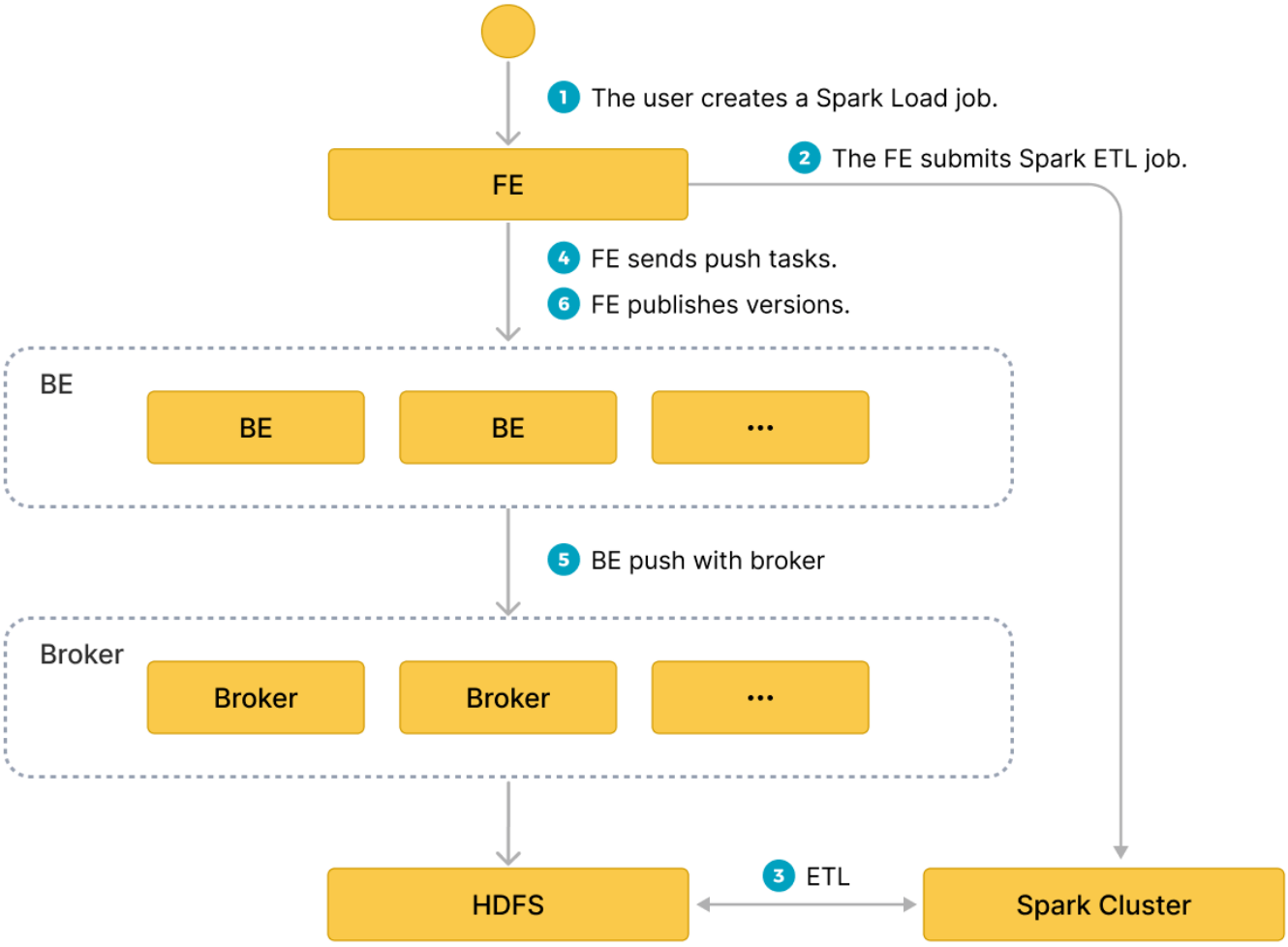
Выполнение задачи Spark load делится на следующие основные фазы.

1. Пользователь отправляет задачу Spark load в FE.
2. FE планирует отправку задачи ETL в cluster Apache Spark™ для выполнения.
3. Cluster Apache Spark™ выполняет задачу ETL, которая включает построение глобального словаря (тип BITMAP), партиционирование, сортировку, агрегацию и т.д.
4. После завершения задачи ETL FE получает путь к данным каждого предварительно обработанного среза и планирует соответствующие BE для выполнения задачи Push.
5. BE читает данные через процесс Broker из HDFS и конвертирует их в формат хранения Selena.

   > Если вы решите не использовать процесс Broker, BE читает данные из HDFS напрямую.

6. FE планирует эффективную версию и завершает задачу импорта.

Следующая диаграмма иллюстрирует основной поток Spark load.



Глобальный словарь

Применимые сценарии

В настоящее время столбец BITMAP в Selena реализован с использованием Roaringbitmap, который принимает только целые числа в качестве типа входных данных. Поэтому, если вы хотите реализовать предварительное вычисление для столбца BITMAP в процессе импорта, вам нужно преобразовать тип входных данных в целое число.

В существующем процессе импорта Selena структура данных глобального словаря реализована на основе таблицы Hive, которая сохраняет сопоставление от исходного значения к закодированному значению.

Процесс построения

1. Чтение данных из вышестоящего источника данных и генерация временной таблицы Hive с именем `hive-table`.
2. Извлечение значений дедуплицируемых полей из `hive-table` для генерации новой таблицы Hive с именем `distinct-value-table`.
3. Создание новой таблицы глобального словаря с именем `dict-table` с одним столбцом для исходных значений и одним столбцом для закодированных значений.
4. Left join между `distinct-value-table` и `dict-table`, а затем использование оконной функции для кодирования этого набора. Наконец, как исходное значение, так и закодированное значение дедуплицированного столбца записываются обратно в `dict-table`.
5. Join между `dict-table` и `hive-table` для завершения работы по замене исходного значения в `hive-table` на целочисленное закодированное значение.
6. `hive-table` будет прочитана при следующей предварительной обработке данных, а затем импортирована в Selena после вычисления.

Предварительная обработка данных

Базовый процесс предварительной обработки данных следующий:

1. Чтение данных из вышестоящего источника данных (файл HDFS или таблица Hive).
2. Выполнение сопоставления полей и вычисления для прочитанных данных, затем генерация `bucket-id` на основе информации о partition.
3. Генерация RollupTree на основе метаданных Rollup таблицы Selena.
4. Итерация по RollupTree и выполнение иерархических операций агрегации. Rollup следующей иерархии может быть вычислен из Rollup предыдущей иерархии.
5. Каждый раз после завершения вычисления агрегации данные разбиваются по bucket в соответствии с `bucket-id` и затем записываются в HDFS.
6. Последующий процесс Broker будет извлекать файлы из HDFS и импортировать их в узел BE Selena.

Базовые операции

Конфигурация cluster ETL

Apache Spark™ используется в качестве внешнего вычислительного ресурса в Selena для работы ETL. К Selena могут быть добавлены и другие внешние ресурсы, такие как Spark/GPU для запросов, HDFS/S3 для внешнего хранилища, MapReduce для ETL и т.д.

Поэтому мы вводим `Resource Management` для управления этими внешними ресурсами, используемыми Selena.

Перед отправкой задачи импорта Apache Spark™ настройте cluster Apache Spark™ для выполнения задач ETL. Синтаксис операции следующий:

```
-- создание ресурса Apache Spark™
CREATE EXTERNAL RESOURCE resource_name
PROPERTIES
(
 type = spark,
 spark_conf_key = spark_conf_value,
 working_dir = path,
 broker = broker_name,
 broker.property_key = property_value
);

-- удаление ресурса Apache Spark™
DROP RESOURCE resource_name;

-- просмотр ресурсов
SHOW RESOURCES
SHOW PROC "/resources";

-- привилегии
GRANT USAGE_PRIV ON RESOURCE resource_name TO user_identityGRANT USAGE_PRIV
ON RESOURCE resource_name TO ROLE role_name;
REVOKE USAGE_PRIV ON RESOURCE resource_name FROM user_identityREVOKE
USAGE_PRIV ON RESOURCE resource_name FROM ROLE role_name;
```

- Создание ресурса

Пример:

```
-- режим yarn cluster
CREATE EXTERNAL RESOURCE "spark0"
PROPERTIES
(
    "type" = "spark",
    "spark.master" = "yarn",
    "spark.submit.deployMode" = "cluster",
    "spark.jars" = "xxx.jar,yyy.jar",
    "spark.files" = "/tmp/aaa,/tmp/bbb",
    "spark.executor.memory" = "1g",
    "spark.yarn.queue" = "queue0",
    "spark.hadoop.yarn.resourcemanager.address" = "127.0.0.1:9999",
    "spark.hadoop.fs.defaultFS" = "hdfs://127.0.0.1:10000",
    "working_dir" = "hdfs://127.0.0.1:10000/tmp/selena",
    "broker" = "broker0",
    "broker.username" = "user0",
    "broker.password" = "password0"
);

-- режим yarn HA cluster
CREATE EXTERNAL RESOURCE "spark1"
PROPERTIES
(
    "type" = "spark",
    "spark.master" = "yarn",
    "spark.submit.deployMode" = "cluster",
    "spark.hadoop.yarn.resourcemanager.ha.enabled" = "true",
    "spark.hadoop.yarn.resourcemanager.ha.rm-ids" = "rm1,rm2",
    "spark.hadoop.yarn.resourcemanager.hostname.rm1" = "host1",
    "spark.hadoop.yarn.resourcemanager.hostname.rm2" = "host2",
    "spark.hadoop.fs.defaultFS" = "hdfs://127.0.0.1:10000",
    "working_dir" = "hdfs://127.0.0.1:10000/tmp/selena",
    "broker" = "broker1"
);
```

`resource-name` — это имя ресурса Apache Spark™, настроенного в Selena.

`PROPERTIES` включает параметры, относящиеся к ресурсу Apache Spark™, следующим образом:

> **Примечание**

> Для получения подробного описания PROPERTIES ресурса Apache Spark™ см. CREATE RESOURCE

- Параметры, связанные со Spark:
  - `type`: Тип ресурса, обязательный, в настоящее время поддерживается только `spark`.
  - `spark.master`: Обязательный, в настоящее время поддерживается только `yarn`.
    - `spark.submit.deployMode`: Режим развёртывания программы Apache Spark™, обязательный, в настоящее время поддерживаются как `cluster`, так и `client`.
    - `spark.hadoop.fs.defaultFS`: Обязательный, если master — yarn.
    - Параметры, связанные с yarn resource manager, обязательные.
      - один ResourceManager на одном узле `spark.hadoop.yarn.resourcemanager.address`: Адрес единственного resource manager.
      - ResourceManager HA

        > Вы можете выбрать указание hostname или адреса ResourceManager.

        - `spark.hadoop.yarn.resourcemanager.ha.enabled`: Включить HA resource manager, установите в `true`.
        - `spark.hadoop.yarn.resourcemanager.ha.rm-ids`: список логических идентификаторов resource manager.
        - `spark.hadoop.yarn.resourcemanager.hostname.rm-id`: Для каждого rm-id укажите hostname, соответствующий resource manager.
        - `spark.hadoop.yarn.resourcemanager.address.rm-id`: Для каждого rm-id укажите `host:port` для клиента для отправки задач.
- `*working_dir`: Каталог, используемый ETL. Обязательный, если Apache Spark™ используется в качестве ресурса ETL. Например: `hdfs://host:port/tmp/selena`.

- Параметры, связанные с Broker:
  - `broker`: Имя Broker. Обязательный, если Apache Spark™ используется в качестве ресурса ETL. Вам нужно заранее выполнить настройку с помощью команды `ALTER SYSTEM ADD BROKER`.
  - `broker.property_key`: Информация (например, информация для аутентификации), которая должна быть указана, когда процесс Broker читает

промежуточный файл, сгенерированный ETL.

Предостережение:

Выше приведено описание параметров для загрузки через процесс Broker. Если вы намерены загружать данные без процесса Broker, следует обратить внимание на следующее.

- Вам не нужно указывать `broker`.
- Если вам нужно настроить аутентификацию пользователя и HA для узлов NameNode, вам нужно настроить параметры в файле hdfs-site.xml в cluster HDFS, см. описания параметров в broker_properties. и вам нужно переместить файл **hdfs-site.xml** в **$FE_HOME/conf** для каждого FE и **$BE_HOME/conf** для каждого BE.

> Примечание
>
> Если файл HDFS доступен только определённому пользователю, вам всё ещё нужно указать имя пользователя HDFS в `broker.name` и пароль пользователя в `broker.password`.

- Просмотр ресурсов

Обычные учётные записи могут просматривать только ресурсы, к которым у них есть доступ `USAGE-PRIV`. Учётные записи root и admin могут просматривать все ресурсы.

- Разрешения ресурсов

Разрешения ресурсов управляются через `GRANT REVOKE`, который в настоящее время поддерживает только разрешения `USAGE-PRIV`. Вы можете предоставить разрешения `USAGE-PRIV` пользователю или роли.

```
-- Предоставить доступ к ресурсам spark0 пользователю user0
GRANT USAGE_PRIV ON RESOURCE "spark0" TO "user0"@"%";

-- Предоставить доступ к ресурсам spark0 роли role0
GRANT USAGE_PRIV ON RESOURCE "spark0" TO ROLE "role0";

-- Предоставить доступ ко всем ресурсам пользователю user0
GRANT USAGE_PRIV ON RESOURCE* TO "user0"@"%";

-- Предоставить доступ ко всем ресурсам роли role0
GRANT USAGE_PRIV ON RESOURCE* TO ROLE "role0";

-- Отозвать привилегии использования ресурсов spark0 у пользователя user0
REVOKE USAGE_PRIV ON RESOURCE "spark0" FROM "user0"@"%";
```

Конфигурация клиента Spark

Настройте клиент Spark для FE, чтобы последний мог отправлять задачи Spark, выполняя команду `spark-submit`. Рекомендуется использовать официальную версию Spark2 2.4.5 или выше (адрес загрузки spark). После загрузки используйте следующие шаги для завершения конфигурации.

- Настройка `SPARK-HOME`

Разместите клиент Spark в каталоге на той же машине, что и FE, и настройте `spark_home_default_dir` в файле конфигурации FE на этот каталог, который по умолчанию является путём `lib/spark2x` в корневом каталоге FE и не может быть пустым.

- **Настройка пакета зависимостей SPARK**

Для настройки пакета зависимостей заархивируйте все jar-файлы в папке jars под клиентом Spark и настройте элемент `spark_resource_path` в конфигурации FE на этот zip-файл. Если эта конфигурация пуста, FE попытается найти файл `lib/spark2x/jars/spark-2x.zip` в корневом каталоге FE. Если FE не найдёт его, будет выдана ошибка.

При отправке задачи Spark load заархивированные файлы зависимостей будут загружены в удалённый репозиторий. Путь репозитория по умолчанию находится в каталоге `working_dir/{cluster_id}` с именем `--spark-repository--{resource-name}`, что

означает, что ресурс в cluster соответствует удалённому репозиторию. Структура каталога приведена ниже:

```
---spark-repository--spark0/

   |---archive-1.0.0/

   |        |\---lib-990325d2c0d1d5e45bf675e54e44fb16-spark-dpp-1.0.0\-jar-with-dependencies.jar

   |        |\---lib-7670c29daf535efe3c9b923f778f61fc-spark-2x.zip

   |---archive-1.1.0/

   |        |\---lib-64d5696f99c379af2bee28c1c84271d5-spark-dpp-1.1.0\-jar-with-dependencies.jar

   |        |\---lib-1bbb74bb6b264a270bc7fca3e964160f-spark-2x.zip

   |---archive-1.2.0/

   |        |-...
```

Помимо зависимостей Spark (по умолчанию названных `spark-2x.zip`), FE также загружает зависимости DPP в удалённый репозиторий. Если все зависимости, отправленные Spark load, уже существуют в удалённом репозитории, то нет необходимости загружать зависимости снова, что экономит время на повторную загрузку большого количества файлов каждый раз.

Конфигурация клиента YARN

Настройте клиент yarn для FE, чтобы FE мог выполнять команды yarn для получения статуса запущенного приложения или его остановки. Рекомендуется использовать официальную версию Hadoop2 2.5.2 или выше (адрес загрузки hadoop). После загрузки используйте следующие шаги для завершения конфигурации:

- **Настройка пути к исполняемому файлу YARN**

Разместите загруженный клиент yarn в каталоге на той же машине, что и FE, и настройте элемент `yarn_client_path` в файле конфигурации FE на бинарный исполняемый файл

yarn, который по умолчанию является путём `lib/yarn-client/hadoop/bin/yarn` в корневом каталоге FE.

- **Настройка пути к файлу конфигурации, необходимому для генерации YARN (опционально)**

Когда FE через клиент yarn получает статус приложения или останавливает приложение, по умолчанию Selena генерирует файл конфигурации, необходимый для выполнения команды yarn, в пути `lib/yarn-config` корневого каталога FE. Этот путь можно изменить, настроив запись `yarn_config_dir` в файле конфигурации FE, которая в настоящее время включает `core-site.xml` и `yarn-site.xml`.

Создание задачи импорта

Синтаксис:

```
LOAD LABEL load_label
    (data_desc, ...)
WITH RESOURCE resource_name
[resource_properties]
[PROPERTIES (key1=value1, ... )]

* load_label:
    db_name.label_name

* data_desc:
    DATA INFILE ('file_path', ...)
    [NEGATIVE]
    INTO TABLE tbl_name
    [PARTITION (p1, p2)]
    [COLUMNS TERMINATED BY separator ]
    [(col1, ...)]
    [COLUMNS FROM PATH AS (col2, ...)]
    [SET (k1=f1(xx), k2=f2(xx))]
    [WHERE predicate]

    DATA FROM TABLE hive_external_tbl
    [NEGATIVE]
    INTO TABLE tbl_name
    [PARTITION (p1, p2)]
    [SET (k1=f1(xx), k2=f2(xx))]
    [WHERE predicate]

* resource_properties:
 (key2=value2, ...)
```

Пример 1: Случай, когда вышестоящий источник данных — HDFS

```
LOAD LABEL db1.label1
(
    DATA INFILE("hdfs://abc.com:8888/user/selena/test/ml/file1")
    INTO TABLE tbl1
    COLUMNS TERMINATED BY ","
    (tmp_c1,tmp_c2)
    SET
    (
        id=tmp_c2,
        name=tmp_c1
    ),
    DATA INFILE("hdfs://abc.com:8888/user/selena/test/ml/file2")
    INTO TABLE tbl2
    COLUMNS TERMINATED BY ","
    (col1, col2)
    where col1 > 1
)
WITH RESOURCE 'spark0'
(
    "spark.executor.memory" = "2g",
    "spark.shuffle.compress" = "true"
)
PROPERTIES
(
    "timeout" = "3600"
);
```

Пример 2: Случай, когда вышестоящий источник данных — Hive.

- Шаг 1: Создайте новый ресурс hive

```
CREATE EXTERNAL RESOURCE hive0
PROPERTIES
(
    "type" = "hive",
    "hive.metastore.uris" = "thrift://xx.xx.xx.xx:8080"
);
```

- Шаг 2: Создайте новую внешнюю таблицу hive

```
CREATE EXTERNAL TABLE hive_t1
(
    k1 INT,
    K2 SMALLINT,
    k3 varchar(50),
    uuid varchar(100)
)
ENGINE=hive
PROPERTIES
(
    "resource" = "hive0",
    "database" = "tmp",
    "table" = "t1"
);
```

- Шаг 3: Отправьте команду загрузки, требуя, чтобы столбцы в импортируемой таблице Selena существовали во внешней таблице hive.

```
LOAD LABEL db1.label1
(
    DATA FROM TABLE hive_t1
    INTO TABLE tbl1
    SET
    (
        uuid=bitmap_dict(uuid)
    )
)
WITH RESOURCE 'spark0'
(
    "spark.executor.memory" = "2g",
    "spark.shuffle.compress" = "true"
)
PROPERTIES
(
    "timeout" = "3600"
);
```

Введение в параметры в Spark load:

- **Label**

Label задачи импорта. Каждая задача импорта имеет Label, который уникален в пределах базы данных, следуя тем же правилам, что и broker load.

- **Параметры класса описания данных**

В настоящее время поддерживаемые источники данных — CSV и таблица Hive. Остальные правила такие же, как и для broker load.

- **Параметры задачи импорта**

Параметры задачи импорта относятся к параметрам, принадлежащим разделу `opt_properties` оператора импорта. Эти параметры применяются ко всей задаче импорта. Правила такие же, как и для broker load.

- **Параметры ресурса Spark**

Ресурсы Spark должны быть заранее настроены в Selena, и пользователям необходимо предоставить разрешения USAGE-PRIV, прежде чем они смогут применять ресурсы к Spark load. Параметры ресурса Spark можно установить, когда у пользователя есть временная потребность, например, добавление ресурсов для задачи и изменение конфигураций Spark. Настройка действует только для этой задачи и не влияет на существующие конфигурации в cluster Selena.

```
WITH RESOURCE 'spark0'
(
    "spark.driver.memory" = "1g",
    "spark.executor.memory" = "3g"
)
```

- **Импорт, когда источник данных — Hive**

В настоящее время для использования таблицы Hive в процессе импорта вам нужно создать внешнюю таблицу типа `Hive`, а затем указать её имя при отправке команды импорта.

- **Процесс импорта для построения глобального словаря**

В команде загрузки вы можете указать необходимые поля для построения глобального словаря в следующем формате: `имя поля Selena=bitmap_dict(имя поля таблицы`

`hive)`. Обратите внимание, что в настоящее время **глобальный словарь поддерживается только когда вышестоящий источник данных — таблица Hive**.

- **Загрузка данных бинарного типа**

Начиная с версии v1.5.2, Spark Load поддерживает функцию bitmap_from_binary, которая может конвертировать бинарные данные в данные bitmap. Если тип столбца таблицы Hive или файла HDFS — binary, а соответствующий столбец в таблице Selena — агрегатный столбец типа bitmap, вы можете указать поля в команде загрузки в следующем формате: `имя поля Selena=bitmap_from_binary(имя поля таблицы Hive)`. Это устраняет необходимость построения глобального словаря.

Просмотр задач импорта

Импорт Spark load является асинхронным, как и broker load. Пользователь должен записать label задачи импорта и использовать его в команде `SHOW LOAD` для просмотра результатов импорта. Команда просмотра импорта общая для всех методов импорта. Пример следующий.

Обратитесь к Broker Load для получения подробного объяснения возвращаемых параметров. Различия следующие.

```
mysql> show load order by createtime desc limit 1\G
*************************** 1. row ***************************
  JobId: 76391
  Label: label1
  State: FINISHED
 Progress: ETL:100%; LOAD:100%
  Type: SPARK
 EtlInfo: unselected.rows=4; dpp.abnorm.ALL=15; dpp.norm.ALL=28133376
 TaskInfo: cluster:cluster0; timeout(s):10800; max_filter_ratio:5.0E-5
 ErrorMsg: N/A
 CreateTime: 2019-07-27 11:46:42
 EtlStartTime: 2019-07-27 11:46:44
 EtlFinishTime: 2019-07-27 11:49:44
 LoadStartTime: 2019-07-27 11:49:44
LoadFinishTime: 2019-07-27 11:50:16
  URL: http://1.1.1.1:8089/proxy/application_1586619723848_0035/
 JobDetails:
{"ScannedRows":28133395,"TaskNumber":1,"FileNumber":1,"FileSize":200000}
```

- **State**

Текущая стадия импортируемой задачи. PENDING: Задача зафиксирована. ETL: Spark ETL зафиксирован. LOADING: FE планирует BE для выполнения операции push. FINISHED: Push завершён, и версия эффективна.

Есть две финальные стадии задачи импорта — `CANCELLED` и `FINISHED`, обе указывают на завершение задачи загрузки. `CANCELLED` указывает на неудачу импорта, а `FINISHED` указывает на успех импорта.

- **Progress**

Описание прогресса задачи импорта. Есть два типа прогресса — ETL и LOAD, которые соответствуют двум фазам процесса импорта, ETL и LOADING.

- Диапазон прогресса для LOAD — 0~100%.

`LOAD прогресс = количество текущих завершённых tablet всех replica импортов / общее количество tablet этой задачи импорта * 100%`.

- Если все таблицы были импортированы, прогресс LOAD составляет 99% и меняется на 100%, когда импорт входит в финальную фазу проверки.

- Прогресс импорта не линейный. Если прогресс не меняется в течение некоторого времени, это не означает, что импорт не выполняется.

- **Type**

Тип задачи импорта. SPARK для Spark load.

- **CreateTime/EtlStartTime/EtlFinishTime/LoadStartTime/LoadFinishTime**

Эти значения представляют время создания импорта, время начала фазы ETL, время завершения фазы ETL, время начала фазы LOADING и время завершения всей задачи импорта.

- **JobDetails**

Отображает подробный статус выполнения задачи, включая количество импортированных файлов, общий размер (в байтах), количество подзадач, количество обработанных

исходных строк и т.д. Например:

```
{"ScannedRows":139264,"TaskNumber":1,"FileNumber":1,"FileSize":940754064}
```

- **URL**

Вы можете скопировать ввод в ваш браузер для доступа к веб-интерфейсу соответствующего приложения.

Просмотр логов отправки Apache Spark™ Launcher

Иногда пользователям нужно просмотреть подробные логи, сгенерированные во время отправки задачи Apache Spark™. По умолчанию логи сохраняются в пути `log/spark_launcher_log` в корневом каталоге FE с именем `spark-launcher-{load-job-id}-{label}.log`. Логи сохраняются в этом каталоге в течение определённого времени и будут стёрты при очистке информации об импорте в метаданных FE. Время хранения по умолчанию — 3 дня.

Отмена импорта

Когда статус задачи Spark load не `CANCELLED` или `FINISHED`, она может быть отменена вручную пользователем путём указания Label задачи импорта.

Связанные системные конфигурации

Конфигурация FE: Следующая конфигурация является конфигурацией системного уровня для Spark load, которая применяется ко всем задачам импорта Spark load. Значения конфигурации можно настроить главным образом путём изменения `fe.conf`.

- enable-spark-load: Включить Spark load и создание ресурсов со значением по умолчанию false.
- spark-load-default-timeout-second: Таймаут по умолчанию для задачи — 259200 секунд (3 дня).
- spark-home-default-dir: Путь к клиенту Spark (`fe/lib/spark2x`).
- spark-resource-path: Путь к упакованному файлу зависимостей Spark (по умолчанию пустой).
- spark-launcher-log-dir: Каталог, где хранится лог отправки клиента Spark (`fe/log/spark-launcher-log`).

- yarn-client-path: Путь к бинарному исполняемому файлу yarn (`fe/lib/yarn-client/hadoop/bin/yarn`).
- yarn-config-dir: Путь к файлу конфигурации Yarn (`fe/lib/yarn-config`).

Лучшие практики

Наиболее подходящий сценарий для использования Spark load — когда исходные данные находятся в файловой системе (HDFS), а объём данных составляет от десятков ГБ до уровня ТБ. Используйте Stream Load или Broker Load для меньших объёмов данных.

Для полного примера импорта Spark load обратитесь к демо на github: https://github.com/Selena/demo/blob/master/docs/03_sparkLoad2Selena.md

FAQ

- `Error: When running with master 'yarn' either HADOOP-CONF-DIR or YARN-CONF-DIR must be set in the environment.`

Использование Spark Load без настройки переменной окружения `HADOOP-CONF-DIR` в `spark-env.sh` клиента Spark.

- `Error: Cannot run program "xxx/bin/spark-submit": error=2, No such file or directory`

Элемент конфигурации `spark_home_default_dir` не указывает корневой каталог клиента Spark при использовании Spark Load.

- `Error: File xxx/jars/spark-2x.zip does not exist.`

Элемент конфигурации `spark-resource-path` не указывает на упакованный zip-файл при использовании Spark load.

- `Error: yarn client does not exist in path: xxx/yarn-client/hadoop/bin/yarn`

Элемент конфигурации yarn-client-path не указывает исполняемый файл yarn при использовании Spark load.

- `ERROR: Cannot execute hadoop-yarn/bin/... /libexec/yarn-config.sh`

При использовании Hadoop с CDH вам нужно настроить переменную окружения `HADOOP_LIBEXEC_DIR`. Поскольку `hadoop-yarn` и каталоги hadoop разные, каталог

`libexec` по умолчанию будет искать `hadoop-yarn/bin/... /libexec`, в то время как `libexec` находится в каталоге hadoop. Команда ```yarn application status`` для получения статуса задачи Spark выдала ошибку, что привело к сбою задачи импорта.

1.5 Loading_tools_integration

1.6 Загрузка данных с помощью инструментов

Selena и её партнёры по экосистеме предлагают следующие инструменты, которые помогут вам беспрепятственно интегрировать Selena с внешними базами данных.

1.6.1 SMT

SMT (Selena Migration Tool) — это инструмент миграции данных, предоставляемый Selena, разработанный для оптимизации сложных конвейеров загрузки данных: исходные базы данных (такие как MySQL, Oracle, PostgreSQL) ---> Flink ---> целевые cluster Selena. Его основные функции следующие:

- Упрощение создания таблиц в Selena: генерирует операторы для создания таблиц в Selena на основе информации из внешних баз данных и целевого cluster Selena.
- Упрощение процесса полной или инкрементальной синхронизации данных в конвейере данных: генерирует SQL-операторы, которые можно выполнить в SQL-клиенте Flink для отправки заданий Flink для синхронизации данных.

Следующая блок-схема иллюстрирует процесс загрузки данных из исходной базы данных MySQL через Flink в Selena.

1.6.2 DataX

DataX — это инструмент для офлайн-синхронизации данных, открытый Alibaba. DataX может синхронизировать данные между различными гетерогенными источниками данных, включая реляционные базы данных (MySQL, Oracle и т.д.), HDFS и Hive. DataX

предоставляет плагин Selena Writer для синхронизации данных из источников данных, поддерживаемых DataX, в Selena.

1.6.3 CloudCanal

CloudCanal Community Edition — это бесплатная платформа для миграции и синхронизации данных, опубликованная ClouGence Co., Ltd, которая интегрирует миграцию схемы, полную миграцию данных, проверку, исправление и инкрементальную синхронизацию в реальном времени. Вы можете напрямую добавить Selena в качестве источника данных в визуальном интерфейсе CloudCanal и создать задачи для автоматической миграции или синхронизации данных из исходных баз данных (например, MySQL, Oracle, PostgreSQL) в Selena.

1.6.4 Kettle connector

Kettle — это инструмент ETL (Extract, Transform, Load) с визуальным графическим интерфейсом, который позволяет пользователям создавать рабочие процессы обработки данных путём перетаскивания компонентов и настройки параметров. Этот интуитивный метод значительно упрощает процесс обработки и загрузки данных, позволяя пользователям более удобно обрабатывать данные. Кроме того, Kettle предоставляет богатую библиотеку компонентов, позволяя пользователям выбирать подходящие компоненты в соответствии со своими потребностями и выполнять различные сложные задачи обработки данных.

Selena предлагает Kettle Connector для интеграции с Kettle. Объединяя мощные возможности обработки и трансформации данных Kettle с высокопроизводительными возможностями хранения и аналитики данных Selena, можно достичь более гибких и эффективных рабочих процессов обработки данных.

1.6.5 Selena Migration Tool (SMT)

Selena Migration Tool (SMT) — это инструмент миграции данных, предоставляемый Selena для загрузки данных из исходных баз данных через Flink в Selena. Основные возможности SMT:

- Генерация операторов создания таблиц в Selena на основе информации об исходной базе данных и целевом cluster Selena.
- Генерация SQL-операторов, которые могут быть выполнены в SQL-клиенте Flink для отправки заданий Flink по синхронизации данных, что упрощает полную или

инкрементную синхронизацию данных в pipeline. В настоящее время SMT поддерживает следующие исходные базы данных:

Исходная база данных	Генерация оператора создания таблицы в Selena	Полная синхронизация данных	Инкрементная синхронизация данных
MySQL	Поддерживается	Поддерживается	Поддерживается
PostgreSQL	Поддерживается	Поддерживается	Поддерживается
Oracle	Поддерживается	Поддерживается	Поддерживается
Hive	Поддерживается	Поддерживается	Не поддерживается
ClickHouse	Поддерживается	Поддерживается	Не поддерживается
SQL Server	Поддерживается	Поддерживается	Поддерживается
TiDB	Поддерживается	Поддерживается	Поддерживается

Ссылка для скачивания: https://cdn-thirdparty.selena.com/smt.tar.gz?r=2

Шаги для использования SMT

Общие шаги следующие:

1. Настройте файл **conf/config_prod.conf**.
2. Выполните **selena-migration-tool**.
3. После выполнения SQL-скрипты генерируются в директории **result** по умолчанию.

   Затем вы можете использовать SQL-скрипты из директории **result** для синхронизации метаданных или данных.

Конфигурация SMT

- `[db]`: информация для подключения к источнику данных. Настройте информацию для подключения к источнику данных, соответствующему типу базы данных, указанному в параметре `type`.
- `[other]`: дополнительные настройки. Рекомендуется указать фактическое количество BE-узлов в параметре `be_num`.
- `flink.selena.sink.*`: конфигурации flink-connector-selena. Подробное описание конфигураций см. в описании конфигурации.

- `[table-rule.1]`: правило для сопоставления таблиц в источнике данных. Оператор CREATE TABLE генерируется на основе регулярных выражений, настроенных в правиле для сопоставления имен баз данных и таблиц в источнике данных. Можно настроить несколько правил, и каждое правило генерирует соответствующий файл результата, например:
  - `[table-rule.1]` -> `result/selena-create.1.sql`
  - `[table-rule.2]` -> `result/selena-create.2.sql`

  Каждое правило должно содержать конфигурации базы данных, таблицы и flink-connector-selena.

```
[table-rule.1]
# pattern to match databases for setting properties
database = ^ database1.*$
# pattern to match tables for setting properties
table = ^.*$
schema = ^.*$
##############################################
### flink sink configurations
### DO NOT set `connector`, `table-name`, `database-name`, they are
auto-generated
##############################################
flink.selena.jdbc-url=jdbc:mysql://192.168.1.1:9030
flink.selena.load-url= 192.168.1.1:8030
flink.selena.username=root
flink.selena.password=
flink.selena.sink.max-retries=10
flink.selena.sink.buffer-flush.interval-ms=15000
flink.selena.sink.properties.format=json
flink.selena.sink.properties.strip_outer_array=true

[table-rule.2]
# pattern to match databases for setting properties
database = ^database2.*$
# pattern to match tables for setting properties
table = ^.*$
schema = ^.*$
##############################################
### flink sink configurations
### DO NOT set `connector`, `table-name`, `database-name`, they are
auto-generated
##############################################
flink.selena.jdbc-url=jdbc:mysql://192.168.1.1:9030
flink.selena.load-url= 192.168.1.1:8030
flink.selena.username=root
flink.selena.password=
flink.selena.sink.max-retries=10
flink.selena.sink.buffer-flush.interval-ms=15000
flink.selena.sink.properties.format=json
flink.selena.sink.properties.strip_outer_array=true
```

- Для большой таблицы, разделенной на шарды в базах данных, можно настроить отдельное правило. Например, предположим, что две базы данных `edu_db_1` и `edu_db_2` содержат таблицы `course_1` и `course_2` соответственно, и эти две

таблицы имеют одинаковую структуру. Вы можете использовать следующее правило для загрузки данных из этих двух таблиц в одну таблицу Selena для анализа.

```
[table-rule.3]
# pattern to match databases for setting properties
database = ^edu_db_[0-9]*$
# pattern to match tables for setting properties
table = ^course_[0-9]*$
schema = ^.*$
```

Это правило автоматически сформирует отношение загрузки "многие к одному". Имя таблицы по умолчанию, которая будет создана в Selena, — `course__auto_shard`, и вы также можете изменить имя таблицы в соответствующем SQL-скрипте, таком как `result/selena-create.3.sql`.

Синхронизация MySQL с Selena

Введение

Flink CDC connector и SMT могут синхронизировать данные из MySQL в течение доли секунды.

Как показано на изображении, SMT может автоматически генерировать операторы CREATE TABLE для исходных и целевых таблиц Flink на основе информации о cluster и структуре таблиц MySQL и Selena. Flink CDC connector читает MySQL Binlog, а Flink-connector-selena записывает данные в Selena.

Шаги

Зависимость	Имя пакета	Ссылка для скачивания
Flink	flink-x.x.x-bin-scala_2.12.tgz	Нажмите здесь
Flink CDC connector	flink-sql-connector-mysql-cdc-x.x.x.jar	Нажмите здесь

Зависимость	Имя пакета	Ссылка для скачивания
Flink-connector-selena	flink-connector-selena-x.x.x_flink-x.x.jar	Нажмите здесь
SMT	smt.tar.gz	Нажмите здесь

1. Скачайте Flink. Поддерживается Flink 1.11 или более поздней версии.

2. Скачайте Flink CDC connector. Убедитесь, что вы скачали `flink-sql-connector-mysql-cdc-xxx.jar`, соответствующий версии Flink.

3. Скачайте Flink-connector-selena.

4. Скопируйте **flink-sql-connector-mysql-cdc-xxx.jar** и **flink-connector-selena-xxx.jar** в **flink-xxx/lib/**.

5. Скачайте smt.tar.gz.

6. Распакуйте и измените файл конфигурации SMT.

```
[db]
host = 192.168.1.1
port = 3306
user = root
password =
type = mysql

[other]
# number of backends in Selena
be_num = 3
# `decimal_v3` is supported since Selena-1.18.1
use_decimal_v3 = false
# directory to save the converted DDL SQL
output_dir = ./result

[table-rule.1]
# pattern to match databases for setting properties
database = ^db$
# pattern to match tables for setting properties
table = ^table$
schema = ^.*$

##############################################
### flink sink configurations
### DO NOT set `connector`, `table-name`, `database-name`, they are
auto-generated
##############################################
flink.selena.jdbc-url=jdbc:mysql://192.168.1.1:9030
flink.selena.load-url= 192.168.1.1:8030
flink.selena.username=root
flink.selena.password=
flink.selena.sink.max-retries=10
flink.selena.sink.buffer-flush.interval-ms=15000
flink.selena.sink.properties.format=json
flink.selena.sink.properties.strip_outer_array=true
```

7. Выполните **selena-migrate-tool**. Все SQL-скрипты генерируются в директории result.

```
$./selena-migrate-tool
$ls result
flink-create.1.sql    smt.tar.gz              selena-create.all.sql
flink-create.all.sql  selena-create.1.sql  selena-external-
create.all.sql
```

8. Используйте SQL-скрипт с префиксом **selena-create** для создания таблицы в Selena.

```
mysql -hxx.xx.xx.x -P9030 -uroot -p < selena-create.all.sql
```

9. Используйте SQL-скрипт с префиксом **flink-create** для создания исходных и целевых таблиц Flink и запуска задания Flink для синхронизации данных.

```
bin/sql-client.sh embedded < flink-create.all.sql
```

После успешного выполнения указанной выше команды задание Flink по синхронизации данных продолжает работать.

10. Наблюдайте за статусом задания Flink.

```
bin/flink list
```

Если при выполнении задания возникает ошибка, вы можете просмотреть подробную информацию об ошибке в логах Flink. Кроме того, вы можете изменить конфигурации Flink в файле **conf/flink-conf.yaml**, такие как память и slot.

Примечания

- Как включить MySQL binlog?

  i. Измените /etc/my.cnf:

```
# Enable binlog
log-bin=/var/lib/mysql/mysql-bin

#log_bin=ON
## Base name of binlog files
#log_bin_basename=/var/lib/mysql/mysql-bin
## Index file for binlog files, managing all binlog files
#log_bin_index=/var/lib/mysql/mysql-bin.index
# Configure server id
server-id=1
binlog_format = row
```

ii. Перезапустите mysqld. Вы можете проверить, включен ли MySQL binlog, выполнив `SHOW VARIABLES LIKE 'log_bin';`.

Синхронизация PostgreSQL с Selena

Введение

Flink CDC connector и SMT могут синхронизировать данные из PostgreSQL в течение доли секунды.

SMT может автоматически генерировать операторы CREATE TABLE для исходных и целевых таблиц Flink на основе информации о cluster и структуре таблиц PostgreSQL и Selena.

Flink CDC connector читает WAL PostgreSQL, а Flink-connector-selena записывает данные в Selena.

Шаги

1. Скачайте Flink. Поддерживается версия Flink 1.11 или более поздняя.

2. Скачайте Flink CDC connector. Убедитесь, что вы скачали flink-sql-connector-postgres-cdc-xxx.jar, соответствующий версии Flink.

3. Скачайте Flink Selena connector.

4. Скопируйте **flink-sql-connector-postgres-cdc-xxx.jar** и **flink-connector-selena-xxx.jar** в **flink-xxx/lib/**.

5. Скачайте smt.tar.gz.

6. Распакуйте и измените файл конфигурации SMT.

```
[db]
host = 192.168.1.1
port = 5432
user = xxx
password = xxx
type = pgsql

[other]
# number of backends in Selena
be_num = 3
# `decimal_v3` is supported since Selena-1.18.1
use_decimal_v3 = false
# directory to save the converted DDL SQL
output_dir = ./result

[table-rule.1]
# pattern to match databases for setting properties
database = ^db$
# pattern to match tables for setting properties
table = ^table$
# pattern to match schemas for setting properties
schema = ^.*$

##############################################
### flink sink configurations
### DO NOT set `connector`, `table-name`, `database-name`, they are
auto-generated
##############################################
flink.selena.jdbc-url=jdbc:mysql://192.168.1.1:9030
flink.selena.load-url= 192.168.1.1:8030
flink.selena.username=root
flink.selena.password=
flink.selena.sink.max-retries=10
flink.selena.sink.buffer-flush.interval-ms=15000
flink.selena.sink.properties.format=json
flink.selena.sink.properties.strip_outer_array=true
```

7. Выполните **selena-migrate-tool**. Все SQL-скрипты генерируются в директории result.

```
$./selena-migrate-tool
$ls result
flink-create.1.sql    smt.tar.gz              selena-create.all.sql
flink-create.all.sql  selena-create.1.sql
```

8. Используйте SQL-скрипт с префиксом **selena-create** для создания таблицы в Selena.

```
mysql -hxx.xx.xx.x -P9030 -uroot -p < selena-create.all.sql
```

9. Используйте SQL-скрипт с префиксом **flink-create** для создания исходных и целевых таблиц Flink и запуска задания Flink для синхронизации данных.

```
bin/sql-client.sh embedded < flink-create.all.sql
```

После успешного выполнения указанной выше команды задание Flink по синхронизации данных продолжает работать.

10. Наблюдайте за статусом задания Flink.

```
bin/flink list
```

Если при выполнении задания возникает ошибка, вы можете просмотреть подробную информацию об ошибке в логах Flink. Кроме того, вы можете изменить конфигурации Flink в файле **conf/flink-conf.yaml**, такие как память и slot.

Примечания

- Для PostgreSQL `v9.*` требуется специальная конфигурация flink-cdc, как показано ниже (Рекомендуется использовать PostgreSQL `v10.*` или более поздней версии. В противном случае вам необходимо установить плагины декодирования WAL):

```
##############################################
##############################################
### flink-cdc plugin configuration for `postgresql`
##############################################
### for `9.*` decoderbufs, wal2json, wal2json_rds, wal2json_streaming,
wal2json_rds_streaming
### refer to https://ververica.github.io/flink-cdc-
connectors/master/content/connectors/postgres-cdc.html
### and https://debezium.io/documentation/reference/postgres-
plugins.html
### flink.cdc.decoding.plugin.name = decoderbufs
```

- Как включить PostgreSQL WAL?

```
# Open connection permissions
echo "host all all 0.0.0.0/32 trust" >> pg_hba.conf
echo "host replication all 0.0.0.0/32 trust" >> pg_hba.conf
# Enable wal logical replication
echo "wal_level = logical" >> postgresql.conf
echo "max_wal_senders = 2" >> postgresql.conf
echo "max_replication_slots = 8" >> postgresql.conf
```

Укажите replica identity FULL для таблиц, которые необходимо синхронизировать.

```
ALTER TABLE schema_name.table_name REPLICA IDENTITY FULL
```

После внесения этих изменений перезапустите PostgreSQL.

Синхронизация Oracle с Selena

Введение

Flink CDC connector и SMT могут синхронизировать данные из Oracle в течение доли секунды.

SMT может автоматически генерировать операторы CREATE TABLE для исходных и целевых таблиц Flink на основе информации о cluster и структуре таблиц Oracle и Selena.

Flink CDC connector читает logminer Oracle, а Flink-connector-selena записывает данные в Selena.

Шаги

1. Скачайте Flink. Поддерживается версия Flink 1.11 или более поздняя.

2. Скачайте Flink CDC connector. Убедитесь, что вы скачали flink-sql-connector-oracle-cdc-xxx.jar, соответствующий версии Flink.

3. Скачайте Flink Selena connector.

4. Скопируйте `flink-sql-connector-oracle-cdc-xxx.jar` и `flink-connector-selena-xxx.jar` в `flink-xxx/lib/`.

5. Скачайте smt.tar.gz.

6. Распакуйте и измените файл конфигурации SMT.

```
[db]
host = 192.168.1.1
port = 1521
user = xxx
password = xxx
type = oracle

[other]
# number of backends in Selena
be_num = 3
# `decimal_v3` is supported since Selena-1.18.1
use_decimal_v3 = false
# directory to save the converted DDL SQL
output_dir = ./result

[table-rule.1]
# pattern to match databases for setting properties
database = ^db$
# pattern to match tables for setting properties
table = ^table$
# pattern to match schemas for setting properties
schema = ^.*$

##############################################
### flink sink configurations
### DO NOT set `connector`, `table-name`, `database-name`, they are
auto-generated
##############################################
flink.selena.jdbc-url=jdbc:mysql://192.168.1.1:9030
flink.selena.load-url= 192.168.1.1:8030
flink.selena.username=root
flink.selena.password=
flink.selena.sink.max-retries=10
flink.selena.sink.buffer-flush.interval-ms=15000
flink.selena.sink.properties.format=json
flink.selena.sink.properties.strip_outer_array=true
```

7. Выполните **selena-migrate-tool**. Все SQL-скрипты генерируются в директории result.

```
$./selena-migrate-tool
$ls result
flink-create.1.sql    smt.tar.gz              selena-create.all.sql
flink-create.all.sql  selena-create.1.sql
```

8. Используйте SQL-скрипт с префиксом selena-create для создания таблицы в Selena.

```
mysql -hxx.xx.xx.x -P9030 -uroot -p < selena-create.all.sql
```

9. Используйте SQL-скрипт с префиксом flink-create для создания исходных и целевых таблиц Flink и запуска задания Flink для синхронизации данных.

```
bin/sql-client.sh embedded < flink-create.all.sql
```

После успешного выполнения указанной выше команды задание Flink по синхронизации данных продолжает работать.

10. Наблюдайте за статусом задания Flink.

```
bin/flink list
```

Если при выполнении задания возникает ошибка, вы можете просмотреть подробную информацию об ошибке в логах Flink. Кроме того, вы можете изменить конфигурации Flink в файле **conf/flink-conf.yaml**, такие как память и slot.

Примечания

- Синхронизация Oracle с использованием logminer:

```
# Enable logging
alter system set db_recovery_file_dest = '/home/oracle/data'
scope=spfile;
alter system set db_recovery_file_dest_size = 10G;
shutdown immediate;
startup mount;
alter database archivelog;
alter database open;

ALTER TABLE schema_name.table_name ADD SUPPLEMENTAL LOG DATA (ALL)
COLUMNS;
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;

# Authorize user creation and grant permissions
GRANT CREATE SESSION TO flinkuser;
GRANT SET CONTAINER TO flinkuser;
GRANT SELECT ON V_$DATABASE TO flinkuser;
GRANT FLASHBACK ANY TABLE TO flinkuser;
GRANT SELECT ANY TABLE TO flinkuser;
GRANT SELECT_CATALOG_ROLE TO flinkuser;
GRANT EXECUTE_CATALOG_ROLE TO flinkuser;
GRANT SELECT ANY TRANSACTION TO flinkuser;
GRANT LOGMINING TO flinkuser;
GRANT CREATE TABLE TO flinkuser;
GRANT LOCK ANY TABLE TO flinkuser;
GRANT ALTER ANY TABLE TO flinkuser;
GRANT CREATE SEQUENCE TO flinkuser;
GRANT EXECUTE ON DBMS_LOGMNR TO flinkuser;
GRANT EXECUTE ON DBMS_LOGMNR_D TO flinkuser;
GRANT SELECT ON V_$LOG TO flinkuser;
GRANT SELECT ON V_$LOG_HISTORY TO flinkuser;
GRANT SELECT ON V_$LOGMNR_LOGS TO flinkuser;
GRANT SELECT ON V_$LOGMNR_CONTENTS TO flinkuser;
GRANT SELECT ON V_$LOGMNR_PARAMETERS TO flinkuser;
GRANT SELECT ON V_$LOGFILE TO flinkuser;
GRANT SELECT ON V_$ARCHIVED_LOG TO flinkuser;
GRANT SELECT ON V_$ARCHIVE_DEST_STATUS TO flinkuser;
```

- Конфигурация базы данных в [table-rule.1] не поддерживает регулярные выражения, поэтому необходимо указывать полные имена баз данных.
- Поскольку Oracle12c поддерживает режим CDB, SMT внутренне автоматически определяет, включен ли CDB, и соответствующим образом изменяет конфигурацию

flink-cdc. Однако пользователям необходимо обратить внимание на то, нужно ли добавлять префикс c## к конфигурации `[db].user`, чтобы избежать проблем с недостаточными разрешениями.

Синхронизация Hive с Selena

Введение

Это руководство объясняет, как использовать SMT для синхронизации данных Hive с Selena. Во время синхронизации в Selena создается Duplicate-таблица, и задание Flink продолжает работать для синхронизации данных.

Шаги

Подготовка

```
[db]
# hiveserver2 service ip
host = 127.0.0.1
# hiveserver2 service port
port = 10000
user = hive/emr-header-1.cluster-49148
password =
type = hive
# only takes effect with `type = hive`.
# Available values: kerberos, none, nosasl, kerberos_http, none_http, zk,
ldap
authentication = kerberos
```

Поддерживаемые методы аутентификации следующие:

- nosasl, zk: не нужно указывать `user` и `password`.
- none, none_http, ldap: указывайте `user` и `password`.
- kerberos, kerberos_http: выполните следующие шаги:
  - Выполните `kadmin.local` на cluster Hive и проверьте `list_principals`, чтобы найти соответствующее имя principal. Например, когда имя principal — `hive/emr-header-1.cluster-49148@EMR.49148.COM`, пользователь должен быть установлен как `hive/emr-header-1.cluster-49148`, а пароль остается пустым.
  - Выполните `kinit -kt /path/to/keytab principal` на машине, где выполняется SMT, и выполните `klist`, чтобы проверить, сгенерирован ли правильный токен.

Синхронизация данных

1. Выполните **selena-migrate-tool**.

2. Используйте SQL-скрипт с префиксом **selena-create** для создания таблицы в Selena.

```
mysql -hxx.xx.xx.x -P9030 -uroot -p < selena-create.all.sql
```

3. В **flink/conf/** создайте и отредактируйте файл **sql-client-defaults.yaml**:

```
execution:
planner: blink
type: batch
current-catalog: hive-selena
catalogs:
- name: hive-selena
type: hive
hive-conf-dir: /path/to/apache-hive-xxxx-bin/conf
```

4. Скачайте пакет зависимостей (flink-sql-connector-hive-xxxx) со страницы Hive соответствующей версии Flink и поместите его в директорию `flink/lib`.

5. Запустите cluster Flink и выполните `flink/bin/sql-client.sh embedded < result/flink-create.all.sql` для запуска синхронизации данных.

Синхронизация SQL Server с Selena

Введение

Flink CDC connector и SMT могут синхронизировать данные из SQL Server в течение доли секунды.

SMT может автоматически генерировать операторы CREATE TABLE для исходных и целевых таблиц Flink на основе информации о cluster и структуре таблиц SQL Server и Selena.

Flink CDC connector захватывает и записывает изменения на уровне строк, происходящие на сервере базы данных SQL Server. Принцип заключается в использовании функции CDC, предоставляемой самим SQL Server. Возможность CDC самого SQL Server может архивировать указанные изменения в базе данных в указанные таблицы изменений. SQL

Server CDC connector сначала считывает исторические данные из таблицы с помощью JDBC, а затем извлекает инкрементные изменения из таблиц изменений, тем самым достигая полной инкрементной синхронизации. Затем Flink-connector-selena записывает данные в Selena.

Шаги

1. Скачайте Flink. Поддерживается версия Flink 1.11 или более поздняя.

2. Скачайте Flink CDC connector. Убедитесь, что вы скачали **flink-sql-connector-sqlserver-cdc-xxx.jar**, соответствующий версии Flink.

3. Скачайте Flink Selena connector.

4. Скопируйте **flink-sql-connector-sqlserver-cdc-xxx.jar**, **flink-connector-selena-xxx.jar** в **flink-xxx/lib/**.

5. Скачайте smt.tar.gz.

6. Распакуйте и измените файл конфигурации SMT.

```
[db]
host = 127.0.0.1
port = 1433
user = xxx
password = xxx

# currently available types: `mysql`, `pgsql`, `oracle`, `hive`,
`clickhouse`
type = sqlserver

[other]
# number of backends in Selena
be_num = 3
# `decimal_v3` is supported since Selena-1.18.1
use_decimal_v3 = false
# directory to save the converted DDL SQL
output_dir = ./result

[table-rule.1]
# pattern to match databases for setting properties
database = ^db$
# pattern to match tables for setting properties
table = ^table$
schema = ^.*$

##############################################
### flink sink configurations
### DO NOT set `connector`, `table-name`, `database-name`, they are
auto-generated
##############################################
flink.selena.jdbc-url=jdbc:mysql://192.168.1.1:9030
flink.selena.load-url= 192.168.1.1:8030
flink.selena.username=root
flink.selena.password=
flink.selena.sink.max-retries=10
flink.selena.sink.buffer-flush.interval-ms=15000
flink.selena.sink.properties.format=json
flink.selena.sink.properties.strip_outer_array=true
```

7. Выполните **selena-migrate-tool**. Все SQL-скрипты генерируются в директории `result`.

```Bash $./selena-migrate-tool $ls result flink-create.1.sql smt.tar.gz selena-create.all.sql flink-create.all.sql selena-create.1.sql selena-external-create.all.sql
```

```
8. Используйте SQL-скрипт с префиксом `selena-create` для создания таблицы
в Selena.

```Bash
mysql -hxx.xx.xx.x -P9030 -uroot -p < selena-create.all.sql
```
```

9. Используйте SQL-скрипт с префиксом `flink-create` для создания исходных и целевых таблиц Flink и запуска задания Flink для синхронизации данных.

   ```
   bin/sql-client.sh embedded < flink-create.all.sql
   ```

   После успешного выполнения указанной выше команды задание Flink по синхронизации данных продолжает работать.

10. Наблюдайте за статусом задания Flink.

    ```
    bin/flink list
    ```

    Если при выполнении задания возникает ошибка, вы можете просмотреть подробную информацию об ошибке в логах Flink. Кроме того, вы можете изменить конфигурации Flink в файле **conf/flink-conf.yaml**, такие как память и slot.

Примечания

1. Убедитесь, что Server Agent Service включен.

   Проверьте, работает ли Server Agent Service нормально.

   ```
   EXEC master.dbo.xp_servicecontrol N'QUERYSTATE', N'SQLSERVERAGENT'
   GO
   ```

   Включите Server Agent Service.

   ```
   /opt/mssql/bin/mssql-conf set sqlagent.enabled true
   ```

2. Убедитесь, что CDC для соответствующей базы данных включен.

   Проверьте, включен ли CDC для соответствующей базы данных.

```
select is_cdc_enabled, name from sys.databases where name =
'XXX_databases'
GO
```

   Включите CDC.

   :::note

   При выполнении этой команды убедитесь, что `serverRole` пользователя — `sysadmin`.

   :::

```
USE XXX_databases
GO
EXEC sys.sp_cdc_enable_db
GO
```

3. Убедитесь, что CDC для соответствующей таблицы включен.

```
EXEC sys.sp_cdc_enable_table
@source_schema = 'XXX_schema',
@source_name = 'XXX_table',
@role_name = NULL,
@supports_net_changes = 0;
GO
```

Синхронизация TiDB с Selena

Введение

Flink CDC connector и SMT могут синхронизировать данные из TiDB в течение доли секунды.

SMT может автоматически генерировать DDL-операторы для исходных и целевых таблиц Flink на основе информации о cluster и структуре таблиц TiDB и Selena.

Flink CDC connector захватывает данные путем прямого чтения полных и инкрементных данных из базового хранилища TiKV. Полные данные получаются из диапазонов, разделенных по ключам, а инкрементные данные получаются с использованием CDC Client, предоставляемого TiDB. Затем данные записываются в Selena через Flink-connector-selena.

Шаги

1. Скачайте Flink. Поддерживается версия Flink 1.11 или более поздняя.

2. Скачайте Flink CDC connector. Убедитесь, что вы скачали **flink-sql-connector-tidb-cdc-xxx.jar**, соответствующий версии Flink.

3. Скачайте Flink Selena connector.

4. Скопируйте **flink-sql-connector-tidb-cdc-xxx.jar**, **flink-connector-selena-xxx.jar** в **flink-xxx/lib/**.

5. Скачайте smt.tar.gz.

6. Распакуйте и измените файл конфигурации SMT.

```
[db]
host = 127.0.0.1
port = 4000
user = root
password =
# currently available types: `mysql`, `pgsql`, `oracle`, `hive`,
`clickhouse`, `sqlserver`, `tidb`
type = tidb
# # only takes effect on `type == hive`.
# # Available values: kerberos, none, nosasl, kerberos_http, none_http,
zk, ldap
# authentication = kerberos

[other]
# number of backends in Selena
be_num = 3
# `decimal_v3` is supported since Selena-1.18.1
use_decimal_v3 = false
# directory to save the converted DDL SQL
output_dir = ./result

[table-rule.1]
# pattern to match databases for setting properties
database = ^db$
# pattern to match tables for setting properties
table = ^table$
schema = ^.*$

##############################################
### flink sink configurations
### DO NOT set `connector`, `table-name`, `database-name`, they are
auto-generated
##############################################
flink.selena.jdbc-url=jdbc:mysql://192.168.1.1:9030
flink.selena.load-url= 192.168.1.1:8030
flink.selena.username=root
flink.selena.password=
flink.selena.sink.max-retries=10
flink.selena.sink.buffer-flush.interval-ms=15000
flink.selena.sink.properties.format=json
flink.selena.sink.properties.strip_outer_array=true

##############################################
### flink-cdc configuration for `tidb`
##############################################
```

```
# # Only takes effect on TiDB before v2.0.0.
# # TiKV cluster's PD address.
# flink.cdc.pd-addresses = 127.0.0.1:2379
```

7. Выполните **selena-migrate-tool**. Все SQL-скрипты генерируются в директории `result`.

```
$./selena-migrate-tool
$ls result
flink-create.1.sql    smt.tar.gz              selena-create.all.sql
flink-create.all.sql  selena-create.1.sql  selena-external-
create.all.sql
```

8. Используйте SQL-скрипт с префиксом `selena-create` для создания таблицы в Selena.

```
mysql -hxx.xx.xx.x -P9030 -uroot -p < selena-create.all.sql
```

9. Используйте SQL-скрипт с префиксом `flink-create` для создания исходных и целевых таблиц Flink и запуска задания Flink для синхронизации данных.

```
bin/sql-client.sh embedded < flink-create.all.sql
```

После успешного выполнения указанной выше команды задание Flink по синхронизации данных продолжает работать.

10. Наблюдайте за статусом задания Flink.

```
bin/flink list
```

Если при выполнении задания возникает ошибка, вы можете просмотреть подробную информацию об ошибке в логах Flink. Кроме того, вы можете изменить конфигурации Flink в файле **conf/flink-conf.yaml**, такие как память и slot.

Примечания

Для TiDB версии до v2.0.0 требуется дополнительная конфигурация `flink.cdc.pd-addresses`.

```
##############################################
### flink-cdc configuration for `tidb`
##############################################
# # Only takes effect on TiDB before v2.0.0.
# # TiKV cluster's PD address.
# flink.cdc.pd-addresses = 127.0.0.1:2379
```

1.6.6 DataX writer

Введение

Плагин SelenaWriter позволяет записывать данные в целевую таблицу Selena. В частности, SelenaWriter импортирует данные в Selena в формате CSV или JSON через Stream Load и внутренне кэширует и массово импортирует данные, прочитанные `reader`, в Selena для лучшей производительности записи. Общий поток данных: `source -> Reader -> DataX channel -> Writer -> Selena`.

Скачать плагин

Перейдите на `https://github.com/alibaba/DataX`, чтобы скачать полный пакет DataX, и поместите плагин selenawriter в директорию `datax/plugin/writer/`.

Используйте следующую команду для тестирования: `python datax.py --jvm="-Xms6G -Xmx6G" --loglevel=debug job.json`

Описание функциональности

Пример конфигурации

Вот файл конфигурации для чтения данных из MySQL и загрузки в Selena.

```
{
    "job": {
        "setting": {
            "speed": {
                 "channel": 1
            },
            "errorLimit": {
                "record": 0,
                "percentage": 0
            }
        },
        "content": [
            {
                "reader": {
                    "name": "mysqlreader",
                    "parameter": {
                        "username": "xxxx",
                        "password": "xxxx",
                        "column": [ "k1", "k2", "v1", "v2" ],
                        "connection": [
                            {
                                "table": [ "table1", "table2" ],
                                "jdbcUrl": [

"jdbc:mysql://127.0.0.1:3306/datax_test1"
                                ]
                            },
                            {
                                "table": [ "table3", "table4" ],
                                "jdbcUrl": [

"jdbc:mysql://127.0.0.1:3306/datax_test2"
                                ]
                            }
                        ]
                    }
                },
               "writer": {
                    "name": "selenawriter",
                    "parameter": {
                        "username": "xxxx",
                        "password": "xxxx",
                        "database": "xxxx",
                        "table": "xxxx",
                        "column": ["k1", "k2", "v1", "v2"],
```

```
                        "preSql": [],
                        "postSql": [],
                        "jdbcUrl": "jdbc:mysql://172.28.17.100:9030/",
                        "loadUrl": ["172.28.17.100:8030",
"172.28.17.100:8030"],
                        "loadProps": {}
                    }
                }
            }
        ]
    }
}
```

Описание параметров Selenawriter

- **username**
  - Описание: Имя пользователя базы данных Selena
  - Обязательный: Да
  - Значение по умолчанию: нет
- **password**
  - Описание: Пароль для базы данных Selena
  - Обязательный: Да
  - Значение по умолчанию: нет
- **database**
  - Описание: Имя базы данных для таблицы Selena
  - Обязательный: Да
  - Значение по умолчанию: нет
- **table**

- Описание: Имя таблицы в Selena
- Обязательный: Да
- Значение по умолчанию: нет

- **loadUrl**
  - Описание: Адрес FE Selena для stream load, может быть несколько адресов FE в формате `fe_ip:fe_http_port`.
  - Обязательный: да
  - Значение по умолчанию: нет

- **column**
  - Описание: Поля целевой таблицы, **в которые необходимо записать данные**, столбцы разделяются запятыми. Пример: "column": ["id", "name", "age"].

    > **Параметр конфигурации column должен быть указан и не может быть пустым.** Примечание: Мы настоятельно не рекомендуем оставлять его пустым, потому что ваше задание может работать неправильно или завершиться с ошибкой при изменении количества столбцов, типа и т. д. целевой таблицы. Элементы конфигурации должны быть в том же порядке, что и querySQL или column в reader.

- Обязательный: Да
- Значение по умолчанию: нет
- **preSql**
- Описание: Стандартный оператор, который будет выполнен перед записью данных в целевую таблицу.
- Обязательный: Нет
- Значение по умолчанию: нет
- **jdbcUrl**

  - Описание: Информация о JDBC-подключении к целевой базе данных для выполнения `preSql` и `postSql`.
  - Обязательный: Нет
- Значение по умолчанию: нет
- **loadProps**
  - Описание: Параметры запроса для StreamLoad, подробности см. на странице введения в StreamLoad.
  - Обязательный: Нет
  - Значение по умолчанию: нет

Преобразование типов

По умолчанию входящие данные преобразуются в строки с `t` в качестве разделителя столбцов и `n` в качестве разделителя строк для формирования `csv` файлов для импорта StreamLoad. Чтобы изменить разделитель столбцов, настройте `loadProps` соответствующим образом.

```
"loadProps": {
    "column_separator": "\\x01",
    "row_delimiter": "\\x02"
}
```

Чтобы изменить формат импорта на `json`, настройте `loadProps` соответствующим образом.

```
"loadProps": {
    "format": "json",
    "strip_outer_array": true
}
```

> Формат `json` используется для импорта данных в Selena в формате JSON.

О часовом поясе

Если исходная библиотека находится в другом часовом поясе, при выполнении datax.py добавьте следующий параметр после команды

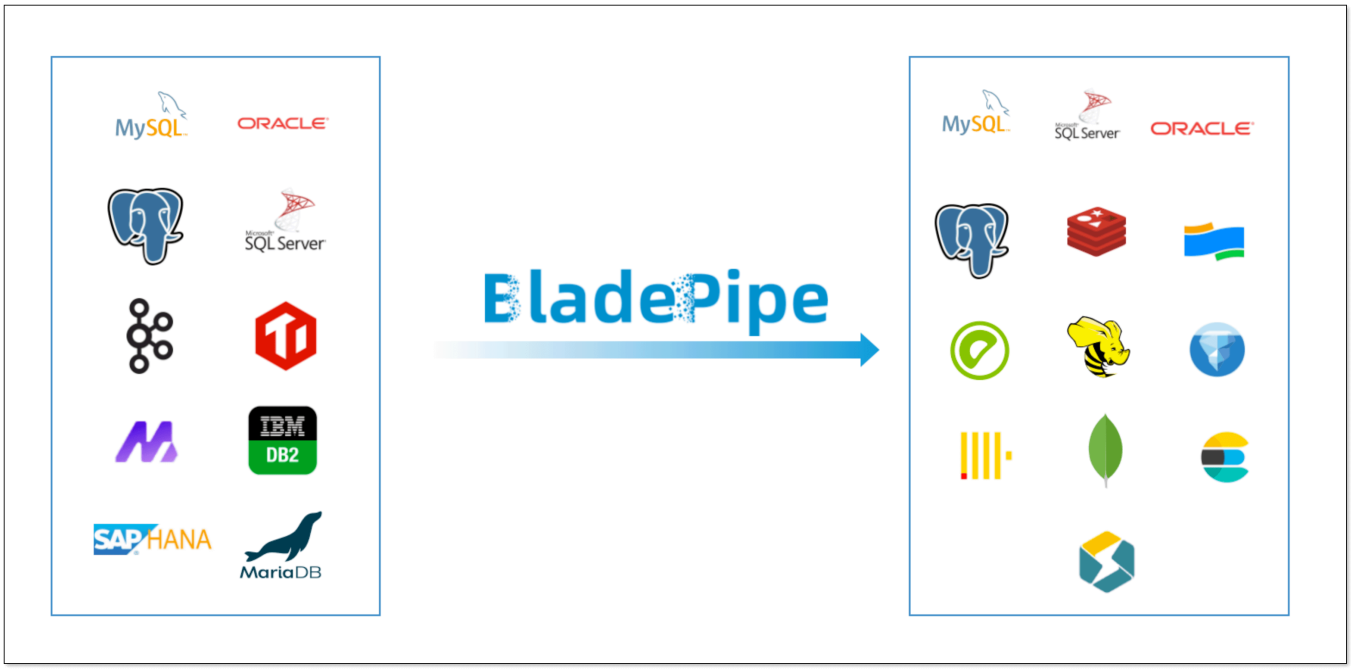
```
"-Duser.timezone=xx"
```

Например, если DataX импортирует данные PostgreSQL и исходная библиотека находится в часовом поясе UTC, добавьте параметр "-Duser.timezone=GMT+0" при запуске.

1.6.7 BladePipe

Введение

BladePipe — это **инструмент для репликации данных в режиме реального времени**, перемещающий данные между **более чем 30** базами данных, очередями сообщений, поисковыми системами, системами кэширования, хранилищами данных реального времени, озерами данных и многим другим с **сверхнизкой задержкой**. Он отличается эффективностью, стабильностью и масштабируемостью, совместимостью с различными движками баз данных, управлением в одном месте, повышенной безопасностью и сложными преобразованиями данных. BladePipe помогает разрушать силосы данных, увеличивая ценность данных.



Функции

BladePipe предоставляет визуализированный интерфейс управления, позволяющий легко создавать DataJobs для достижения **миграции схем, миграции данных, синхронизации, проверки и исправления** и многого другого. Кроме того, он поддерживает более точные и настраиваемые конфигурации через настройки параметров. Теперь BladePipe поддерживает перемещение данных из следующих источников DataSources в Selena:

Источник DataSource	Миграция схемы	Миграция данных	Синхронизация данных	Проверка и исправление
MySQL/MariaDB/AuroraMySQL	Да	Да	Да	Да
Oracle	Да	Да	Да	Да
PostgreSQL/AuroraPostgreSQL	Да	Да	Да	Да
SQL Server	Да	Да	Да	Да
Kafka	Нет	Нет	Да	Нет
AutoMQ	Нет	Нет	Да	Нет
TiDB	Да	Да	Да	Да
Hana	Да	Да	Да	Да
PolarDB for MySQL	Да	Да	Да	Да
Db2	Да	Да	Да	Да

К СВЕДЕНИЮ

Для получения дополнительной информации о поддерживаемых функциях и настройках параметров обратитесь к BladePipe Connections.

Установка

Следуйте инструкциям в Install BladePipe Worker (Docker) или Install BladePipe Worker (Binary) для загрузки и установки BladePipe Worker.

Пример

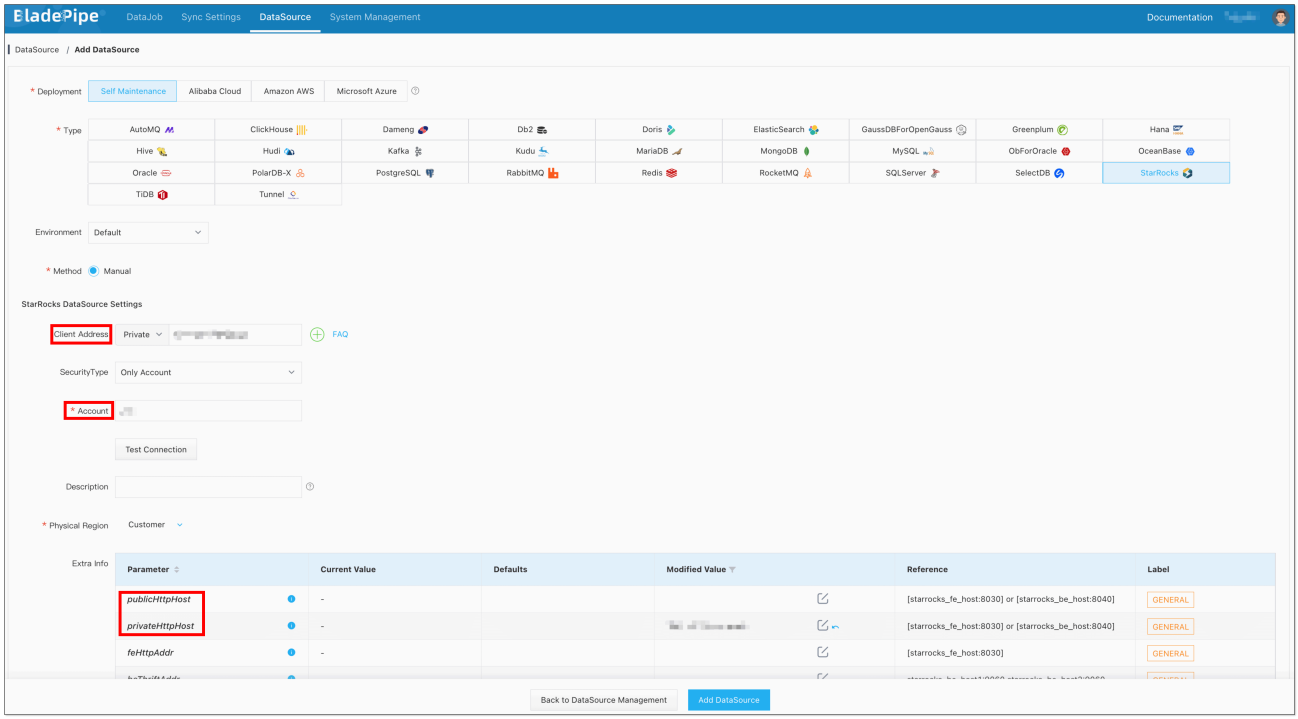
Используя экземпляр MySQL в качестве примера, в следующем разделе описывается, как перенести данные из MySQL в Selena.

Добавление DataSources

1. Войдите в BladePipe Cloud. Нажмите **DataSource** > **Add DataSource**.

2. Выберите **Selena** в качестве **Type** и заполните форму настройки.

   - **Client Address**: Порт, который Selena предоставляет для MySQL Client. BladePipe запрашивает метаданные в базах данных через него.
   - **Account**: Имя пользователя базы данных Selena. Для записи данных в Selena требуется привилегия INSERT. Следуйте инструкциям в GRANT, чтобы предоставить пользователю базы данных привилегию INSERT.
   - **Http Address**: Порт, используемый для приема запроса от BladePipe для записи данных в Selena.



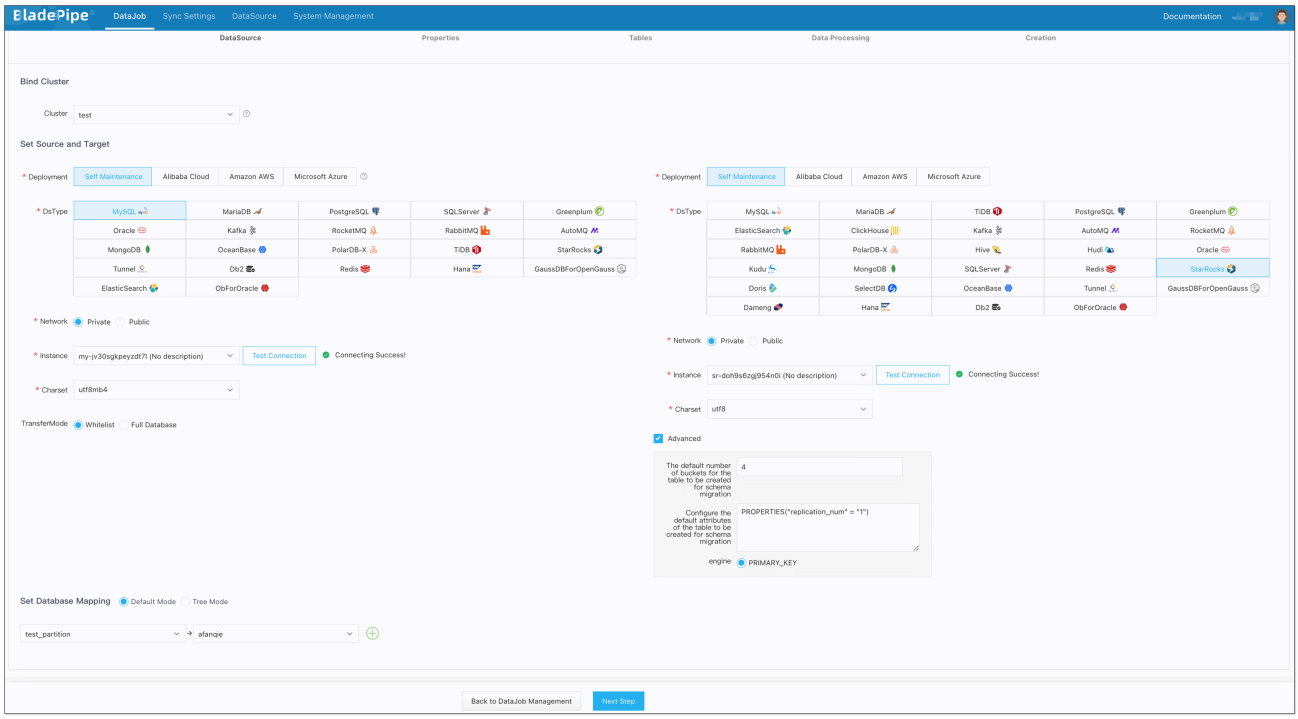
3. Нажмите **Test Connection**. После успешного подключения нажмите **Add DataSource**, чтобы добавить DataSource.

4. Добавьте MySQL DataSource, следуя вышеуказанным шагам.

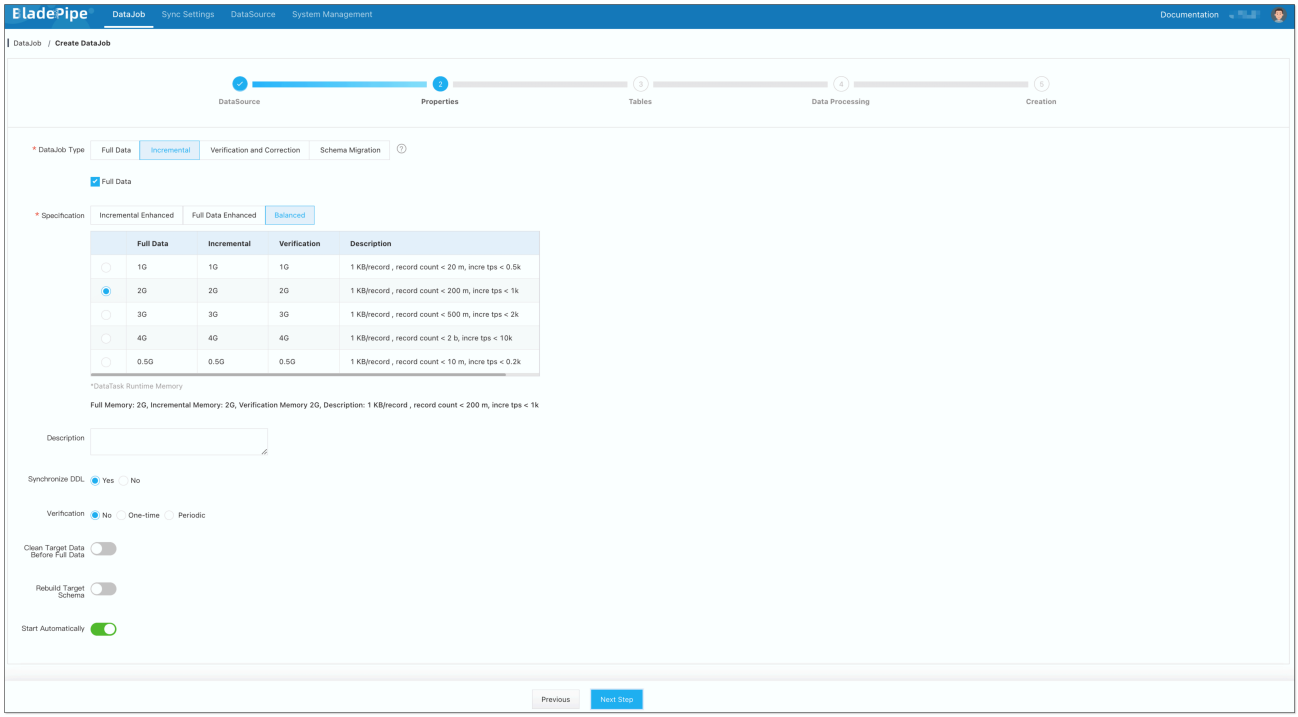
Создание DataJob

1. Нажмите **DataJob** > **Create DataJob**.

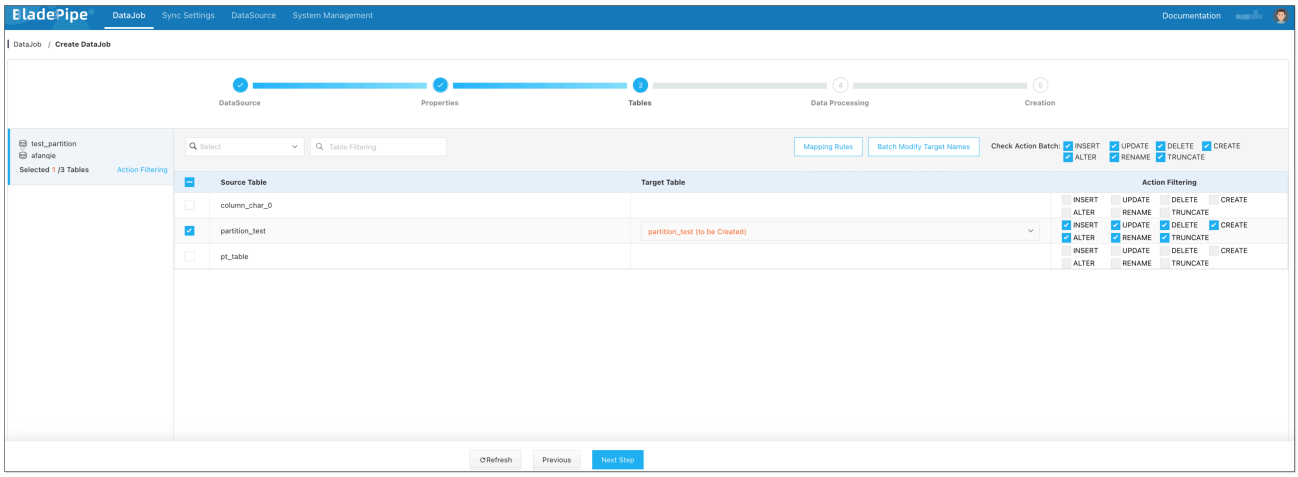
2. Выберите исходный и целевой DataSources и нажмите **Test Connection**, чтобы убедиться, что подключение к исходному и целевому DataSources успешно.



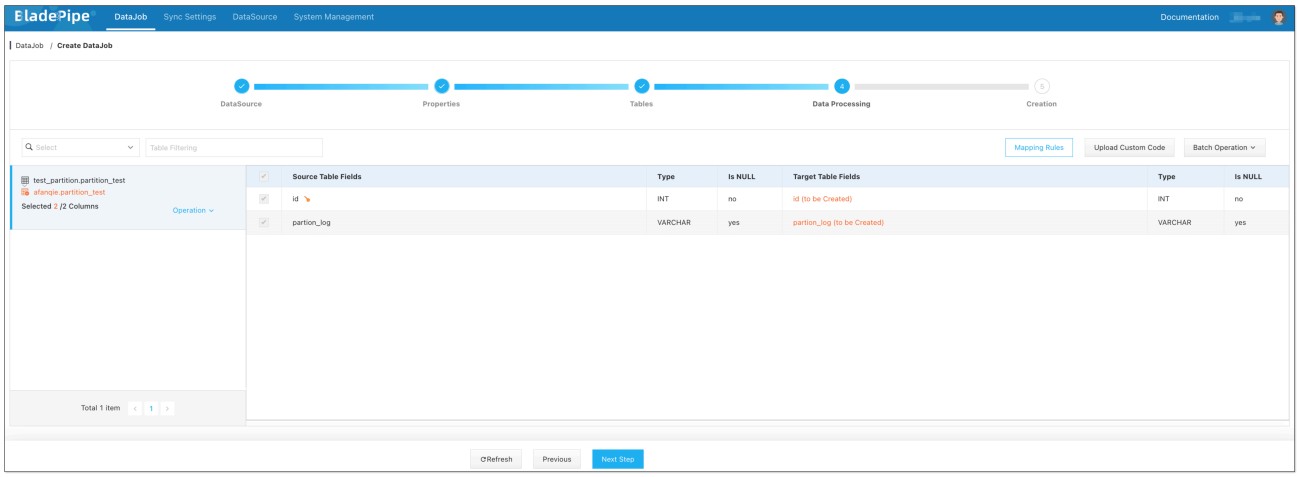
3. Выберите **Incremental** для типа DataJob вместе с опцией **Full Data**.



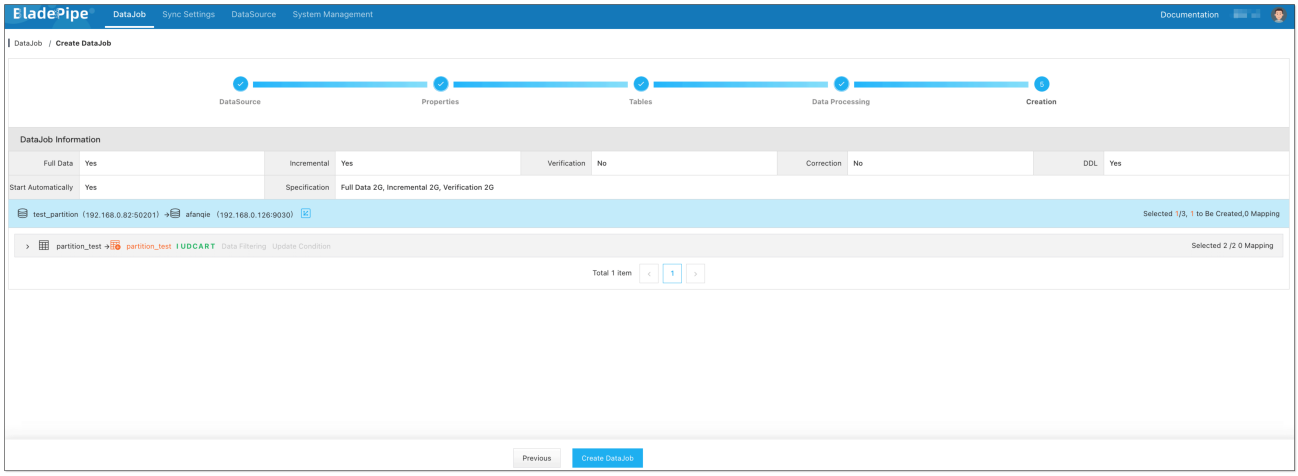
4. Выберите таблицы для репликации. **Обратите внимание, что целевые таблицы Selena, автоматически созданные после миграции схемы, имеют первичные ключи, поэтому исходные таблицы без первичных ключей в настоящее время не поддерживаются**.



5. Выберите столбцы для репликации.

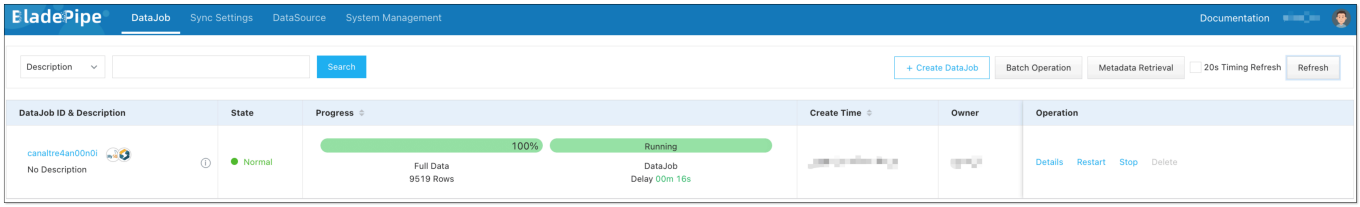


6. Подтвердите создание DataJob.



7. DataJob запускается автоматически. BladePipe автоматически запустит следующие DataTasks:

- **Schema Migration**: Схемы исходных таблиц будут перенесены в целевой экземпляр.
- **Full Data**: Все существующие данные исходных таблиц будут полностью перенесены в целевой экземпляр.
- **Incremental**: Текущие изменения данных будут непрерывно синхронизироваться с целевым экземпляром (с задержкой менее минуты).



1.7 Business Intelligence

> Совет: Если BI-инструмент полагается на максимальную длину строки в метаданных запроса, и эта длина меньше фактических значений (что вызывает пустые возвраты), установите глобальную переменную `max_unknown_string_meta_length` (1~1,048,576, по умолчанию 64) на более высокое значение, чтобы строковые колонки возвращали эту длину в метаданных.

1.7.1 Apache Superset

Apache Superset поддерживает запросы и визуализацию как внутренних, так и внешних данных в Selena.

Предварительные требования

Убедитесь, что вы выполнили следующие установки:

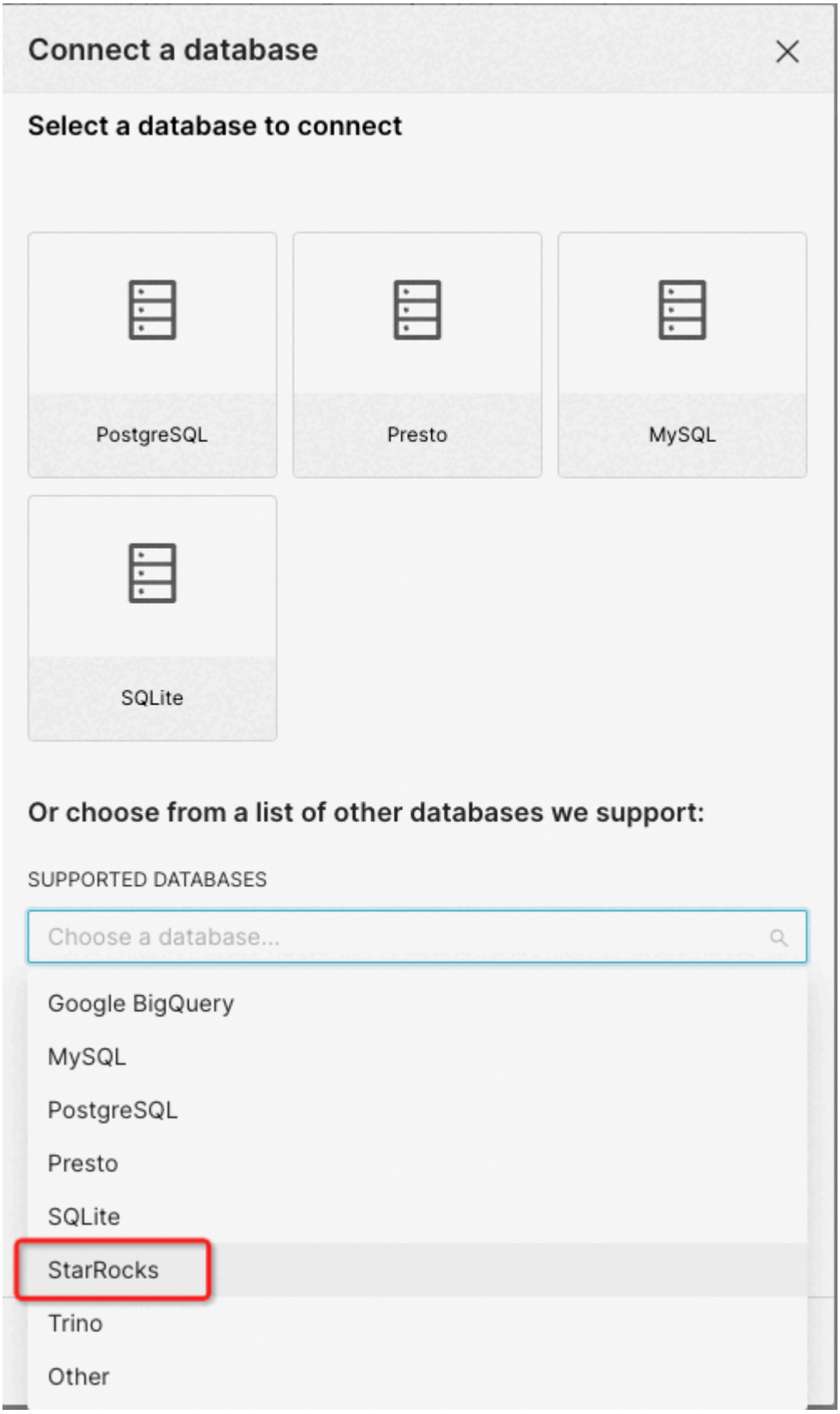
1. Установите Python-клиент для Selena на сервере Apache Superset.

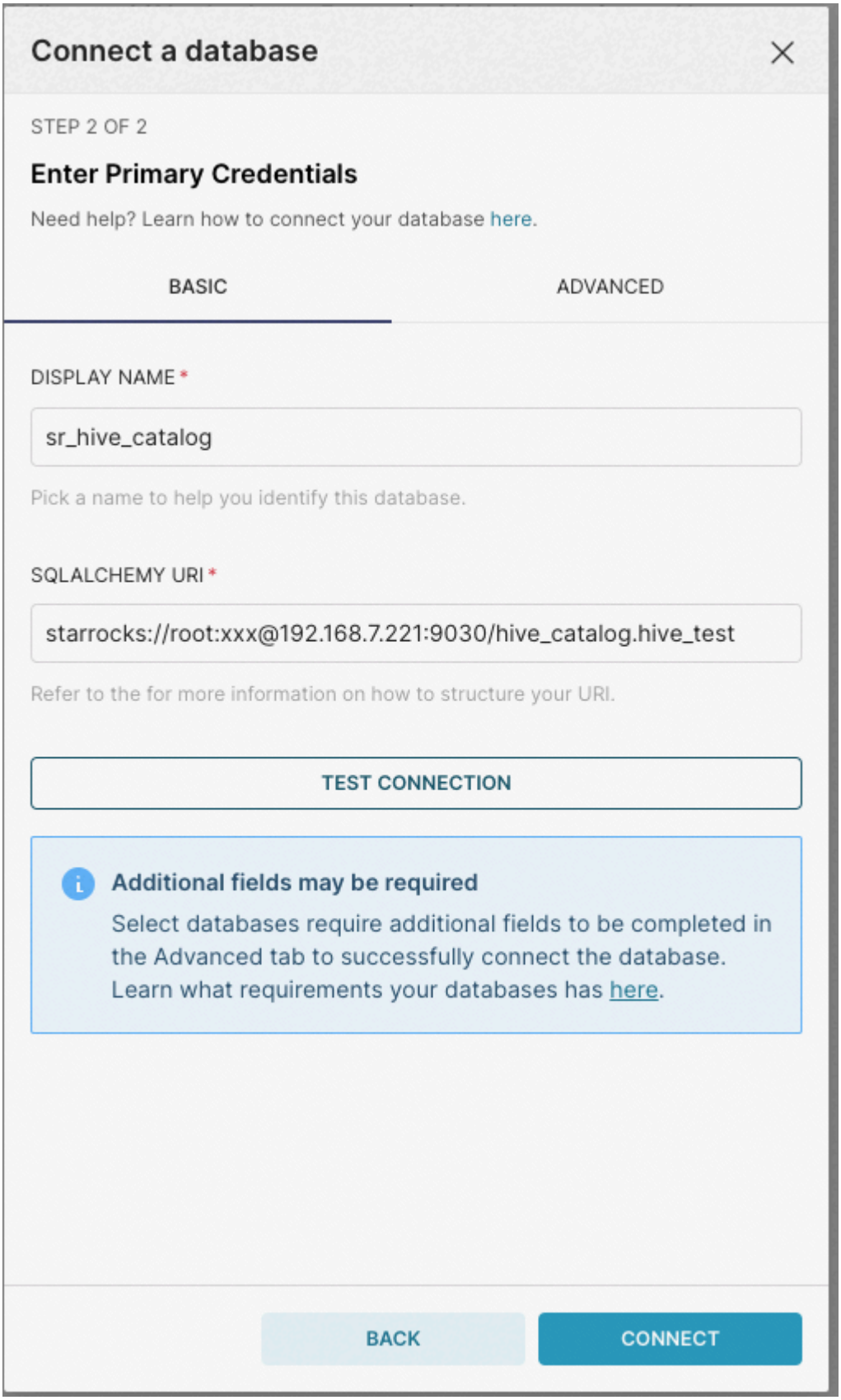
```
pip install selena
```

2. Установите последнюю версию Apache Superset. Подробнее см. Установка Superset.

Интеграция

Создайте базу данных в Apache Superset:


Connect a database
Select a database to connect
PostgreSQL
Presto
MySQL
SQLite
Or choose from a list of other databases we support:
SUPPORTED DATABASES
Choose a database...
Google BigQuery
MySQL
PostgreSQL
Presto
SQLite
StarRocks
Trino
Other



Обратите внимание на следующие моменты:

- Для **SUPPORTED DATABASES** выберите **Selena**, который будет использоваться в качестве источника данных.

- Для **SQLALCHEMY URI** введите URI в формате Selena SQLAlchemy URI, как показано ниже:

  ```
  selena://<User>:<Password>@<Host>:<Port>/<Catalog>.<Database>
  ```

  Параметры в URI описаны ниже:

  - `User`: имя пользователя для входа в ваш cluster Selena, например, `admin`.
  - `Password`: пароль для входа в ваш cluster Selena.
  - `Host`: IP-адрес хоста FE вашего cluster Selena.
  - `Port`: порт запросов FE вашего cluster Selena, например, `9030`.
  - `Catalog`: целевой каталог в вашем cluster Selena. Поддерживаются как внутренние, так и внешние каталоги.
  - `Database`: целевая база данных в вашем cluster Selena. Поддерживаются как внутренние, так и внешние базы данных.

1.7.2 FineBI

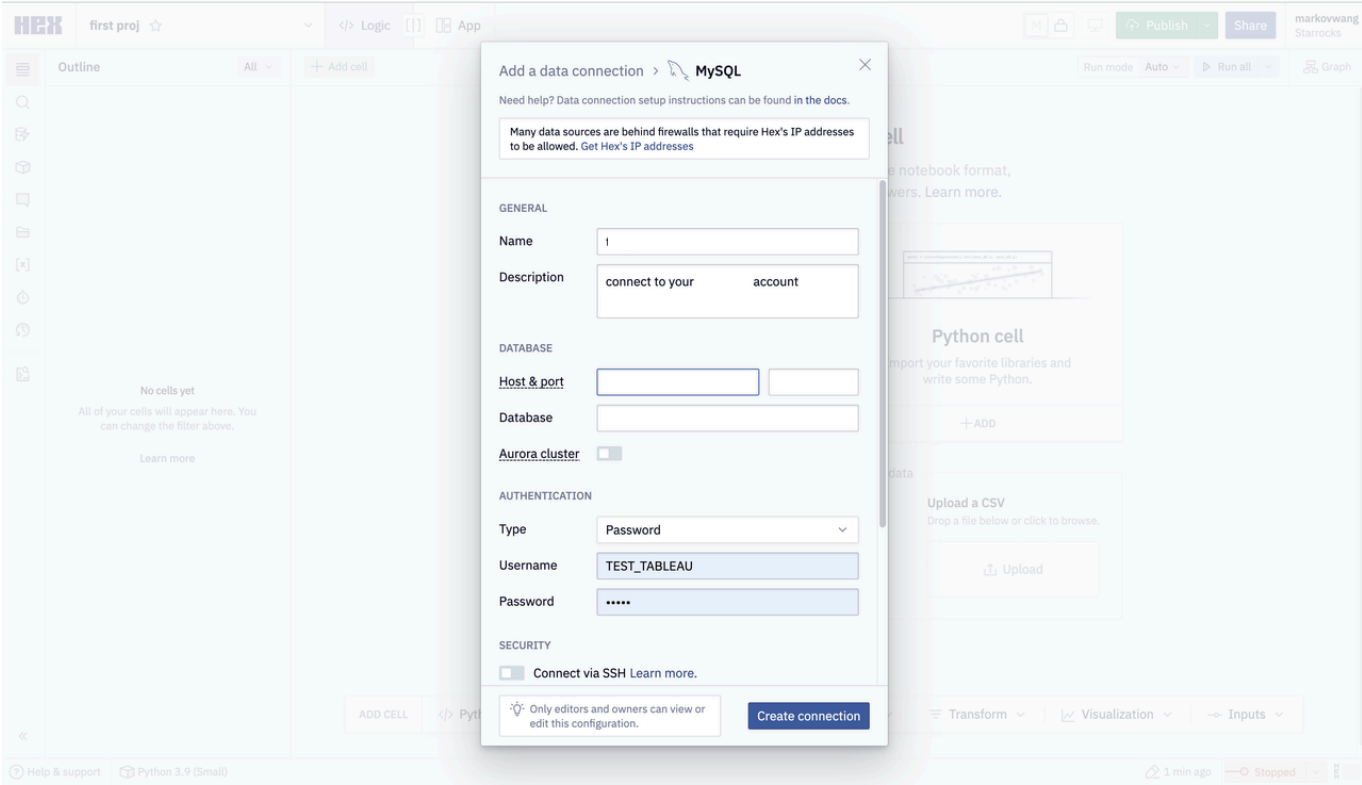
В этом разделе описывается, как подключить FineBI к Selena для выполнения визуального анализа данных Selena в FineBI.

Подробнее о подключении к Selena в FineBI см. Подключение к данным Selena.

1.7.3 Hex

Hex поддерживает запросы и визуализацию как внутренних, так и внешних данных в Selena.

Добавьте подключение к данным в Hex. Обратите внимание, что в качестве типа подключения необходимо выбрать MySQL.



Параметры, которые необходимо настроить, описаны ниже:

- **Name**: имя подключения к данным.
- **Host & port**: IP-адрес хоста FE и порт запросов FE вашего cluster Selena. Пример порта запросов: `9030`.
- **Database**: источник данных, к которому вы хотите получить доступ в вашем cluster Selena. Значение этого параметра имеет формат `<catalog_name>.<database_name>`.
  - `catalog_name`: имя целевого каталога в вашем cluster Selena. Поддерживаются как внутренние, так и внешние каталоги.
  - `database_name`: имя целевой базы данных в вашем cluster Selena. Поддерживаются как внутренние, так и внешние базы данных.
- **Type**: метод аутентификации, который вы хотите использовать. Выберите **Password**.
- **User**: имя пользователя для входа в ваш cluster Selena, например, `admin`.
- **Password**: пароль для входа в ваш cluster Selena.

1.7.4 Querybook

Querybook поддерживает запросы и визуализацию как внутренних, так и внешних данных в Selena.

Предварительные требования

Убедитесь, что вы выполнили следующие подготовительные действия:

1. Клонируйте и загрузите репозиторий Querybook.

```
git clone git@github.com:pinterest/querybook.git
cd querybook
```

2. Создайте файл с именем `local.txt` в папке `requirements` в корневом каталоге проекта.

```
touch requirements/local.txt
```

3. Добавьте необходимые пакеты.

```
echo -e "selena\nmysqlclient" > requirements/local.txt
```

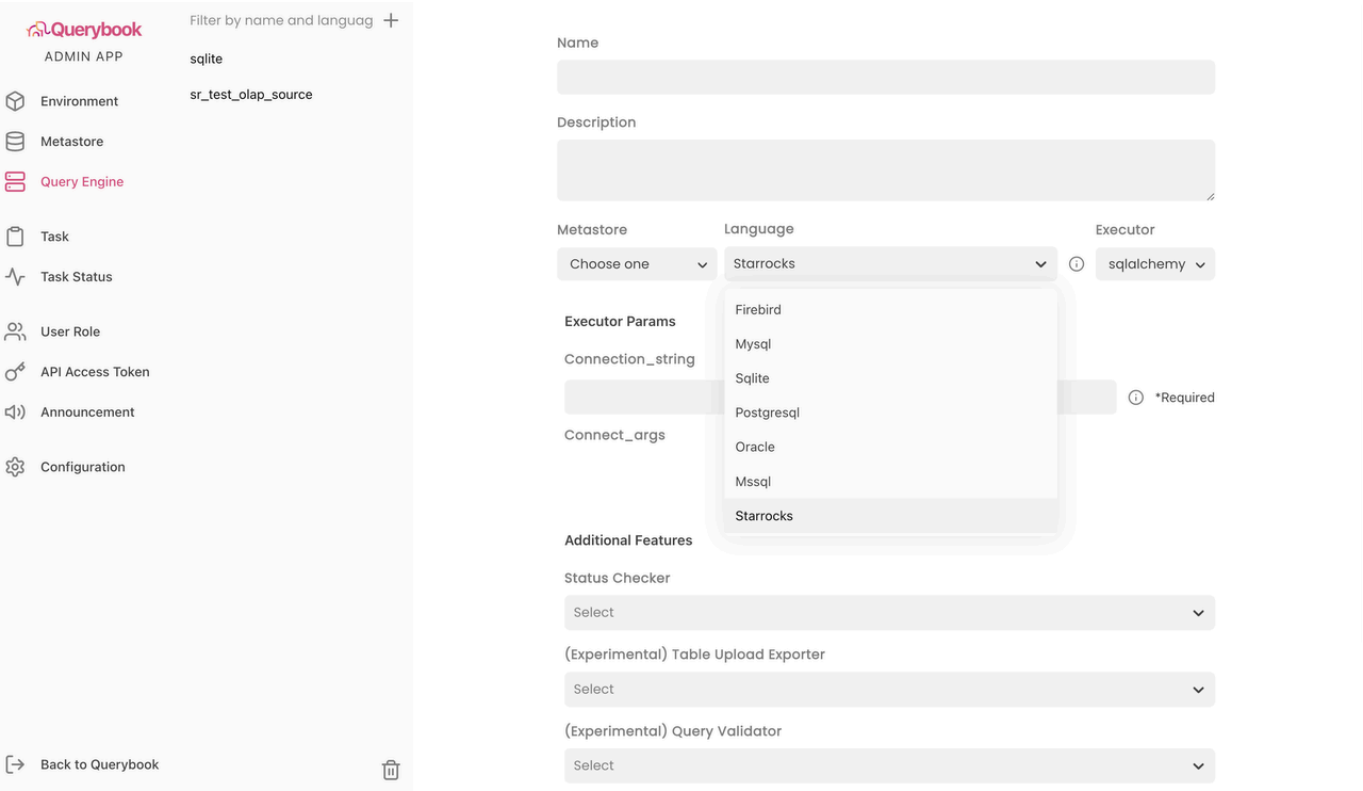
4. Запустите контейнер.

```
make
```

Интеграция

Перейдите по следующему URL и добавьте новый движок запросов:

```
https://localhost:10001/admin/query_engine/
```



Обратите внимание на следующие моменты:

- Для **Language** выберите **Selena**.
- Для **Executor** выберите **sqlalchemy**.
- Для **Connection_string** введите URI в формате Selena SQLAlchemy URI, как показано ниже:

  ```
  selena://<User>:<Password>@<Host>:<Port>/<Catalog>.<Database>
  ```

  Параметры в URI описаны ниже:

  - `User`: имя пользователя для входа в ваш cluster Selena, например, `admin`.
  - `Password`: пароль для входа в ваш cluster Selena.
  - `Host`: IP-адрес хоста FE вашего cluster Selena.
  - `Port`: порт запросов FE вашего cluster Selena, например, `9030`.
  - `Catalog`: целевой каталог в вашем cluster Selena. Поддерживаются как внутренние, так и внешние каталоги.
  - `Database`: целевая база данных в вашем cluster Selena. Поддерживаются как внутренние, так и внешние базы данных.

1.7.5 QuickBI

В этом разделе описывается, как подключить QuickBI к Selena для выполнения визуального анализа данных Selena в QuickBI.

Подробнее о подключении к Selena в QuickBI см. Добавление пользовательского источника данных Selena.

1.7.6 Tableau

В этом разделе описывается, как подключить Selena к Tableau Desktop и Tableau Server с помощью Selena Tableau JDBC Connector.

Обзор

Selena Tableau JDBC Connector - это пользовательское расширение для Tableau Desktop и Tableau Server. Оно упрощает процесс подключения Tableau к Selena и улучшает поддержку стандартной функциональности Tableau, превосходя стандартное подключение Generic ODBC/JDBC.

Ключевые особенности

- Поддержка LDAP: Позволяет выполнять вход через LDAP с запросом пароля для безопасной аутентификации.
- Высокая совместимость: Достигает 99.99% совместимости при тестировании TDVT (Tableau Design Verification Tool), с только одним незначительным случаем сбоя.

Предварительные требования

Перед началом работы убедитесь, что выполнены следующие требования:

- Версия Tableau: Tableau 2020.4 и выше
- Версия Selena: v1.5.2 и выше

Установка коннектора для Tableau Desktop

1. Загрузите MySQL JDBC Driver 8.0.33.

2. Сохраните файл драйвера в следующем каталоге (создайте каталог, если он не существует):

    - macOS: `~/Library/Tableau/Drivers`

- Windows: `C:\Program Files\Tableau\Drivers`

3. Загрузите файл Selena Tableau JDBC Connector.

4. Сохраните файл коннектора в следующем каталоге:

   - macOS: `~/Documents/My Tableau Repository/Connectors`
   - Windows: `C:\Users\[Windows User]\Documents\My Tableau Repository\Connectors`

5. Запустите Tableau Desktop.

6. Перейдите в **Connect** -> **To a Server** -> **Selena JDBC by CelerData**.

Установка коннектора для Tableau Server

1. Загрузите MySQL JDBC Driver 8.0.33.

2. Сохраните файл драйвера в следующем каталоге (создайте каталог, если он не существует):

   - Linux: `/opt/tableau/tableau_driver/jdbc`
   - Windows: `C:\Program Files\Tableau\Drivers`

К СВЕДЕНИЮ

В Linux необходимо разрешить пользователю "Tableau" доступ к каталогу.

Выполните следующие шаги:

i. Создайте каталог и скопируйте файл драйвера в каталог:

```
sudo mkdir -p /opt/tableau/tableau_driver/jdbc

# Замените <path_to_driver_file_name> на абсолютный путь к
файлу драйвера.
sudo cp /<path_to_driver_file_name>.jar
/opt/tableau/tableau_driver/jdbc
```

ii. Предоставьте разрешение пользователю "Tableau".

```
# Замените <driver_file_name> на имя файла драйвера.
sudo chmod 755
/opt/tableau/tableau_driver/jdbc/<driver_file_name>.jar
```

3. Загрузите файл Selena Tableau JDBC Connector.

4. Сохраните файл коннектора в следующем каталоге на каждом узле:

   - Linux: `/opt/tableau/connectors`
   - Windows: `C:\Program Files\Tableau\Connectors`

5. Перезапустите Tableau Server.

```
tsm restart
```

К СВЕДЕНИЮ

Необходимо перезапускать Tableau Server для применения изменений при каждом добавлении, удалении или обновлении коннектора.

Примечания по использованию

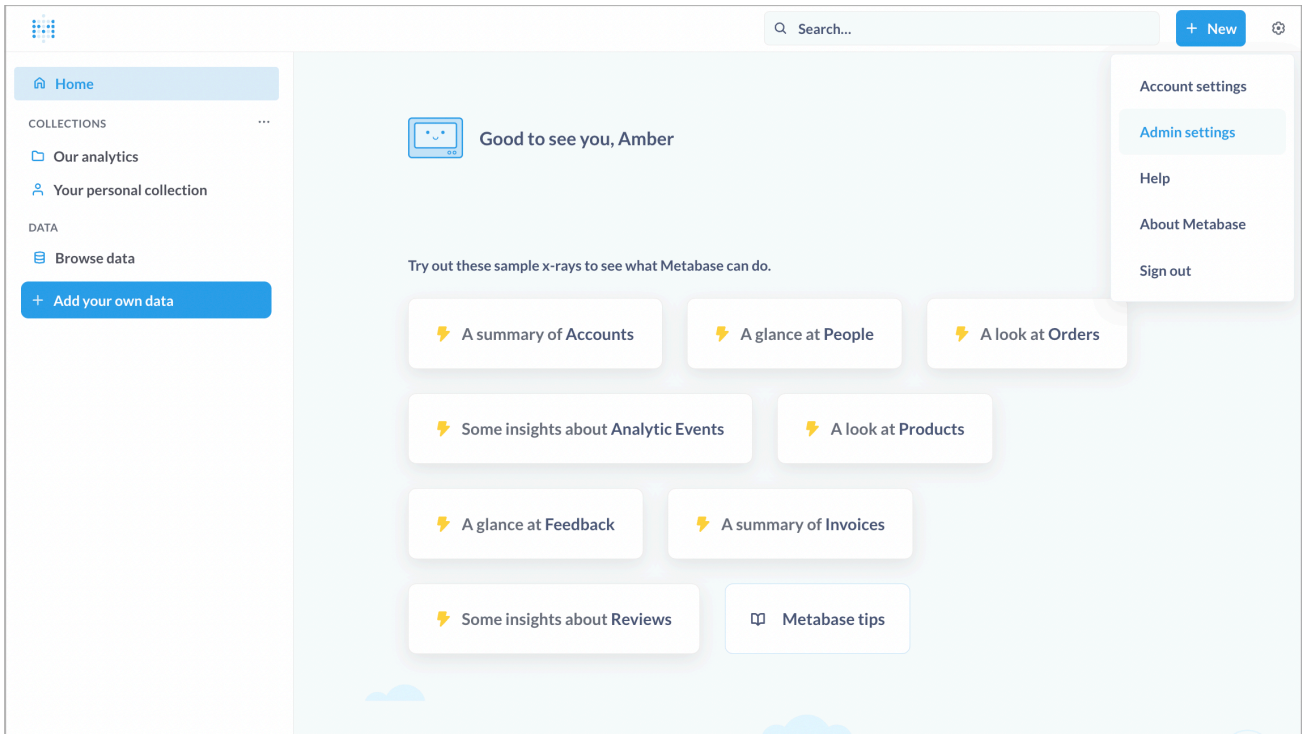
Если требуется поддержка входа через LDAP, вы можете включить переключатель **Enable LDAP** на вкладке **Advanced** во время настройки.

1.7.7 Metabase

Metabase поддерживает запросы и визуализацию как внутренних, так и внешних данных в Selena.

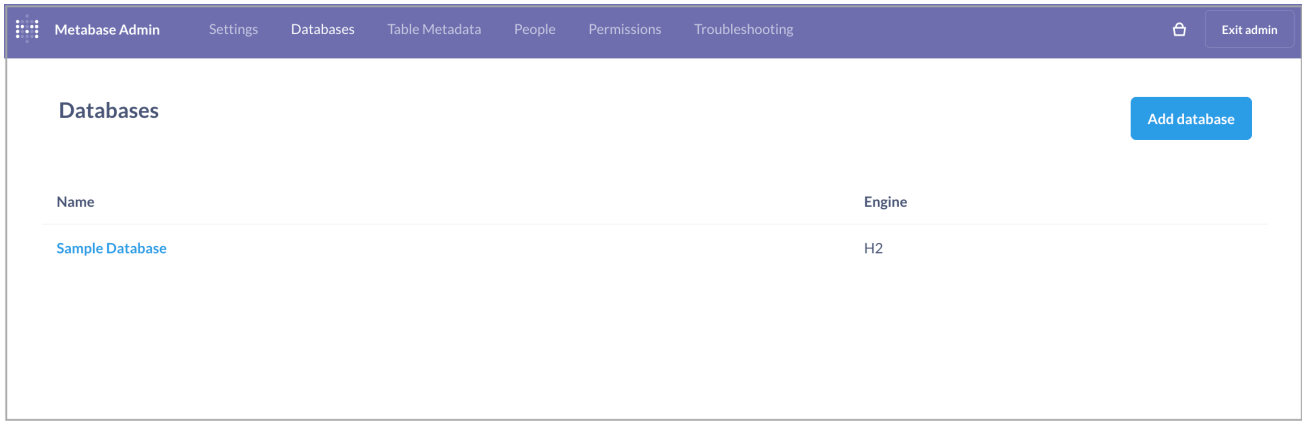
Запустите Metabase и выполните следующие действия:

1. В правом верхнем углу домашней страницы Metabase нажмите значок **Settings** и выберите **Admin settings**.



2. Выберите **Databases** в верхнем меню.

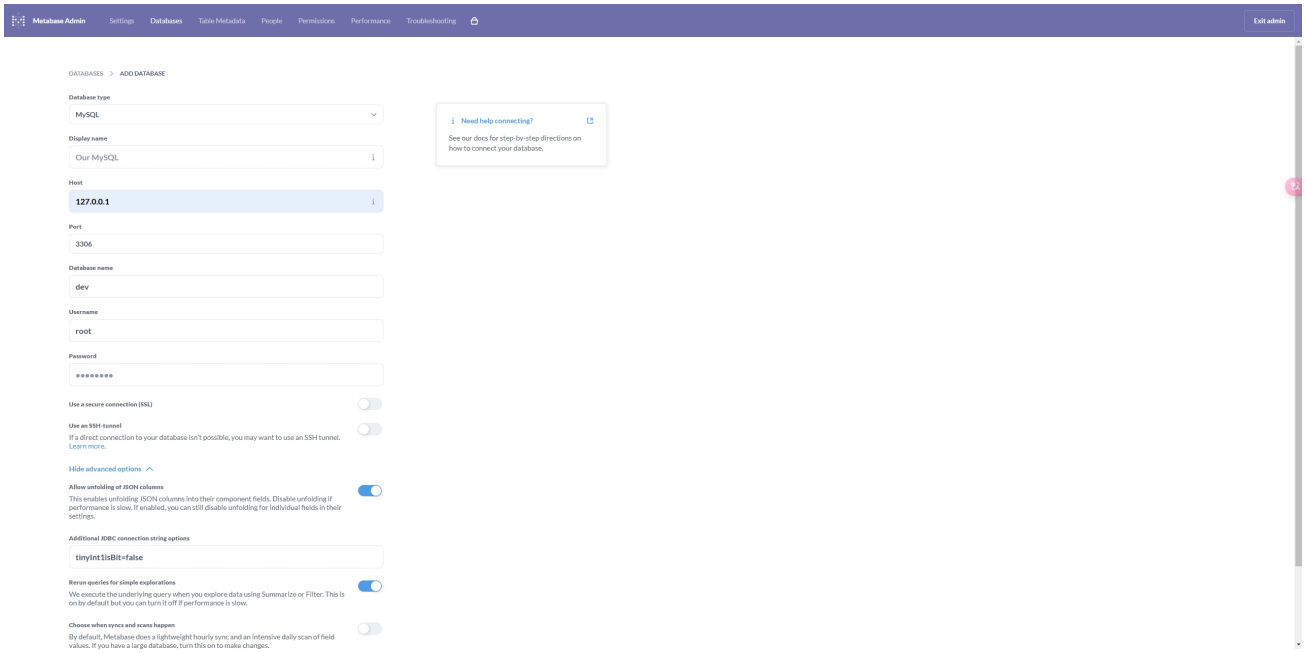
3. На странице **Databases** нажмите **Add database**.



4. На появившейся странице настройте параметры базы данных и нажмите **Save**.

   - **Database type**: Выберите **MySQL**.
   - **Host** и **Port**: Введите информацию о хосте и порте, соответствующую вашему случаю использования.
   - **Database name**: Введите имя базы данных в формате `<catalog_name>.<database_name>`. В версиях Selena ранее v1.5.2 вы можете интегрировать с Metabase только внутренний каталог вашего cluster Selena. Начиная с версии Selena v1.5.2 вы можете интегрировать с Metabase как внутренний каталог, так и внешние каталоги вашего cluster Selena.
   - **Username** и **Password**: Введите имя пользователя и пароль вашего пользователя cluster Selena.
   - **Additional JDBC connection string options**: В этом поле необходимо добавить свойство `tinyInt1isBit=false`. В противном случае может возникнуть ошибка.

   Остальные параметры не относятся к Selena. Настройте их в соответствии с вашими бизнес-потребностями.



ПРИМЕЧАНИЕ: Избегайте использования типов данных DECIMAL, так как Metabase не понимает этот специфический для Selena тип данных столбца.

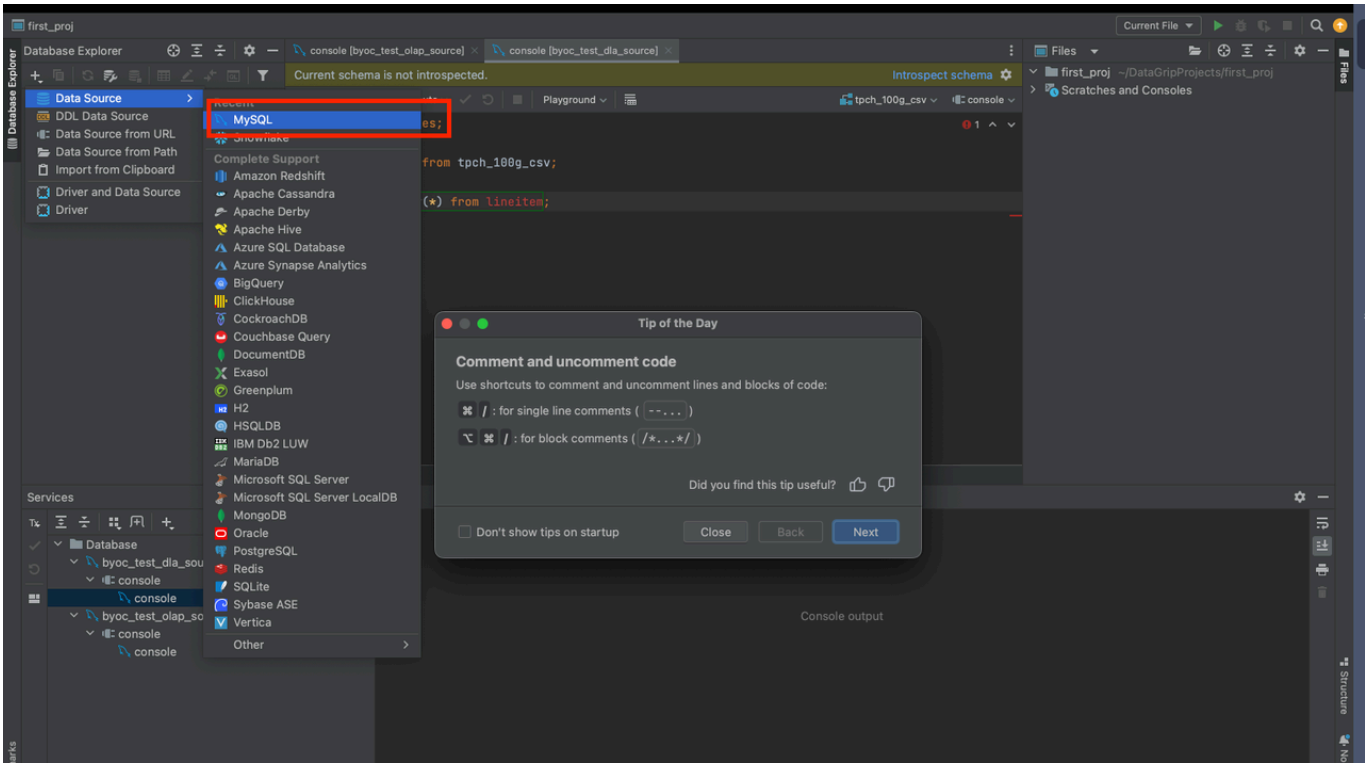
1.8 IDE-инструменты

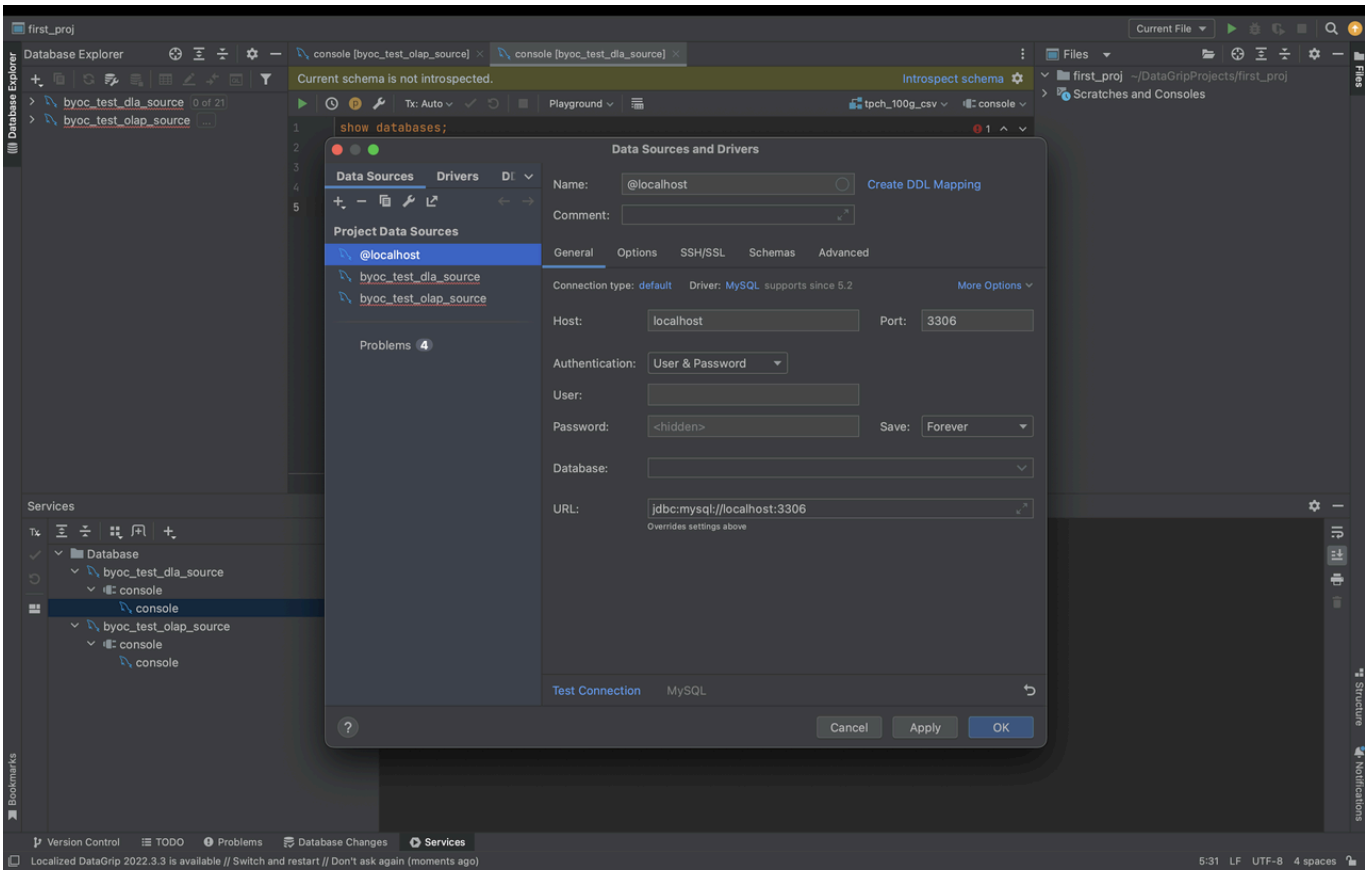
1.8.1 DataGrip

DataGrip поддерживает запросы как к внутренним, так и к внешним данным в Selena.

> **ПОДСКАЗКА**
>
> Документация DataGrip

Создайте источник данных в DataGrip. Обратите внимание, что в качестве источника данных необходимо выбрать MySQL.





Параметры, которые необходимо настроить, описаны ниже:

- **Host**: IP-адрес хоста FE вашего cluster Selena.
- **Port**: порт запросов FE вашего cluster Selena, например, `9030`.

- **Authentication**: метод аутентификации, который вы хотите использовать. Выберите **Username & Password**.
- **User**: имя пользователя для входа в ваш cluster Selena, например, `admin`.
- **Password**: пароль для входа в ваш cluster Selena.
- **Database**: источник данных, к которому вы хотите получить доступ в вашем cluster Selena. Значение этого параметра имеет формат `<catalog_name>.<database_name>`.
  - `catalog_name`: имя целевого каталога в вашем cluster Selena. Поддерживаются как внутренние, так и внешние каталоги.
  - `database_name`: имя целевой базы данных в вашем cluster Selena. Поддерживаются как внутренние, так и внешние базы данных.

1.8.2 Dataphin

Dataphin - это облачная реализация внутренних практик методологии управления данными OneData группы Alibaba. Он предоставляет комплексное решение для интеграции, построения, управления и использования данных на протяжении всего жизненного цикла больших данных, помогая предприятиям значительно повысить уровень управления данными и создать корпоративную платформу данных с высоким и надёжным качеством, удобным потреблением, безопасным и экономичным производством. Dataphin обеспечивает поддержку различных вычислительных платформ и расширяемые открытые возможности для соответствия архитектуре технической платформы и специфическим требованиям предприятий в различных отраслях.

Существует несколько способов интеграции Dataphin с Selena:

- В качестве источника или целевого источника данных для интеграции данных. Данные могут читаться из Selena и передаваться в другие источники данных, или данные могут извлекаться из других источников данных и записываться в Selena.

- В качестве исходной таблицы (неограниченное сканирование), таблицы измерений (ограниченное сканирование) или таблицы результатов (потоковая и пакетная запись) для разработки Flink SQL и datastream.

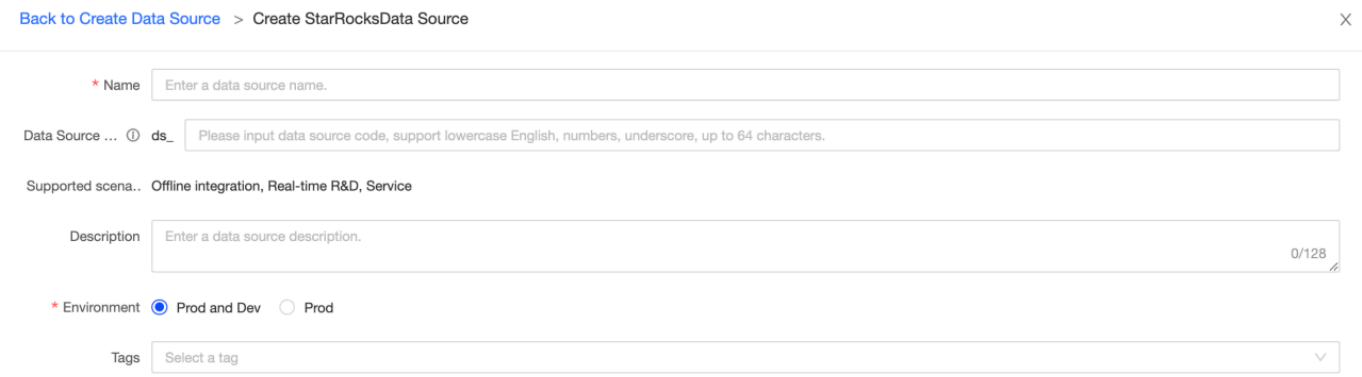
- В качестве хранилища данных или витрины данных. Selena может быть зарегистрирована как вычислительный источник, который может использоваться для разработки SQL-скриптов, планирования, обнаружения качества данных, идентификации безопасности и других задач исследования и управления данными.

Интеграция данных

Вы можете создавать источники данных Selena и использовать их в качестве исходных или целевых баз данных в задачах офлайн-интеграции. Процедура следующая:

Создание источника данных Selena

Основная информация



- **Name**: Обязательно. Введите имя источника данных. Может содержать только китайские символы, буквы, цифры, подчёркивания (_) и дефисы (-). Не может превышать 64 символа.

- **Data source code**: Необязательно. После настройки кода источника данных вы можете использовать формат `код источника данных.таблица` или `код источника данных.схема.таблица` для ссылки на Flink SQL в источнике данных. Если вы хотите автоматически получить доступ к источнику данных в соответствующей среде, используйте формат `${код источника данных}.таблица` или `${код источника данных}.схема.таблица`.

  > **ПРИМЕЧАНИЕ**
  >
  > В настоящее время поддерживаются только источники данных MySQL, Hologres и MaxCompute.

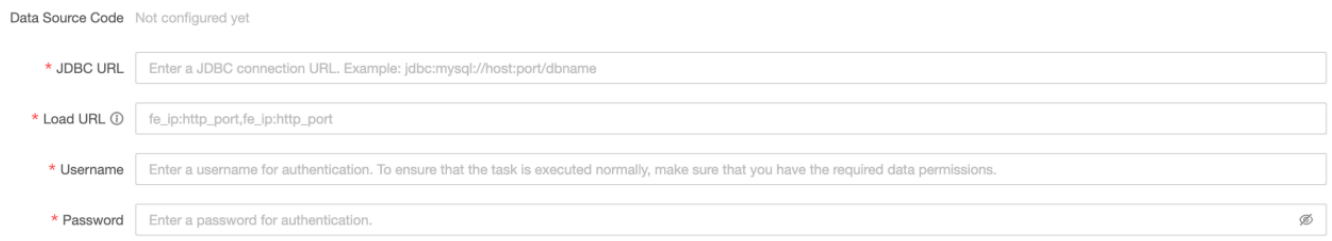
- **Support scenerios**: Сценарии, в которых может применяться источник данных.

- **Description**: Необязательно. Вы можете ввести краткое описание источника данных. Максимум 128 символов.

- **Environment**: Если бизнес-источник данных различает производственный и разрабатываемый источник данных, выберите **Prod and Dev**. Если бизнес-источник данных не различает производственный и разрабатываемый источники данных, выберите **Prod**.

- **Tags**: Вы можете выбрать теги для маркировки источников данных.

Информация о конфигурации



- **JDBC URL**: Обязательно. Формат: `jdbc:mysql://<host>:<port>/<dbname>`. `host` - это IP-адрес хоста FE (Front End) в cluster Selena, `port` - порт запросов FE, а `dbname` - имя базы данных.

- **Load URL**: Обязательно. Формат: `fe_ip:http_port;fe_ip:http_port`. `fe_ip` - это хост FE (Front End), а `http_port` - порт FE.

- **Username**: Обязательно. Имя пользователя базы данных.

- **Password**: Обязательно. Пароль базы данных.

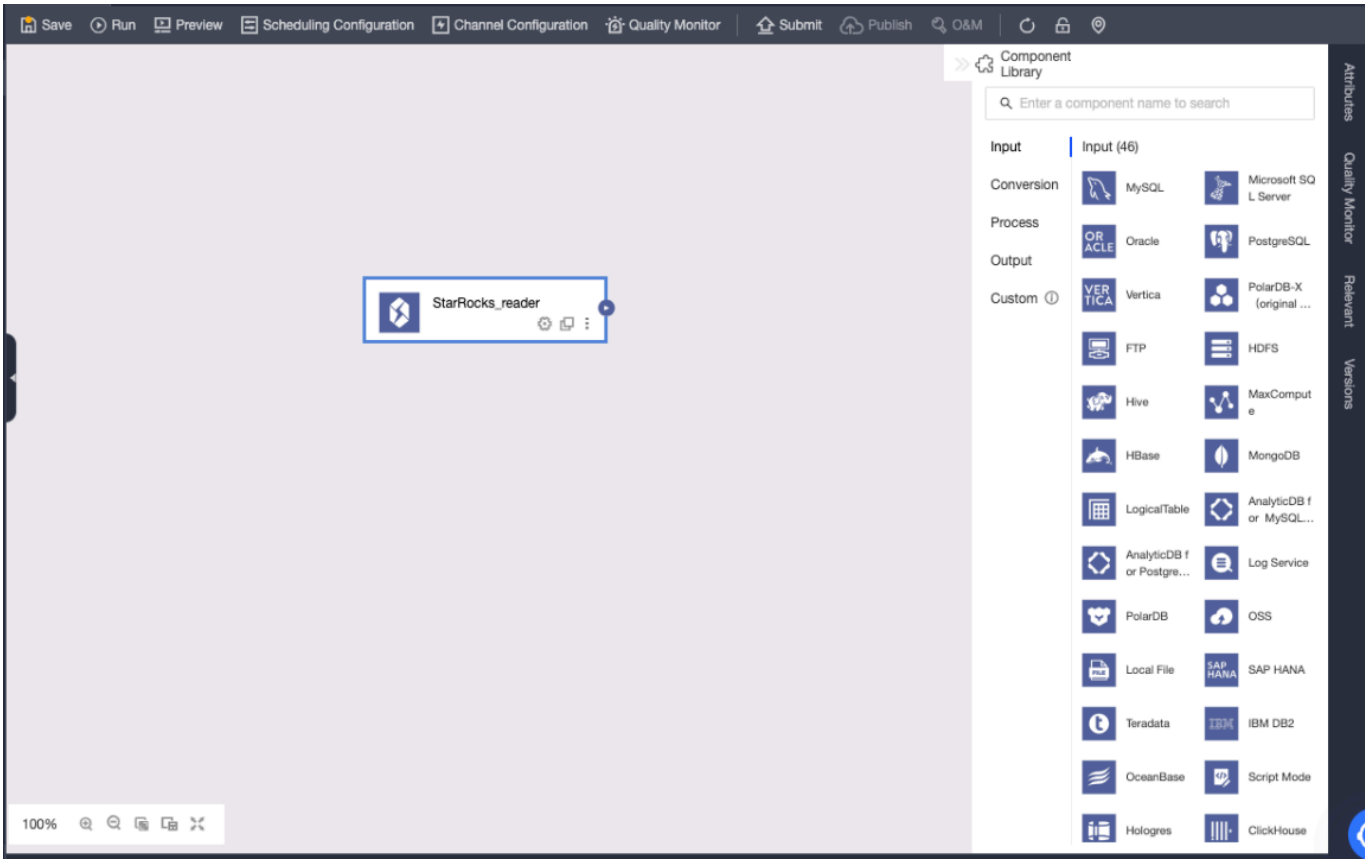
Расширенные настройки

- **connectTimeout**: таймаут подключения (в мс) базы данных. Значение по умолчанию: 900000 миллисекунд (15 минут).

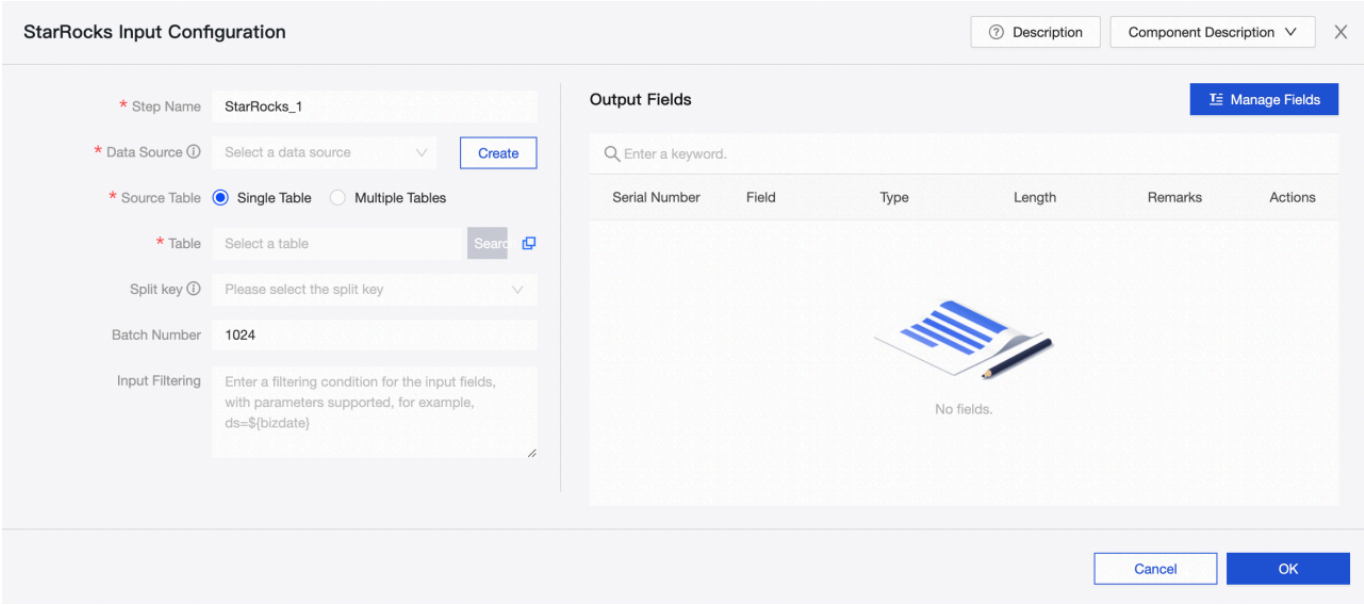
- **socketTimeout**: таймаут сокета (в мс) базы данных. Значение по умолчанию: 1800000 миллисекунд (30 минут).

Чтение данных из источников данных Selena и запись данных в другие источники данных

Перетащите компонент ввода Selena на холст задачи офлайн-интеграции



Конфигурация компонента ввода Selena



- **Step name**: Введите подходящее имя на основе сценария и расположения текущего компонента.

- **Data source**: Выберите источник данных Selena или проект, созданный в Dataphin. Требуется разрешение на чтение источника данных. Если подходящего источника данных нет, вы можете добавить источник данных или запросить соответствующие разрешения.

- **Source table**: Выберите одну таблицу или несколько таблиц с одинаковой структурой в качестве входных данных.

- **Table**: Выберите таблицу в источнике данных Selena из выпадающего списка.

- **Split key**: Используется с конфигурацией параллелизма. Вы можете использовать столбец в исходной таблице данных в качестве ключа разделения. Рекомендуется использовать первичный ключ или индексированный столбец в качестве ключа разделения.

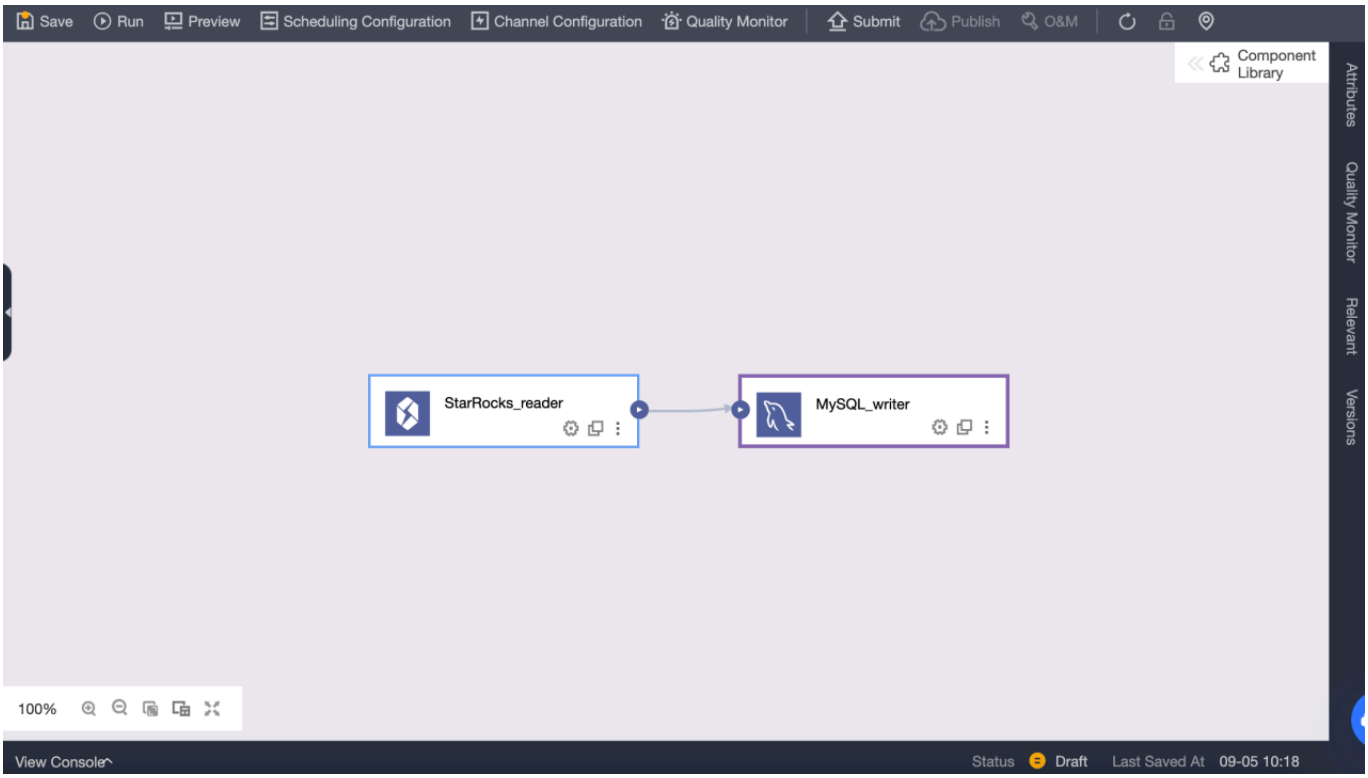
- **Batch number**: Количество записей данных, извлекаемых в пакете.

- **Input Filtering**: Необязательно.

  В следующих двух случаях необходимо заполнить информацию о фильтре:

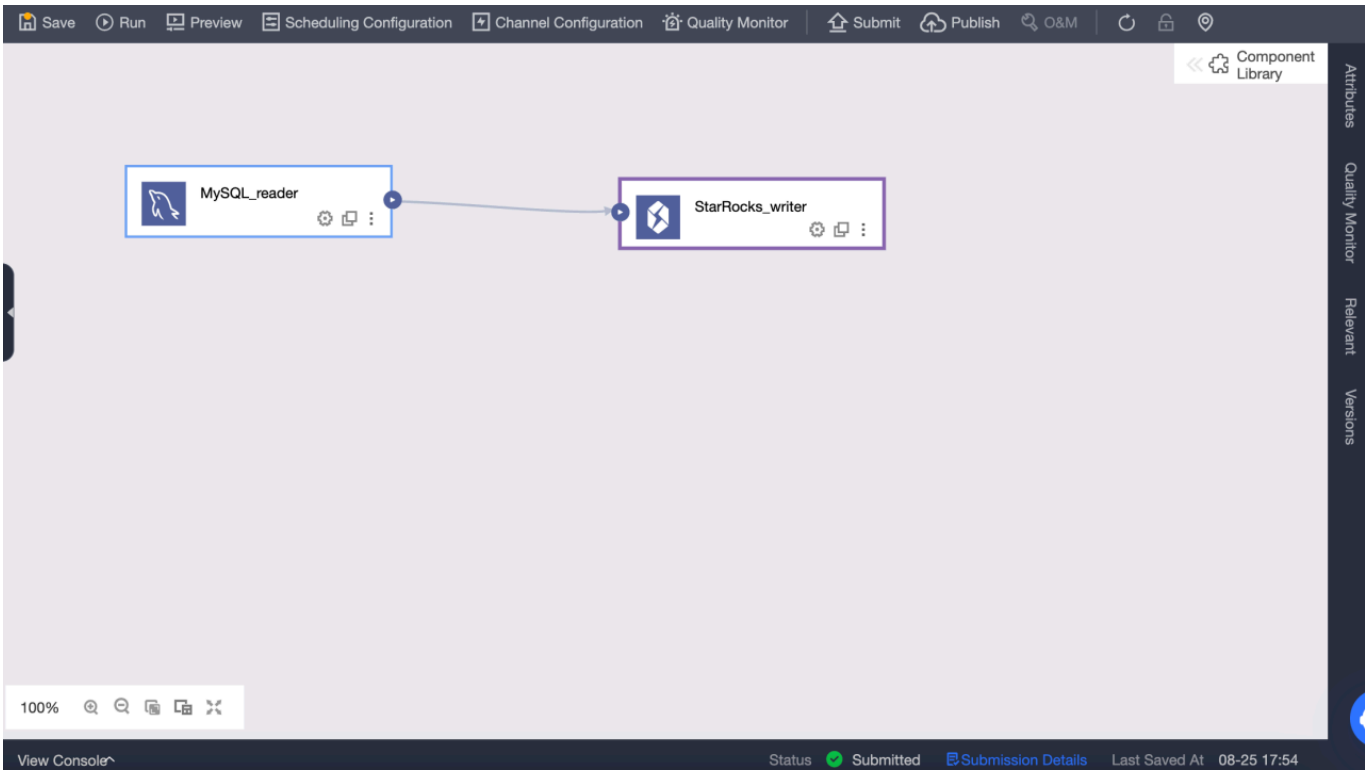
  - Если вы хотите отфильтровать определённую часть данных.
  - Если вам нужно инкрементально добавлять данные ежедневно или получать полные данные, необходимо заполнить дату, значение которой установлено как системное время консоли Dataphin. Например, таблица транзакций в Selena, и дата создания транзакции установлена как `${bizdate}`.
- **Output fields**: Список связанных полей на основе информации о входной таблице. Вы можете снова переименовывать, удалять, добавлять и перемещать поля. Как правило, поля переименовываются для повышения читаемости данных нижестоящего потока или для облегчения сопоставления полей при выводе. Поля могут быть удалены на этапе ввода, потому что соответствующие поля не нужны в сценариях применения. Порядок полей изменяется, чтобы гарантировать эффективное объединение данных или сопоставление выходных данных путём сопоставления полей с разными именами в одной строке при объединении или выводе нескольких входных данных на нижестоящей стороне.

Выберите и настройте компонент вывода в качестве целевого источника данных

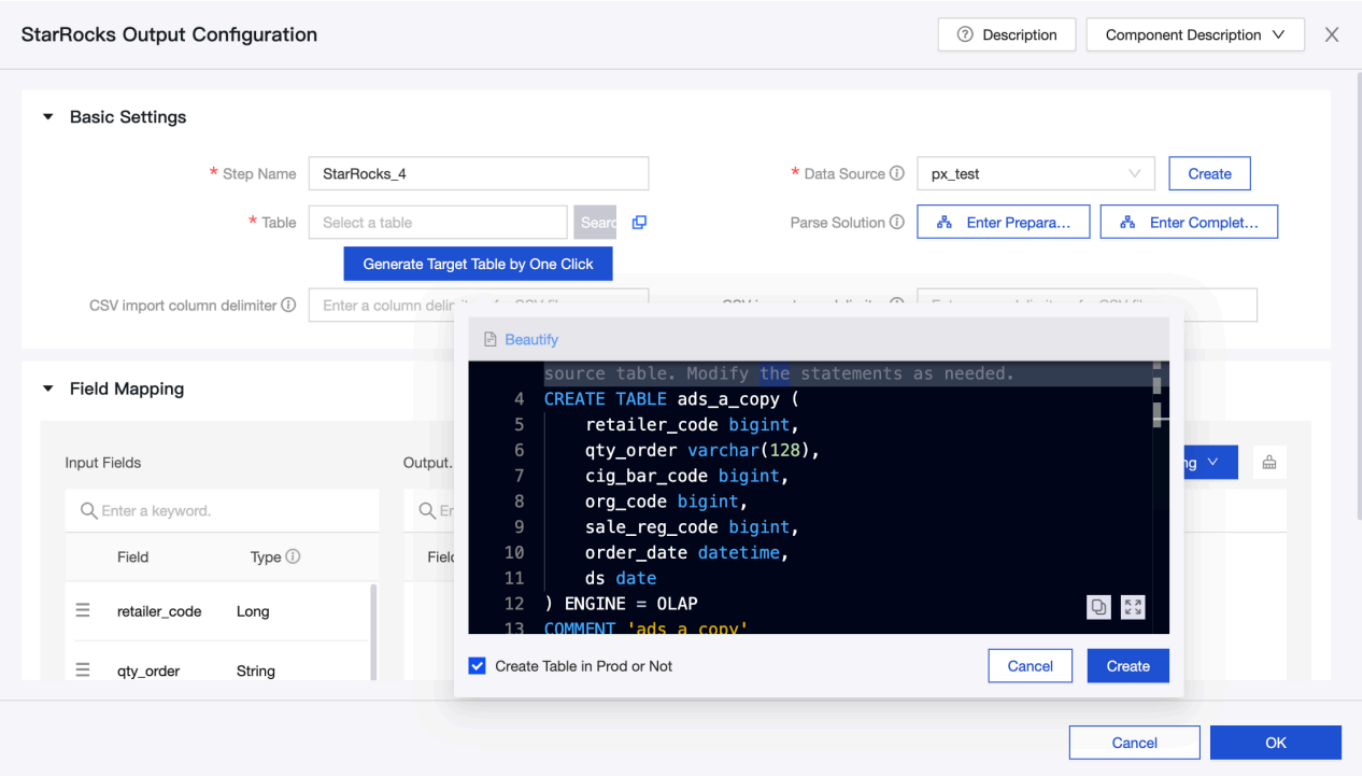


Чтение данных из других источников данных и запись данных в источники данных Selena

Настройте компонент ввода в задаче офлайн-интеграции и выберите и настройте компонент вывода Selena в качестве целевого источника данных



Конфигурация компонента вывода Selena



- **Step Name**: Введите подходящее имя на основе сценария и расположения текущего компонента.

- **Data Source**: Выберите источник данных или проект Dataphin, созданный в Selena. Источник данных, на который у конфигурирующего персонала есть разрешение на синхронную запись. Если источник данных не подходит, вы можете добавить источник данных или запросить соответствующие разрешения.

- **Table**: Выберите таблицу в источнике данных Selena из выпадающего списка.

- **Generate Target Table by One Click**: Если вы не создали целевую таблицу в источнике данных Selena, вы можете автоматически получить имя, тип и примечания полей, считанных из вышестоящего потока, и сгенерировать оператор создания таблицы. Нажмите для создания целевой таблицы одним щелчком.

- **CSV import column delimiter**: Используйте StreamLoad CSV для импорта. Вы можете настроить разделитель столбцов при импорте CSV. Значение по умолчанию `\t`. Не указывайте здесь значение по умолчанию. Если сами данные содержат `\t`, необходимо использовать другие символы в качестве разделителей.

- **CSV import row delimiter**: Используйте StreamLoad CSV для импорта. Вы можете настроить разделитель строк при импорте CSV. Значение по умолчанию: `\n`. Не указывайте здесь значение по умолчанию. Если сами данные содержат `\n`, необходимо использовать другие символы в качестве разделителей.

- **Parse Solution**: Необязательно. Это специальная обработка до или после записи данных. Подготовительный оператор выполняется перед записью данных в источник данных Selena, а оператор завершения выполняется после записи данных.

- **Field Mapping**: Вы можете вручную выбрать поля для сопоставления или использовать сопоставление на основе имени или позиции для обработки нескольких полей одновременно на основе полей из вышестоящего ввода и полей в целевой таблице.

Разработка в реальном времени

Краткое введение

Selena - это быстрая и масштабируемая база данных для анализа в реальном времени. Она обычно используется в вычислениях реального времени для чтения и записи данных для удовлетворения потребностей анализа и запросов данных в реальном времени. Она широко используется в корпоративных сценариях вычислений реального времени. Может использоваться в мониторинге и анализе бизнеса в реальном времени, анализе поведения пользователей в реальном времени, системах торгов рекламой в реальном времени, контроле рисков в реальном времени, защите от мошенничества, мониторинге и раннем предупреждении в реальном времени и других прикладных сценариях. Анализируя и запрашивая данные в реальном времени, предприятия могут быстро понимать условия бизнеса, оптимизировать решения, предоставлять лучшие услуги и защищать свои интересы.

Selena Connector

Коннектор Selena поддерживает следующую информацию:

Категория	Факты и цифры
Поддерживаемые типы	Исходная таблица, таблица измерений, таблица результатов
Режим работы	Потоковый режим и пакетный режим
Формат данных	JSON и CSV

Категория	Факты и цифры
Специальные метрики	Нет
Тип API	Datastream и SQL
Поддержка обновления или удаления данных в таблице результатов?	Да

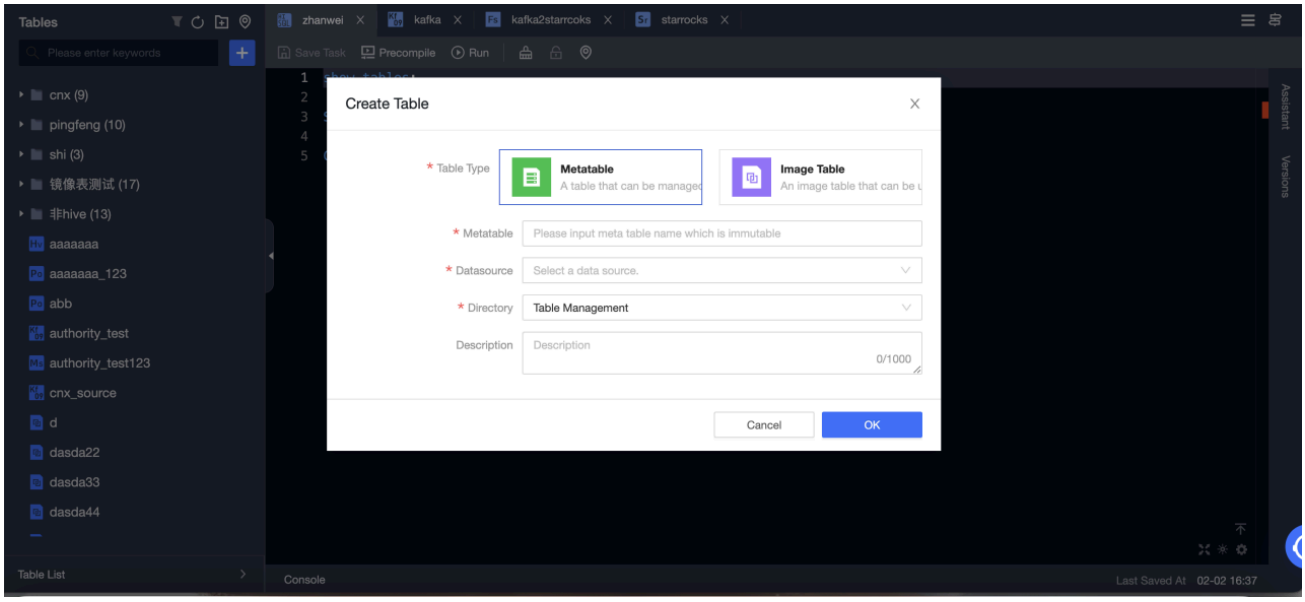
Как использовать?

Dataphin поддерживает источники данных Selena в качестве целей чтения и записи для вычислений реального времени. Вы можете создавать мета-таблицы Selena и использовать их для задач вычислений реального времени:

Создание мета-таблицы Selena

1. Перейдите в **Dataphin** > **R & D** > **Develop** > **Tables**.

2. Нажмите **Create** для выбора таблицы вычислений реального времени.



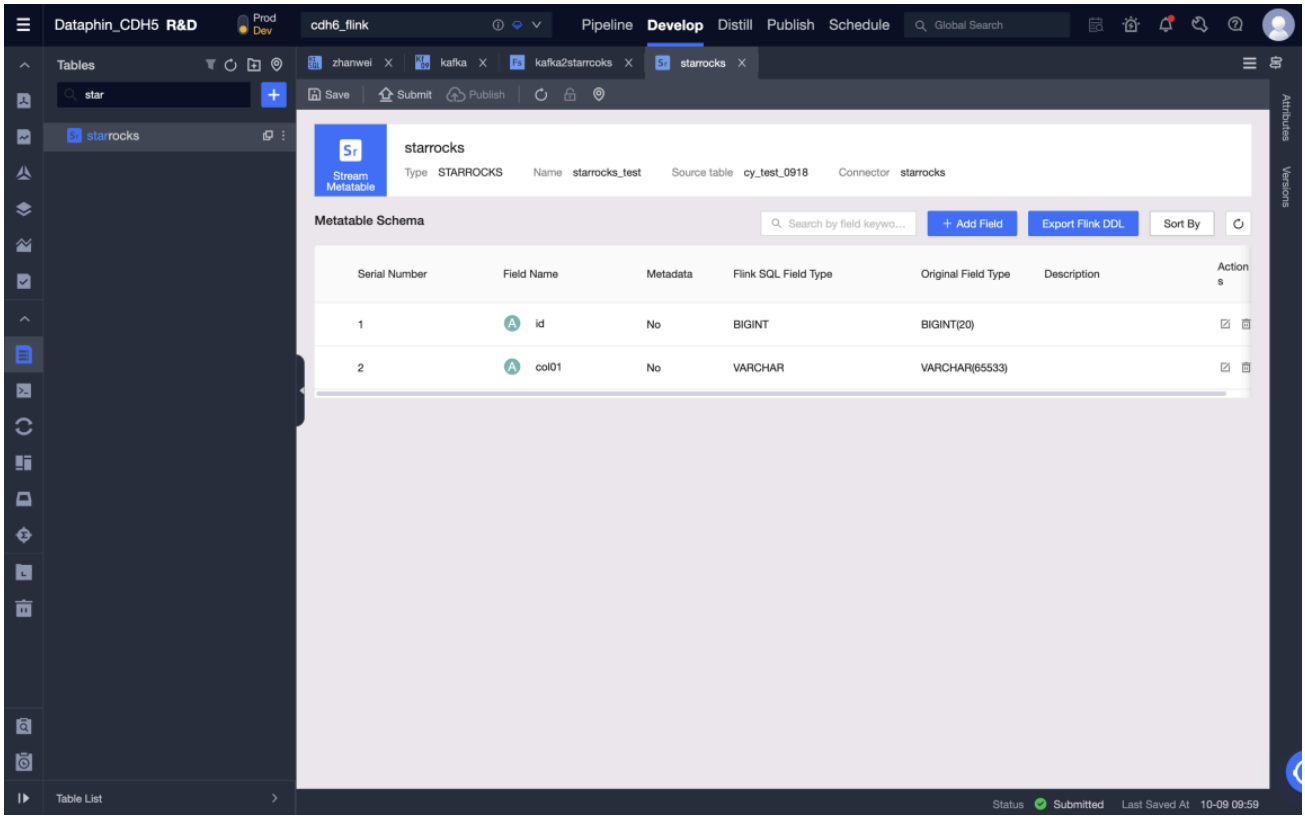
   - **Table type**: Выберите **Metatable**.

   - **Metatable**: Введите имя мета-таблицы. Имя неизменяемо.

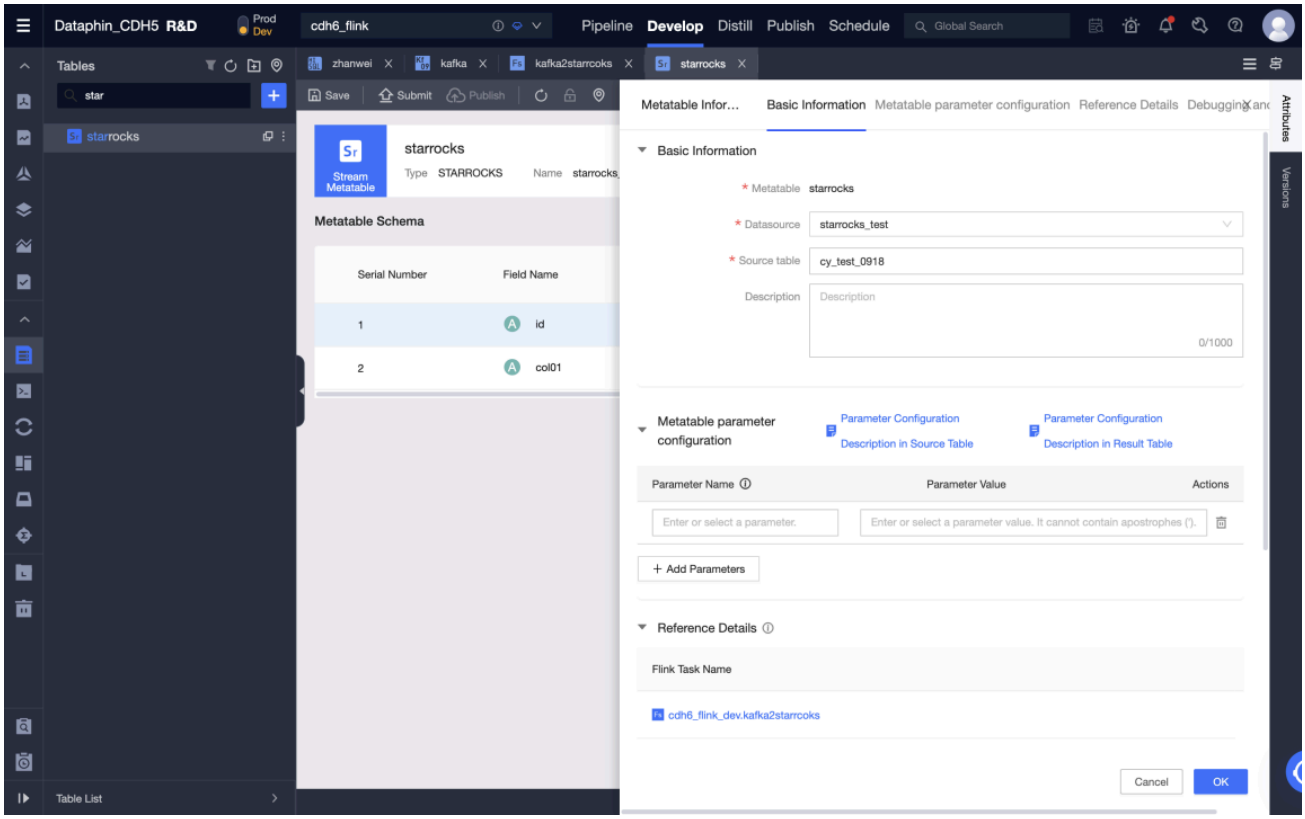
   - **Datasource**: Выберите источник данных Selena.

   - **Directory**: Выберите каталог, в котором вы хотите создать таблицу.

   - **Description**: Необязательно.



3. После создания мета-таблицы вы можете редактировать мета-таблицу, включая изменение источников данных, исходных таблиц, полей мета-таблицы и настройку параметров мета-таблицы.

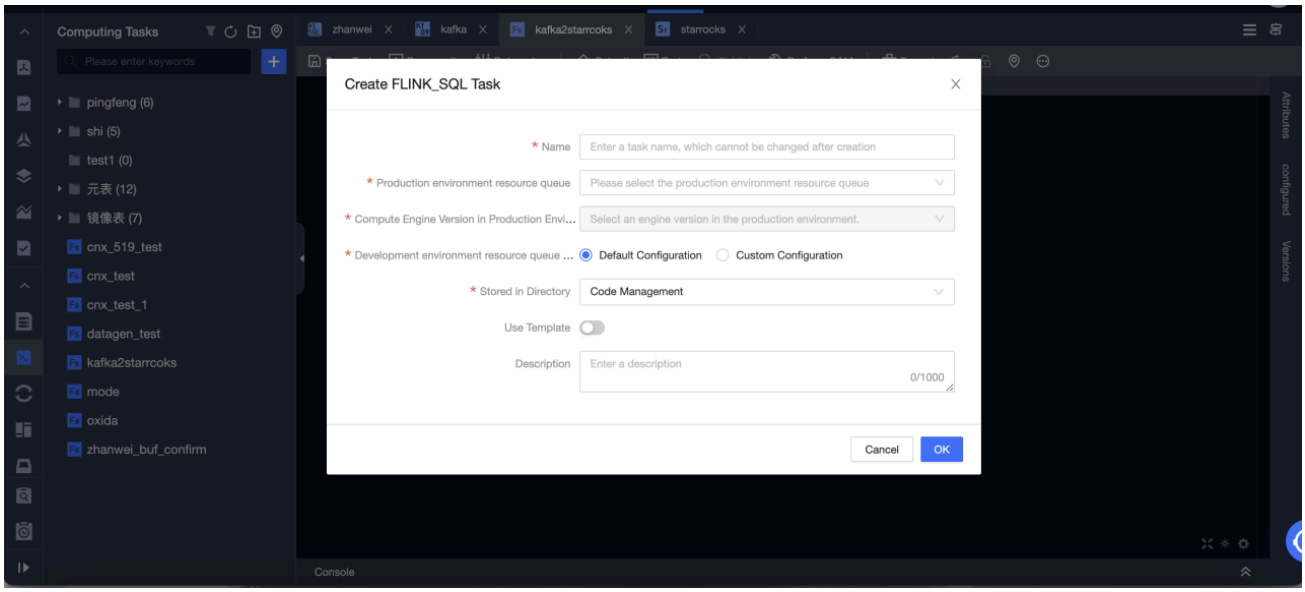


4. Отправьте мета-таблицу.

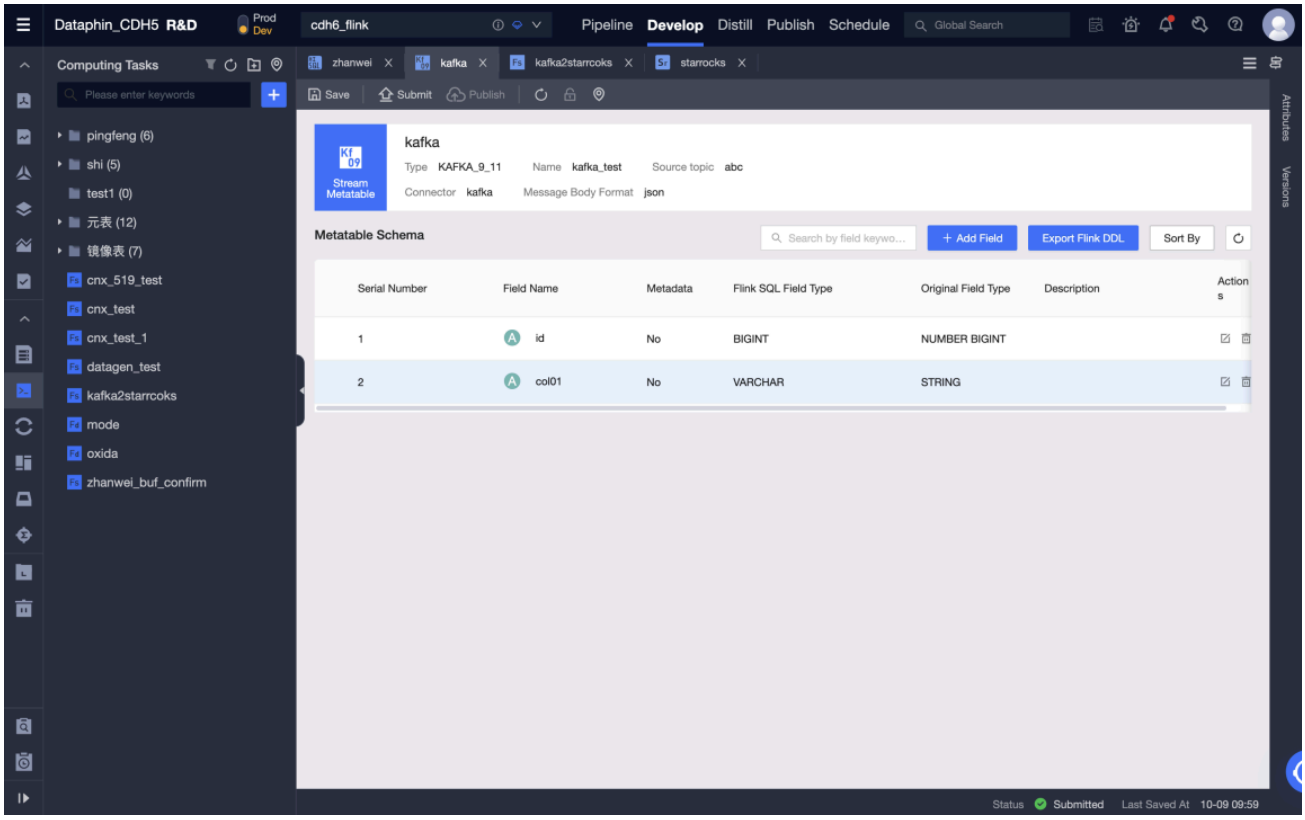
Создание задачи Flink SQL для записи данных из Kafka в Selena в реальном времени

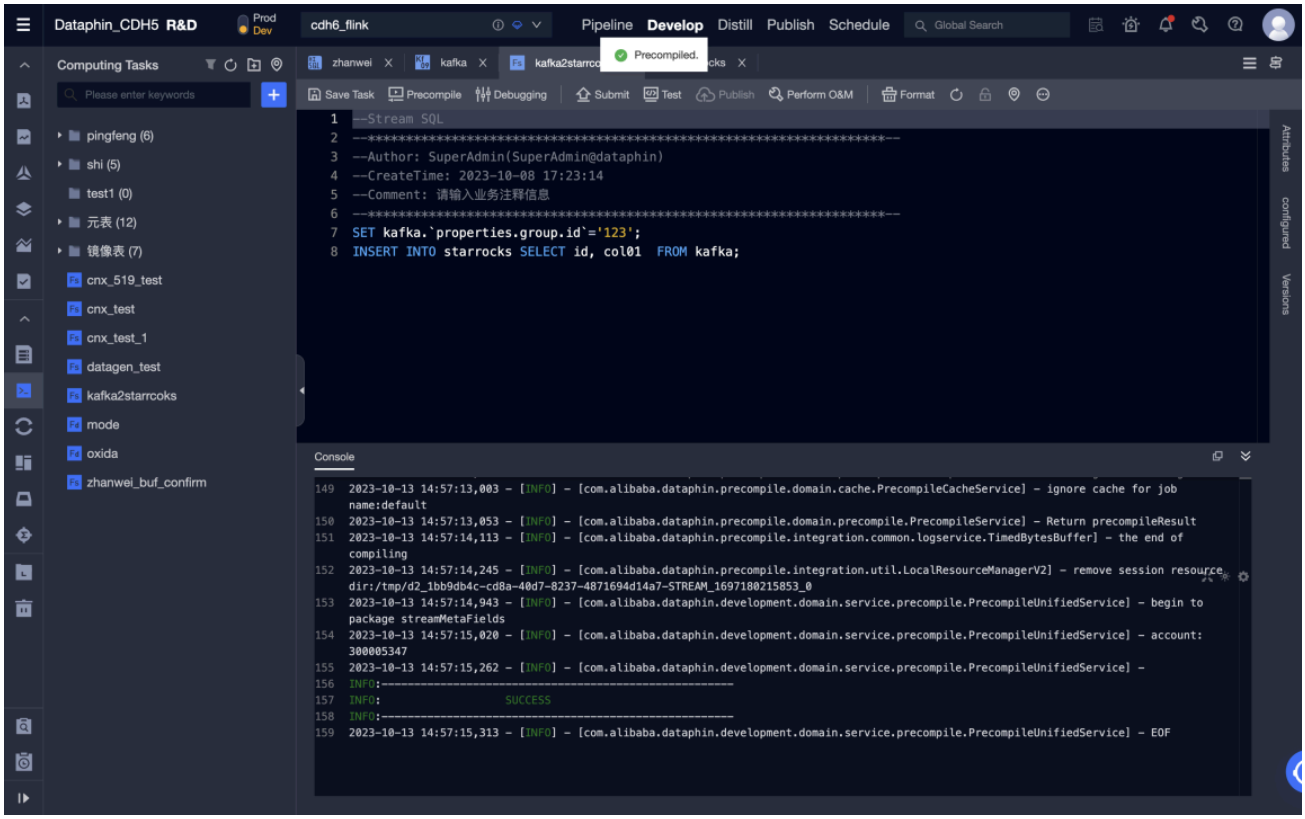
1. Перейдите в **Dataphin** > **R & D** > **Develop** > **Computing Tasks**.

2. Нажмите **Create Flink SQL task**.



3. Отредактируйте код Flink SQL и предварительно скомпилируйте его. Мета-таблица Kafka используется как входная таблица, а мета-таблица Selena - как выходная таблица.

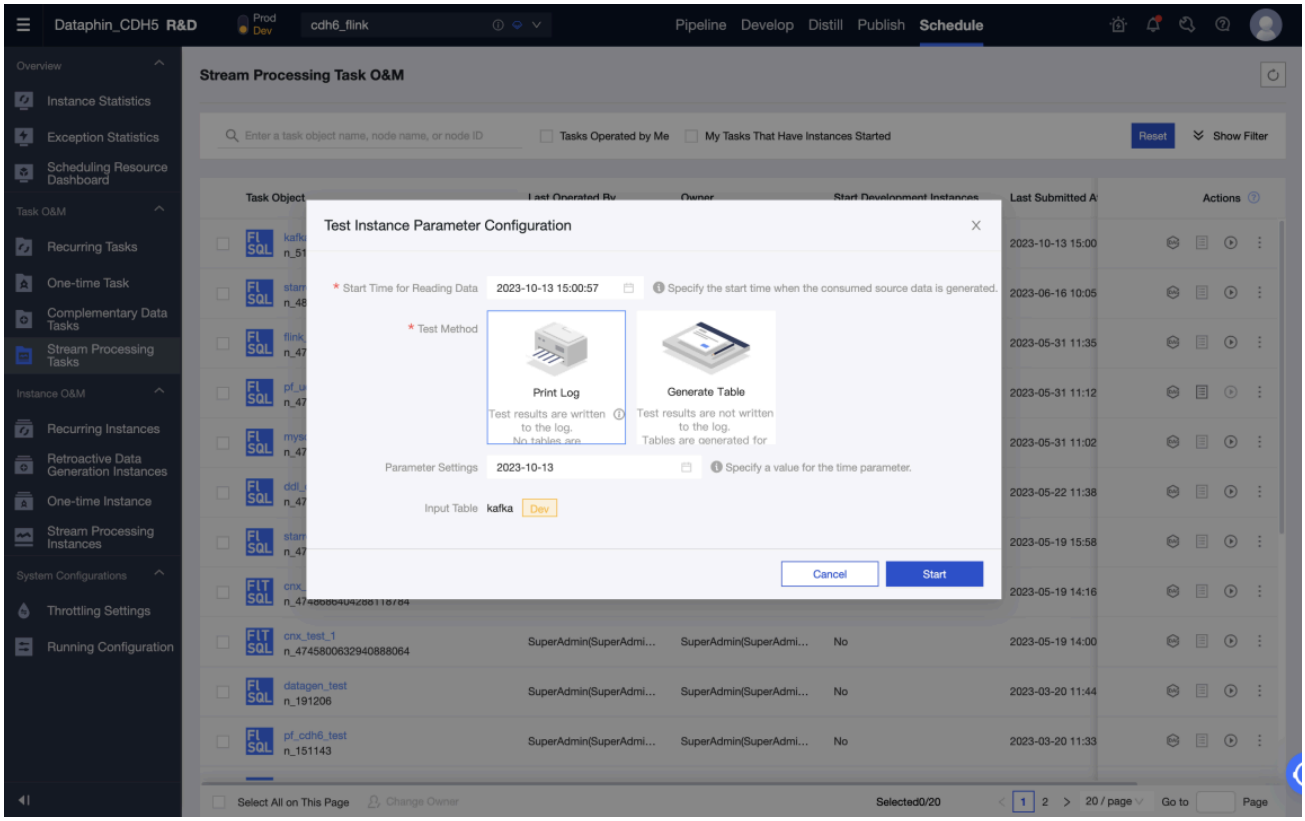


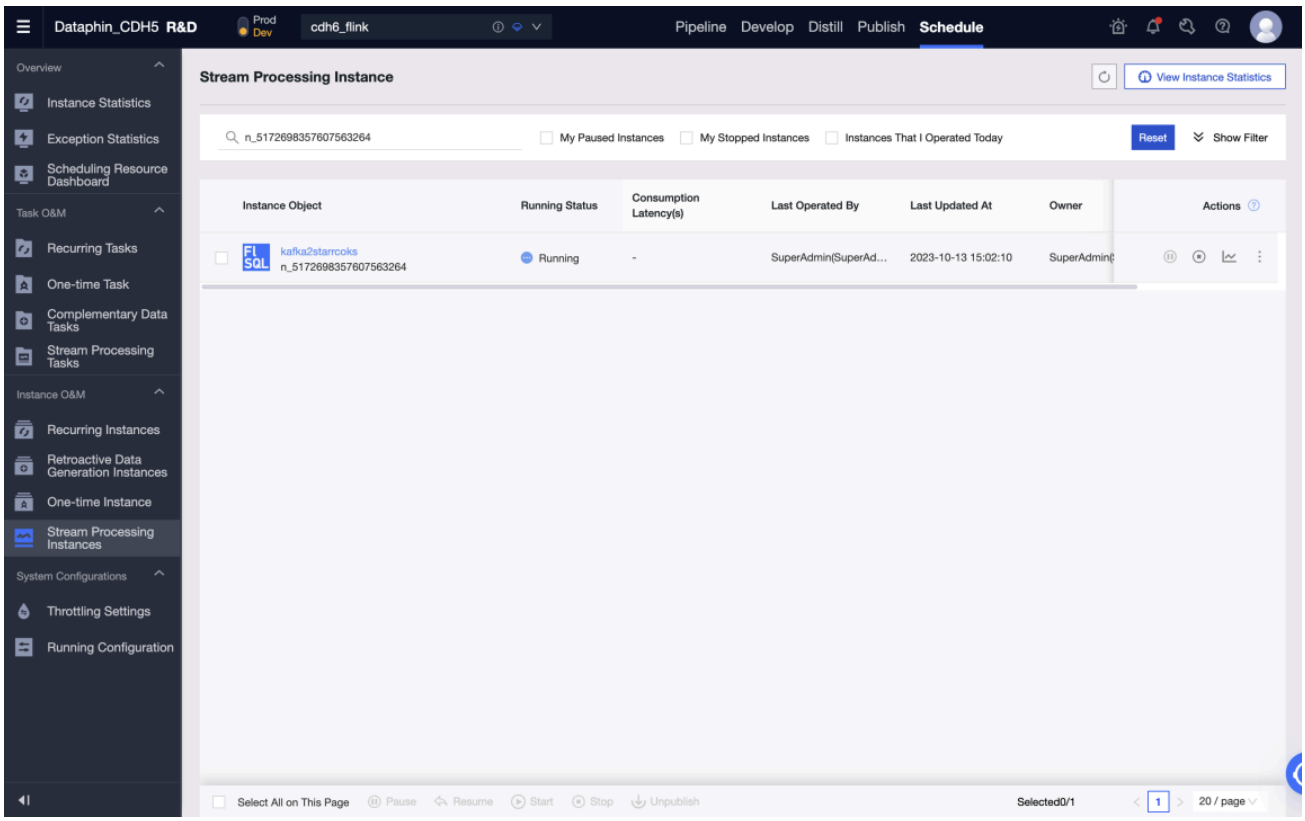


4. После успешной предварительной компиляции вы можете отладить и отправить код.

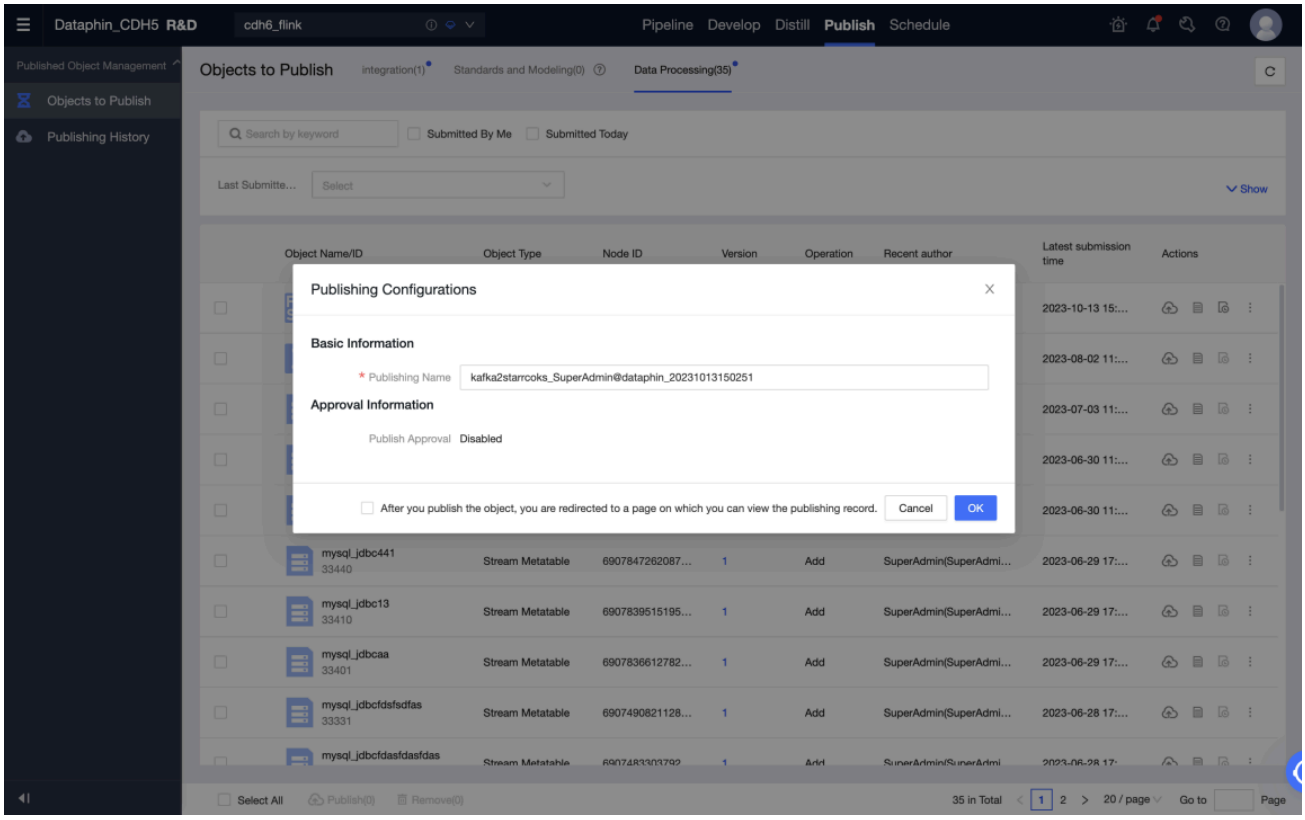
5. Тестирование в среде разработки может выполняться путём печати логов и записи тестовых таблиц. Тестовые таблицы можно установить в Meta Tables > Properties >

debugging test configurations.

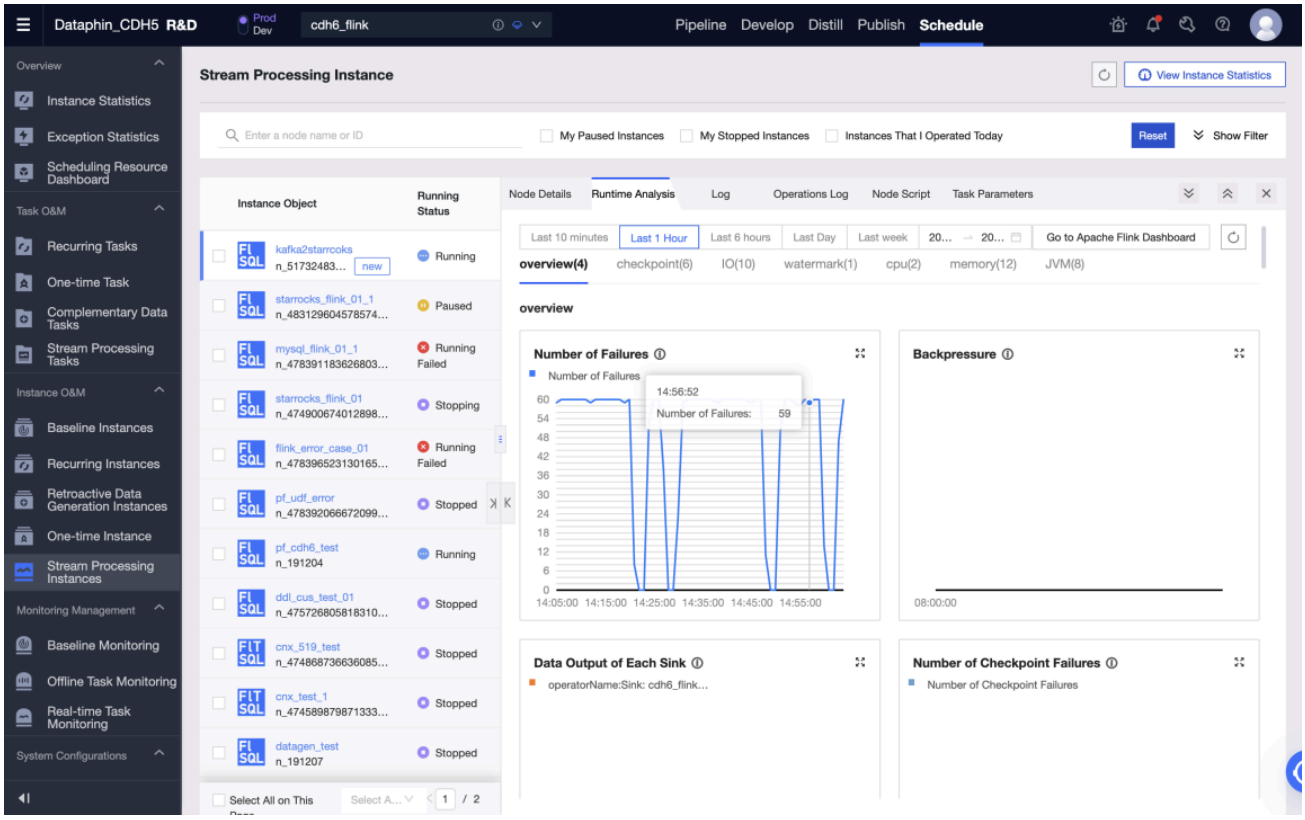


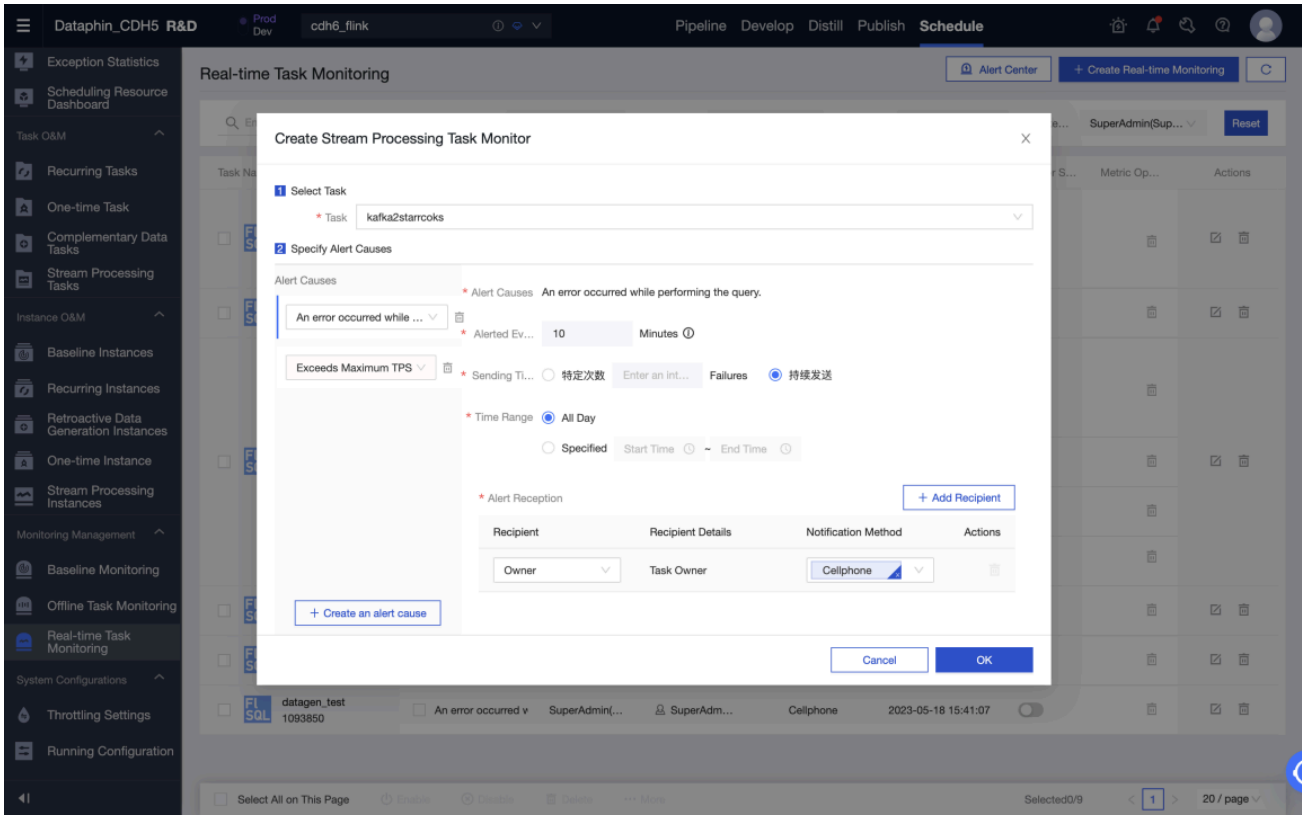


6. После нормальной работы задачи в среде разработки вы можете опубликовать задачу и используемую мета-таблицу в производственную среду.



7. Запустите задачу в производственной среде для записи данных из Kafka в Selena в реальном времени. Вы можете просматривать статус и логи каждой метрики в анализе выполнения, чтобы узнать о статусе выполнения задачи, или настроить оповещения мониторинга для задачи.





Хранилище данных или витрина данных

Предварительные требования

- Версия Selena 3.0.6 или выше.

- Dataphin установлен, и версия Dataphin 3.12 или выше.
- Сбор статистики должен быть включён. После установки Selena сбор включён по умолчанию. Подробнее см. Сбор статистики для СВО.
- Поддерживается внутренний каталог Selena (каталог по умолчанию), внешний каталог не поддерживается.

Конфигурация подключения

Настройки хранилища метаданных

Dataphin может представлять и отображать информацию на основе метаданных, включая информацию об использовании таблиц и изменения метаданных. Вы можете использовать Selena для обработки и расчёта метаданных. Поэтому перед использованием необходимо инициализировать вычислительный движок метаданных (хранилище метаданных). Процедура следующая:

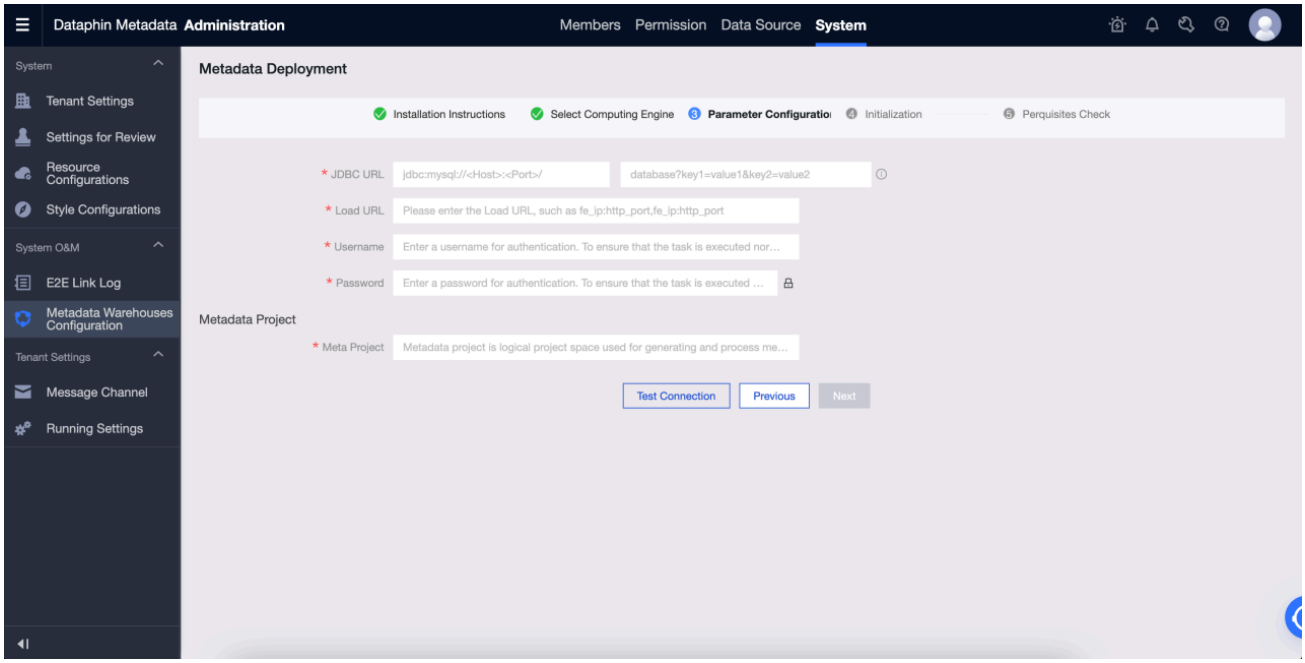
1. Используйте учётную запись администратора для входа в арендатора хранилища метаданных Dataphin
2. Перейдите в Administration > System > Metadata Warehouse Configuration

   a. Нажмите Start

   b. Выберите Selena

   c. Настройте параметры. После прохождения тестового подключения нажмите далее.

   d. Завершите инициализацию хранилища метаданных



Параметры описаны следующим образом:

- **JDBC URL**: Строка подключения JDBC, которая делится на две части:
  - Часть I: Формат `jdbc:mysql://<Host>:<Port>/`. `Host` - это IP-адрес хоста FE в cluster Selena. `Port` - порт запросов FE. Значение по умолчанию: `9030`.
  - Часть II: формат `database? key1 = value1 & key2 = value2`, где `database` - это имя базы данных Selena, используемой для расчёта метаданных, которое обязательно. Параметр после '?' необязателен.
- **Load URL**: Формат `fe_ip:http_port;fe_ip:http_port`. `fe_ip` - это хост FE (Front End), а `http_port` - порт FE.
- **Username**: Имя пользователя для подключения к Selena.

  Пользователь должен иметь разрешения на чтение и запись в базу данных, указанную в JDBC URL, и должен иметь разрешения на доступ к следующим базам данных и таблицам:

  - Все таблицы в Information Schema
  - *statistics*.column_statistics
  - *statistics*.table_statistic_v1

- **Password**: Пароль для подключения к Selena.
- **Meta Project**: Имя проекта, используемого для обработки метаданных в Dataphin. Используется только внутри системы Dataphin. Рекомендуется использовать `dataphin_meta` в качестве имени проекта.

Создание проекта Selena и начало разработки данных

Чтобы начать разработку данных, выполните следующие шаги:

1. Настройки вычислений.
2. Создайте вычислительный источник Selena.
3. Создайте проект.
4. Создайте задачу Selena SQL.

Настройки вычислений

Настройки вычислений устанавливают тип вычислительного движка и адрес cluster арендатора. Подробные шаги следующие:

1. Войдите в Dataphin как системный администратор или суперадминистратор.
2. Перейдите в **Administration** > **System** > **Computation Configuration**.
3. Выберите **Selena** и нажмите **Next**.
4. Введите JDBC URL и проверьте его. Формат JDBC URL: `jdbc:mysql://<Host>:<Port>/`. `Host` - это IP-адрес хоста FE в cluster Selena. `Port` - порт запросов FE. Значение по умолчанию: `9030`.

Вычислительный источник Selena

Вычислительный источник - это концепция Dataphin. Его основная цель - связать и зарегистрировать пространство проекта Dataphin с пространством хранения и вычислений Selena (базой данных). Вы должны создать вычислительный источник для каждого проекта. Подробные шаги следующие:

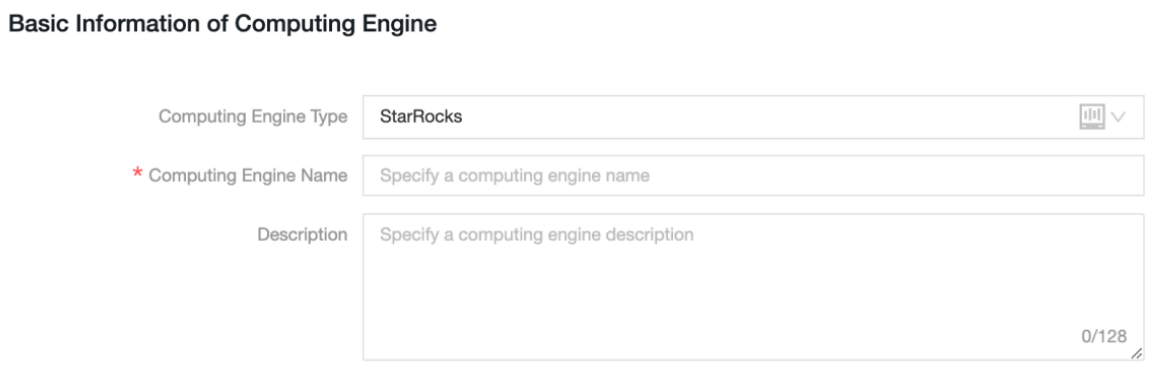
1. Войдите в Dataphin как системный администратор или суперадминистратор.

2. Перейдите в **Planning** > **Engine**.

3. Нажмите **Add Computing Engine** в правом верхнем углу для создания вычислительного источника.

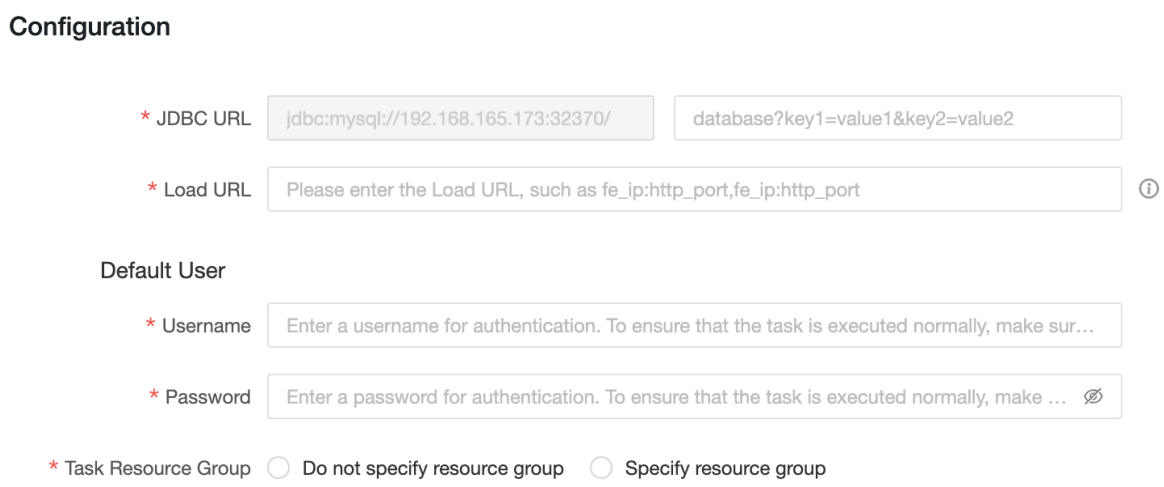
Подробная информация о конфигурации следующая:

1. **Основная информация**

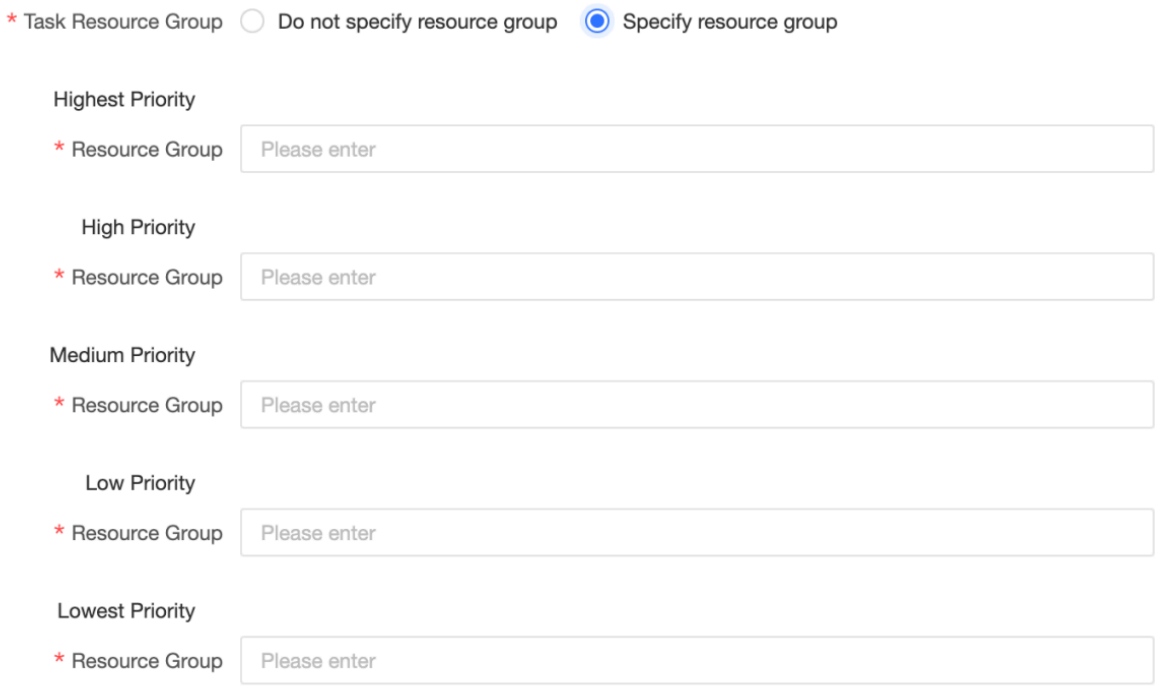


   - **Computing Engine Type**: Выберите **Selena**.
   - **Computing Engine Name**: Рекомендуется использовать то же имя, что и у создаваемого проекта. Для проектов разработки добавьте суффикс `_dev`.
   - **Description**: Необязательно. Введите описание вычислительного источника.

2. **Информация о конфигурации**



- **JDBC URL**: Формат `jdbc:mysql://<Host>:<Port>/`. `Host` - это IP-адрес хоста FE в cluster Selena. `Port` - порт запросов FE. Значение по умолчанию: `9030`.
- **Load URL**: Формат `fe_ip:http_port;fe_ip:http_port`. `fe_ip` - это хост FE (Front End), а `http_port` - порт FE.
- **Username**: Имя пользователя для подключения к Selena.
- **Password**: Пароль Selena.
- **Task Resource Group**: вы можете указать разные группы ресурсов Selena для задач с разными приоритетами. Когда вы выбираете не указывать группу ресурсов, движок Selena определяет группу ресурсов для выполнения. Когда вы выбираете указать группу ресурсов, задачи с разными приоритетами назначаются указанной группе ресурсов Dataphin. Если группа ресурсов указана в коде SQL-задачи или в материализованной конфигурации логической таблицы, конфигурация группы ресурсов задачи вычислительного источника игнорируется при выполнении задачи.



Проект Dataphin

После создания вычислительного источника вы можете привязать его к проекту Dataphin. Проект Dataphin управляет участниками проекта, хранилищем и вычислительным

пространством Selena, а также управляет и поддерживает вычислительные задачи.

Чтобы создать проект Dataphin, выполните следующие шаги:

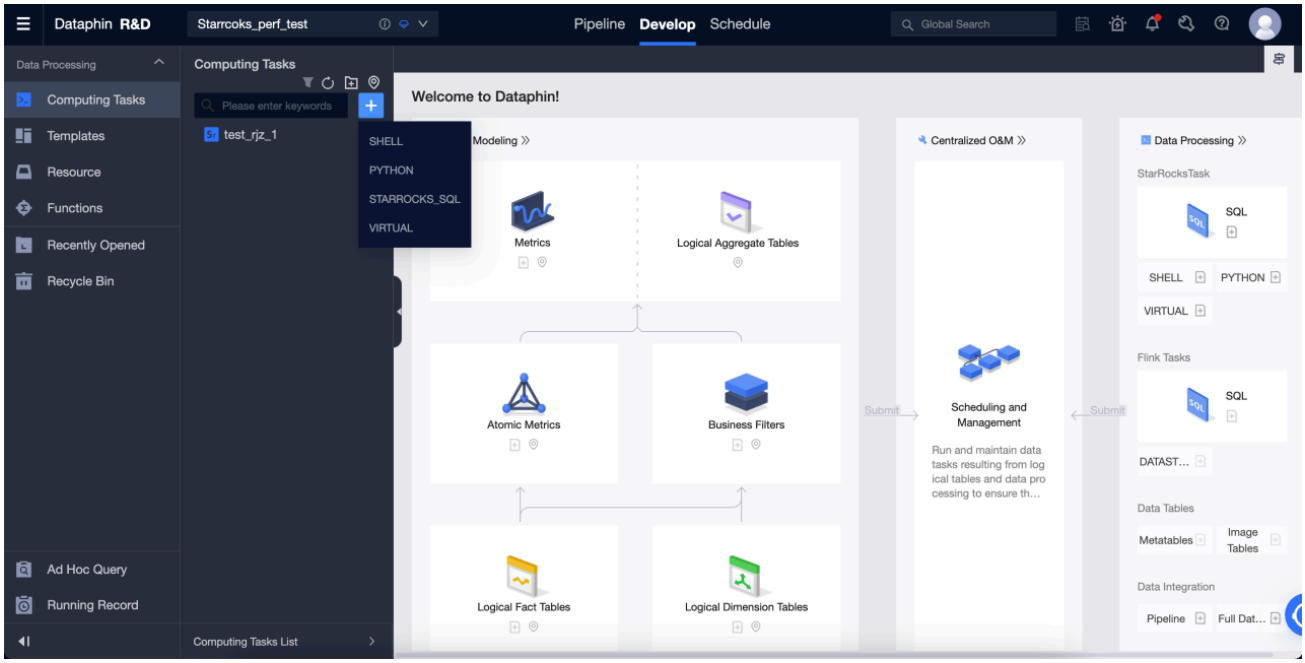
1. Войдите в Dataphin как системный администратор или суперадминистратор.
2. Перейдите в **Planning** > **Project Management**.
3. Нажмите **Create project** в правом верхнем углу для создания проекта.
4. Введите основную информацию и выберите движок Selena, созданный на предыдущем шаге, из офлайн-движка.
5. Нажмите **Create**.

Selena SQL

После создания проекта вы можете создать задачу Selena SQL для выполнения операций DDL или DML в Selena.

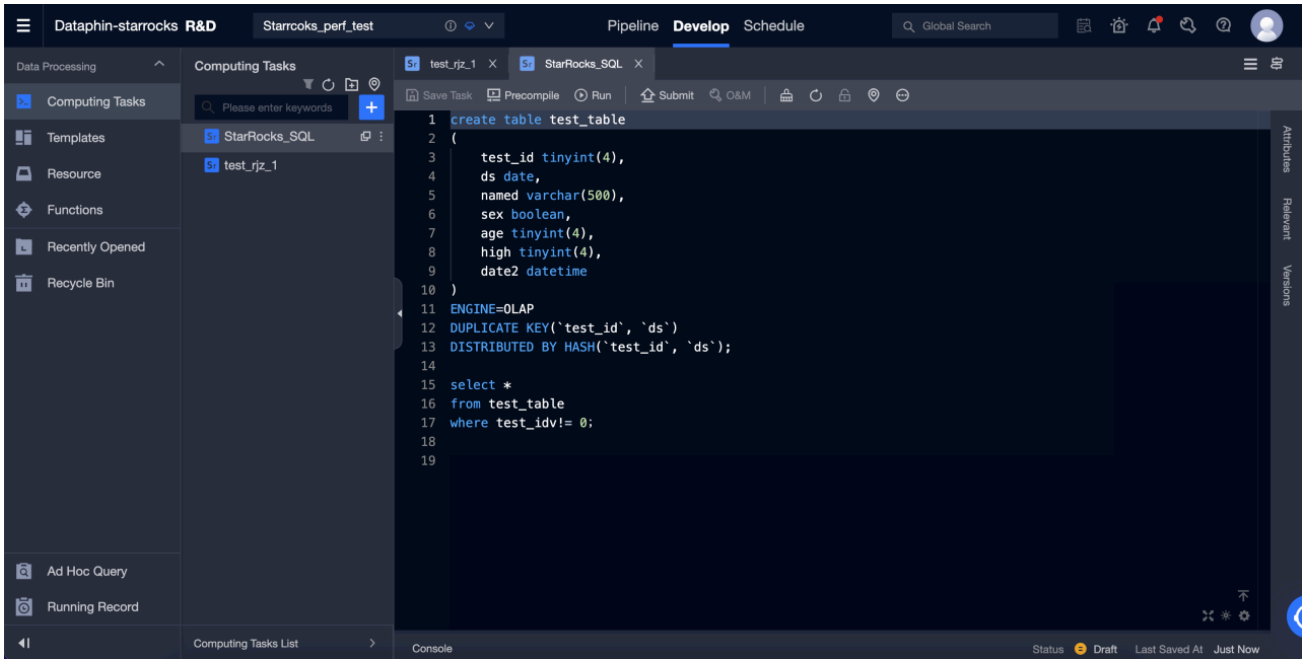
Подробные шаги следующие:

1. Перейдите в **R & D** > **Develop**.
2. Нажмите '+' в правом верхнем углу для создания задачи Selena SQL.



3. Введите имя и тип планирования для создания SQL-задачи.

4. Введите SQL в редакторе для начала операций DDL и DML в Selena.



1.8.3 DBeaver

DBeaver - это программное приложение SQL-клиента и инструмент администрирования баз данных, который предлагает полезного помощника, проводящего вас через процесс подключения к базе данных.

Предварительные требования

Убедитесь, что вы установили DBeaver.

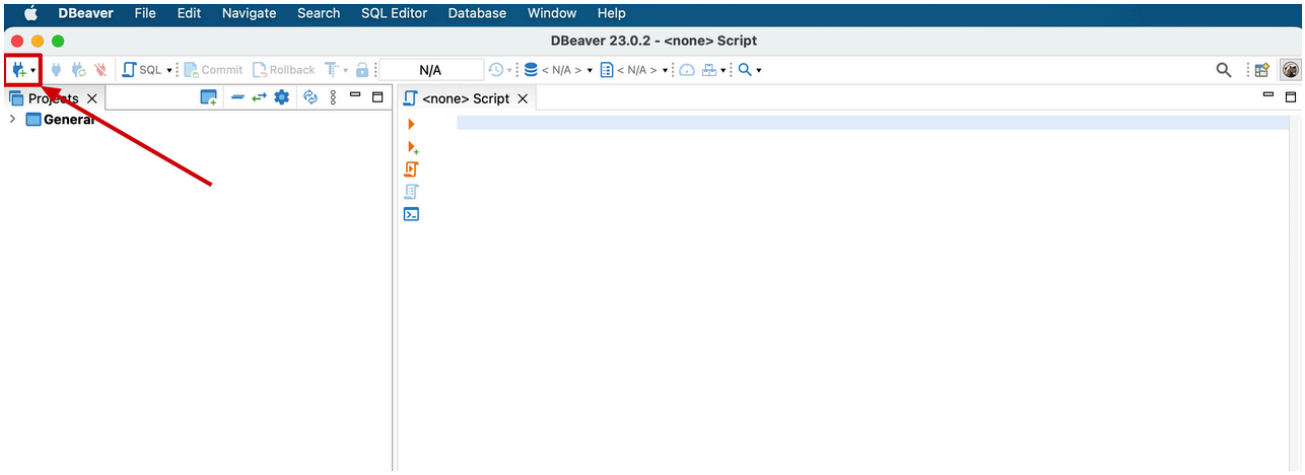
Вы можете загрузить DBeaver Community edition по адресу https://dbeaver.io или DBeaver PRO edition по адресу https://dbeaver.com.

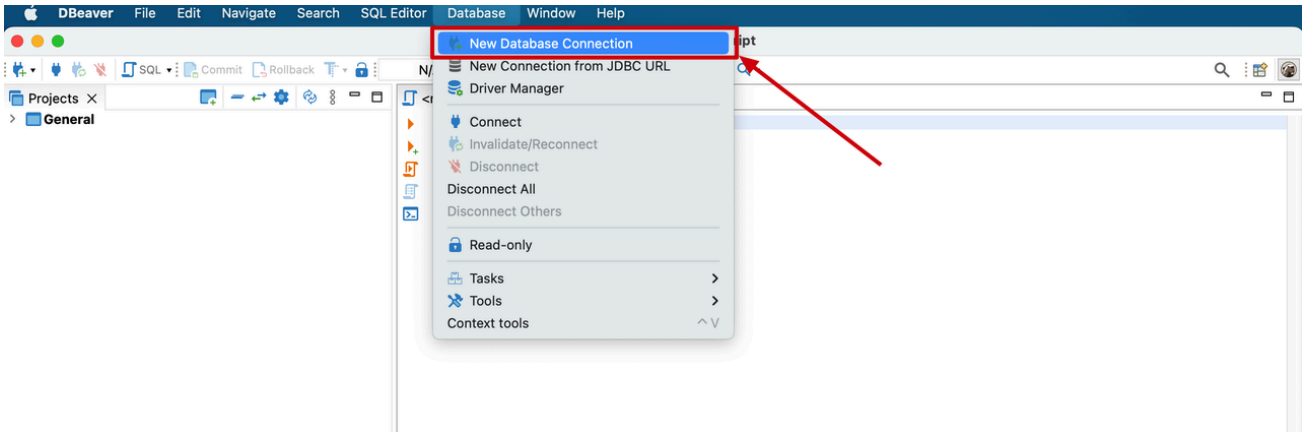
Интеграция

Выполните следующие шаги для подключения к базе данных:

1. Запустите DBeaver.

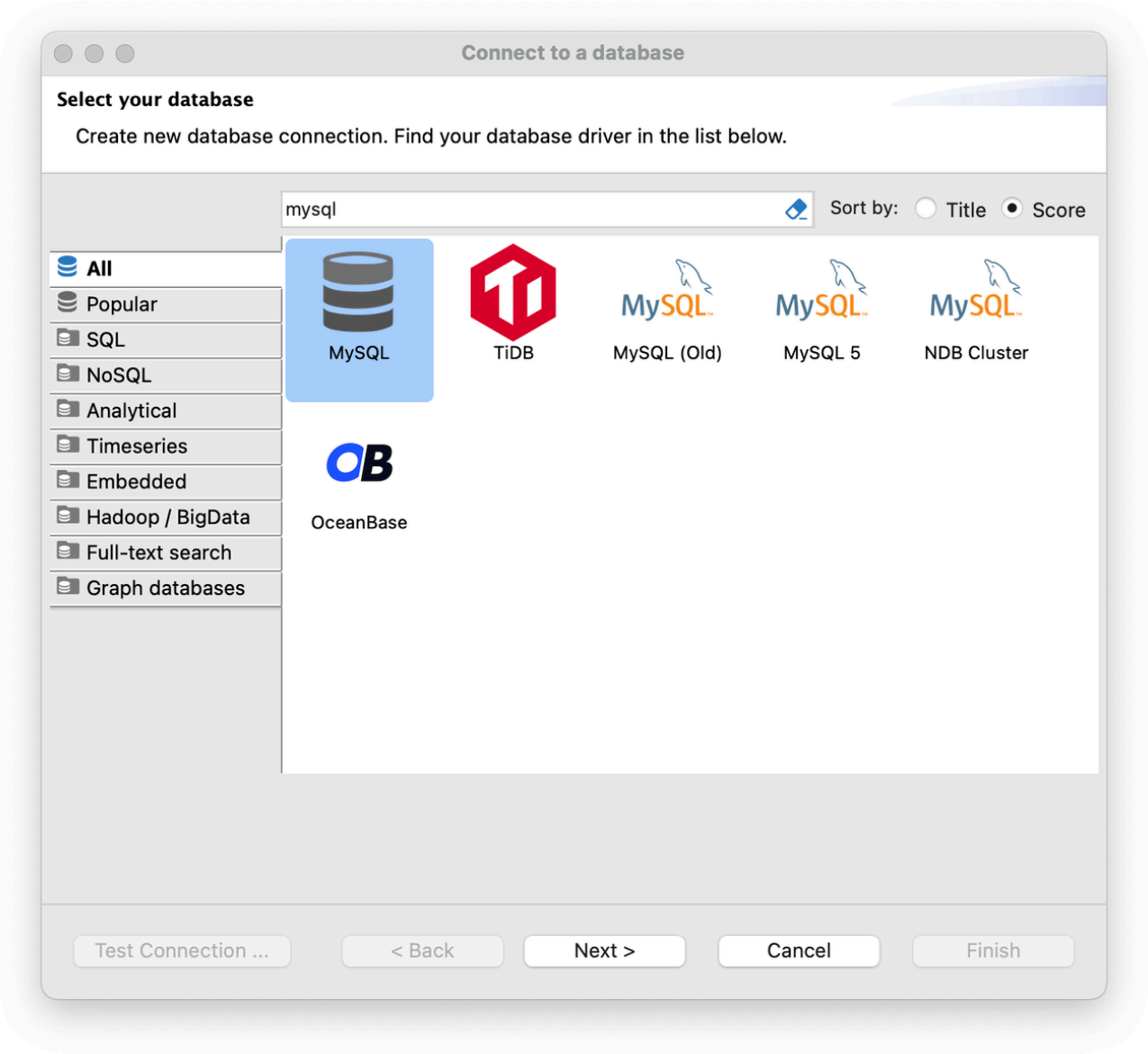
2. Нажмите значок плюса (**+**) в верхнем левом углу окна DBeaver или выберите **Database** > **New Database Connection** в строке меню для доступа к помощнику.





3. Выберите драйвер MySQL.

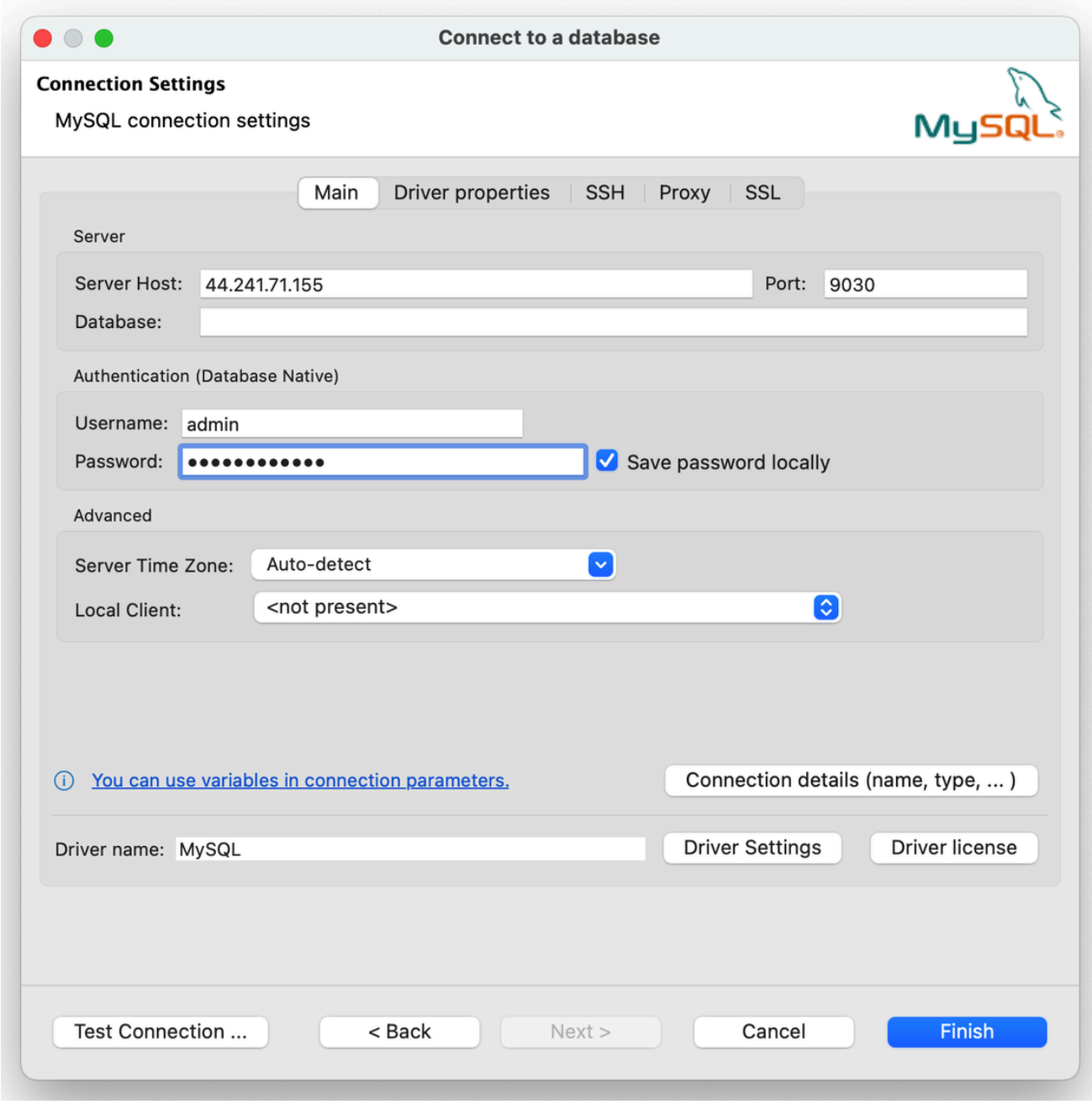
   На шаге **Select your database** вам представлен список доступных драйверов. Нажмите **Analytical** на левой панели для быстрого поиска драйвера MySQL. Затем дважды щёлкните по значку **MySQL**.



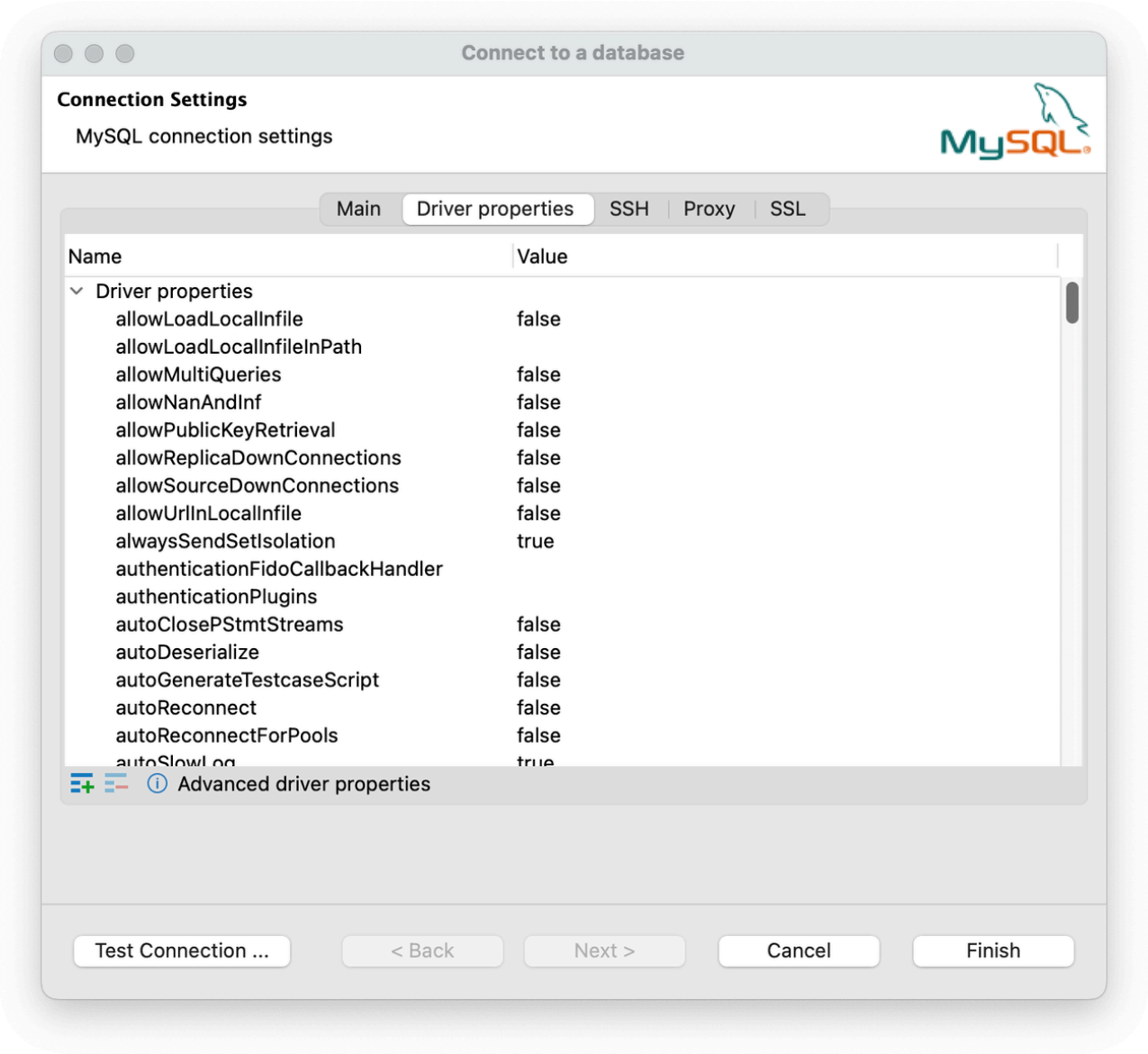
4. Настройте подключение к базе данных.

   На шаге **Connection Settings** перейдите на вкладку **Main** и настройте следующие основные параметры подключения:

   - **Server Host**: IP-адрес хоста FE вашего cluster Selena.
   - **Port**: порт запросов FE вашего cluster Selena, например, `9030`.
   - **Database**: целевая база данных в вашем cluster Selena. Поддерживаются как внутренние, так и внешние базы данных, но функциональность для внешних баз данных может быть неполной.
   - **Username**: имя пользователя для входа в ваш cluster Selena, например, `admin`.
   - **Password**: пароль для входа в ваш cluster Selena.

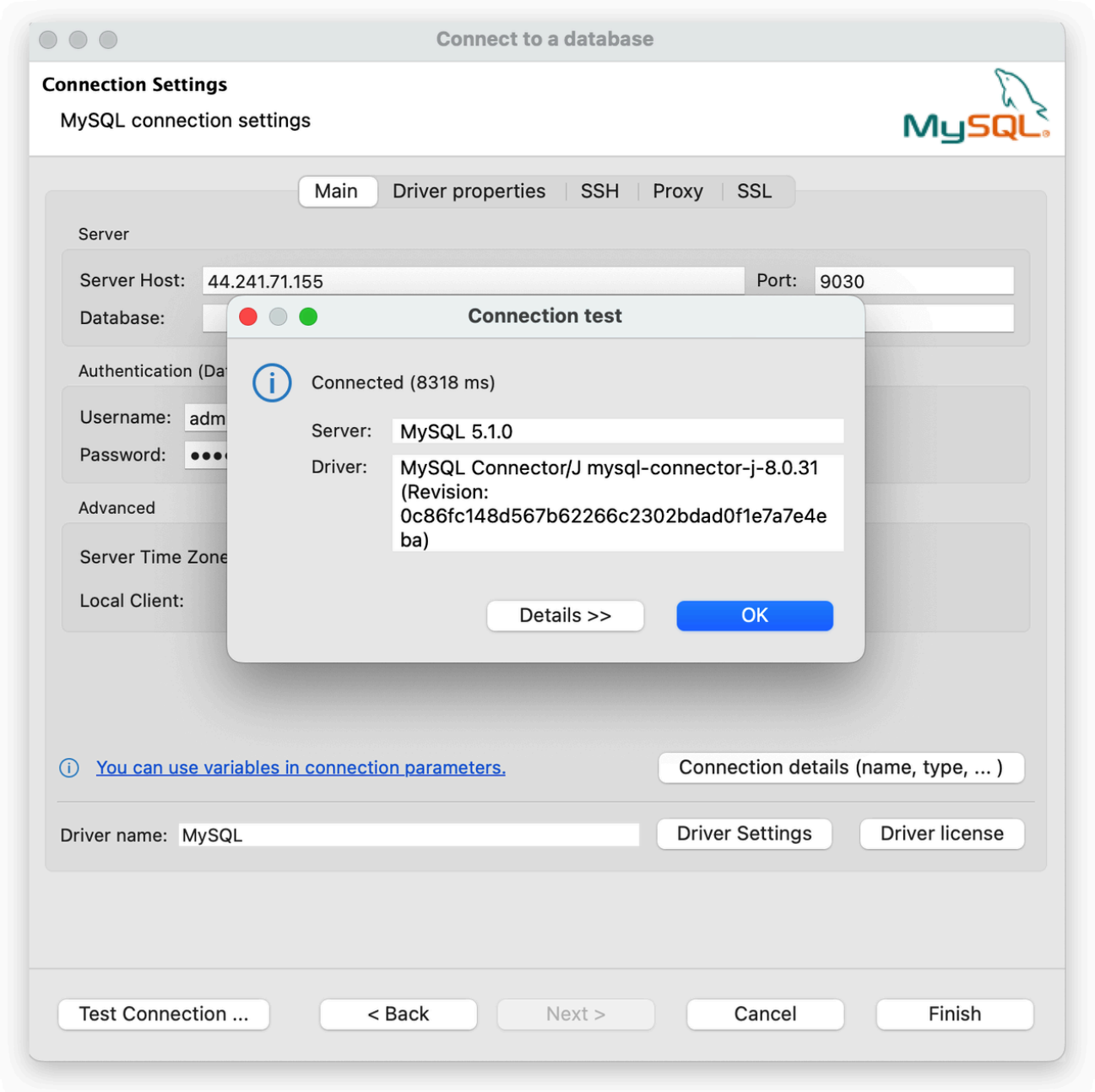


При необходимости вы также можете просмотреть и изменить свойства драйвера MySQL на вкладке **Driver properties**. Чтобы изменить определённое свойство, щёлкните по строке в столбце **Value** для этого свойства.



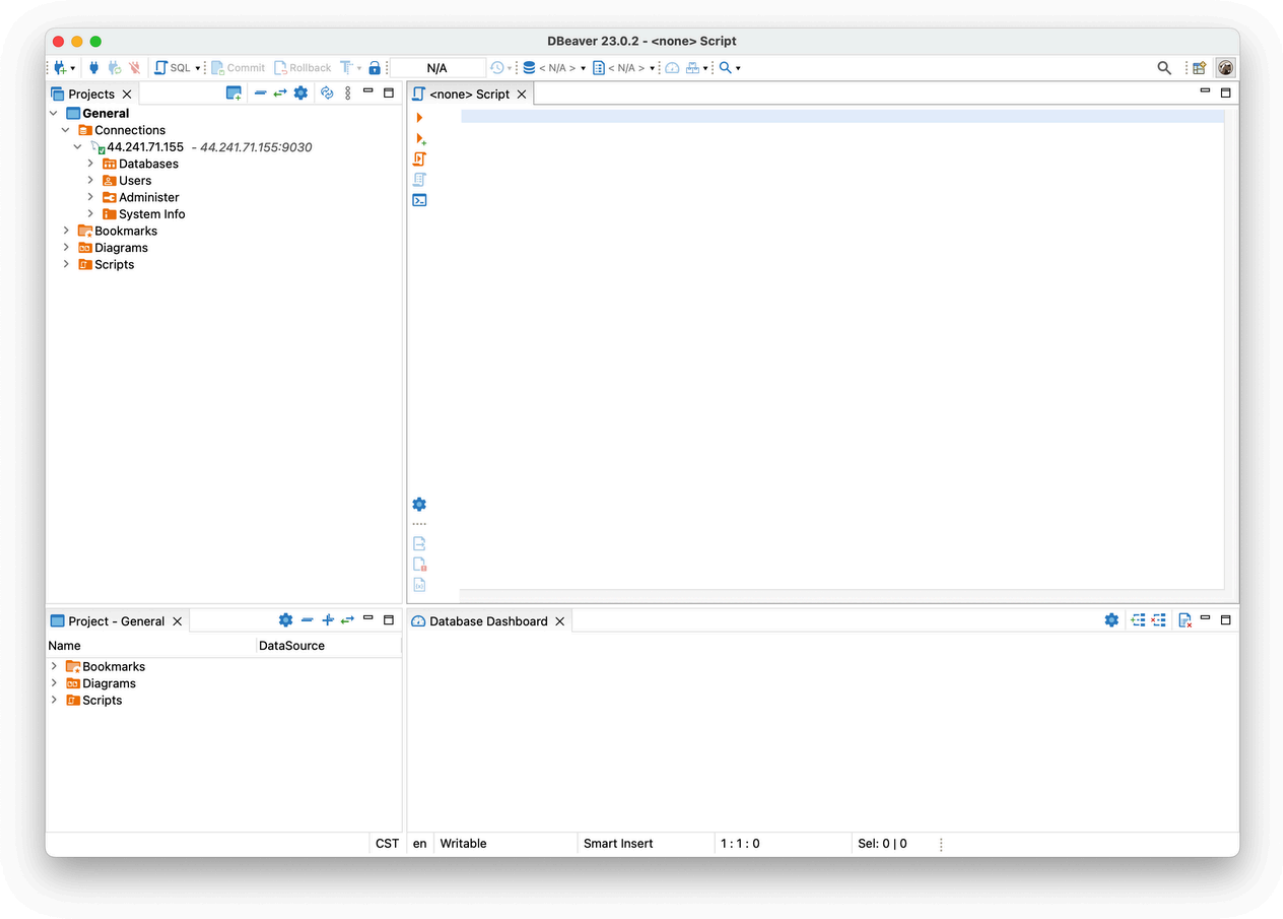
5. Протестируйте подключение к базе данных.

   Нажмите **Test Connection** для проверки правильности настроек подключения. Появится диалоговое окно с информацией о драйвере MySQL. Нажмите **OK** в диалоговом окне для подтверждения информации. После успешной настройки параметров подключения нажмите **Finish** для завершения процесса.



6. Подключитесь к базе данных.

После установки подключения вы можете увидеть его в дереве подключений к базам данных на левой панели, и DBeaver сможет эффективно подключаться к базе данных.



1.8.4 Jupyter

Это руководство описывает, как интегрировать ваш cluster Selena с Jupyter, новейшей веб-средой интерактивной разработки для блокнотов, кода и данных.

Всё это стало возможным благодаря JupySQL, который позволяет запускать SQL и строить графики больших наборов данных в Jupyter с помощью магических команд %sql, %%sql и %sqlplot.

Вы можете использовать JupySQL поверх Jupyter для выполнения запросов к Selena.

После загрузки данных в cluster вы можете запрашивать и визуализировать их с помощью SQL-графиков.

Предварительные требования

Перед началом работы на вашем компьютере должно быть установлено следующее программное обеспечение:

- JupySQL: `pip install jupysql`

- Jupyterlab: `pip install jupyterlab`
- SKlearn Evaluation: `pip install sklearn-evaluation`
- Python
- pymysql: `pip install pymysql`

> **ПРИМЕЧАНИЕ**
>
> После выполнения вышеуказанных требований вы можете открыть Jupyter lab, просто вызвав `jupyterlab` - это откроет интерфейс блокнота. Если Jupyter lab уже запущен в блокноте, вы можете просто запустить ячейку ниже для получения зависимостей.

```
# Установка необходимых пакетов.
%pip install --quiet jupysql sklearn-evaluation pymysql
```

> **ПРИМЕЧАНИЕ**
>
> Возможно, потребуется перезапустить ядро для использования обновлённых пакетов.

```
import pandas as pd
from sklearn_evaluation import plot

# Импорт расширения JupySQL Jupyter для создания SQL-ячеек.
%load_ext sql
%config SqlMagic.autocommit=False
```

Вам нужно убедиться, что ваш экземпляр Selena запущен и доступен для следующих этапов.

> **ПРИМЕЧАНИЕ**
>
> Вам нужно будет настроить строку подключения в соответствии с типом экземпляра, к которому вы пытаетесь подключиться (url, пользователь и пароль). В примере ниже используется локальный экземпляр.

Подключение к Selena через JupySQL

В этом примере используется экземпляр docker, и это отражено в строке подключения.

Пользователь `root` используется для подключения к локальному экземпляру Selena, создания базы данных и проверки того, что данные действительно могут быть прочитаны из таблицы и записаны в неё.

```
%sql mysql+pymysql://root:@localhost:9030
```

Создайте и используйте базу данных JupySQL:

```
%sql CREATE DATABASE jupysql;
```

```
%sql USE jupysql;
```

Создайте таблицу:

```
%%sql
CREATE TABLE tbl(c1 int, c2 int) distributed by hash(c1) properties
("replication_num" = "1");
INSERT INTO tbl VALUES (1, 10), (2, 20), (3, 30);
SELECT * FROM tbl;
```

Сохранение и загрузка запросов

Теперь, после создания базы данных, вы можете записать в неё некоторые примерные данные и запросить их.

JupySQL позволяет разбивать запросы на несколько ячеек, упрощая процесс построения больших запросов.

Вы можете писать сложные запросы, сохранять их и выполнять при необходимости, аналогично CTE в SQL.

```
# Это ожидает следующего релиза JupySQL.
%%sql --save initialize-table --no-execute
CREATE TABLE tbl(c1 int, c2 int) distributed by hash(c1) properties
("replication_num" = "1");
INSERT INTO tbl VALUES (1, 1), (2, 2), (3, 3);
SELECT * FROM tbl;
```

> **ПРИМЕЧАНИЕ**
>
> `--save` сохраняет запрос, а не данные.

Обратите внимание, что мы используем `--with;`, это извлечёт ранее сохранённые запросы и добавит их в начало (используя CTE). Затем мы сохраняем запрос в `track_fav`.

Построение графиков напрямую в Selena

JupySQL поставляется с несколькими графиками по умолчанию, позволяя визуализировать данные непосредственно в SQL.

Вы можете использовать столбчатую диаграмму для визуализации данных в вашей только что созданной таблице:

```
top_artist = %sql SELECT * FROM tbl
top_artist.bar()
```

Теперь у вас есть новая столбчатая диаграмма без какого-либо дополнительного кода. Вы можете запускать SQL прямо из своего блокнота через JupySQL (от ploomber). Это открывает множество возможностей вокруг Selena для специалистов по данным и инженеров. Если вы застряли или вам нужна поддержка, пожалуйста, свяжитесь с нами через Slack.

1.9 Мониторинг с помощью Datadog

В этом разделе описано, как интегрировать ваш cluster Selena с Datadog, платформой мониторинга и безопасности.

1.9.1 Предварительные требования

Перед началом работы на ваших instances должно быть установлено следующее программное обеспечение:

- Datadog Agent
- Python

> **ПРИМЕЧАНИЕ**
>
> При первой установке Datadog Agent также устанавливается Python как зависимость. Рекомендуется использовать этот Python на следующих шагах.

1.9.2 Подготовка исходного кода Selena

Поскольку Datadog пока не предоставляет готовый комплект интеграции для Selena, необходимо выполнить интеграцию с использованием исходного кода.

1. Запустите терминал, перейдите в локальную директорию, к которой у вас есть права на чтение и запись, и выполните следующую команду для создания выделенной директории для исходного кода Selena.

```
mkdir -p selena
```

2. Загрузите пакет исходного кода Selena с помощью следующей команды или с GitHub в созданную директорию.

```
cd selena
# Замените <selena_ver> на фактическую версию Selena, например,
"2.5.2".
wget
https://github.com/Selena/selena/archive/refs/tags/<selena_ver>.tar.gz
```

3. Распакуйте файлы из пакета.

```
# Замените <selena_ver> на фактическую версию Selena, например,
"2.5.2".
tar -xzvf <selena_ver>.tar.gz --strip-components 1
```

1.9.3 Установка и настройка комплекта интеграции FE

1. Установите комплект интеграции Datadog для FE из исходного кода.

```
/opt/datadog-agent/embedded/bin/pip install contrib/datadog-connector/selena_fe
```

2. Создайте конфигурационный файл интеграции FE **/etc/datadog-agent/conf.d/selena_fe.d/conf.yaml**.

```
sudo mkdir -p /etc/datadog-agent/conf.d/selena_fe.d
sudo cp contrib/datadog-connector/selena_fe/datadog_checks/selena_fe/data/conf.yaml.example /etc/datadog-agent/conf.d/selena_fe.d/conf.yaml
```

3. Измените конфигурационный файл интеграции FE **/etc/datadog-agent/conf.d/selena_fe.d/conf.yaml**.

   Примеры некоторых важных параметров конфигурации:

Config	Пример	Описание
fe_metric_url	`http://localhost:8030/metrics`	URL для доступа к метрикам Selena FE.
metrics	`- selena_fe_*`	Метрики для мониторинга на FE. Вы можете использовать символы подстановки `*` для соответствия параметрам конфигурации.

1.9.4 Установка и настройка комплекта интеграции BE

1. Установите комплект интеграции Datadog для BE из исходного кода.

```
/opt/datadog-agent/embedded/bin/pip install contrib/datadog-connector/selena_be
```

2. Создайте конфигурационный файл интеграции BE **/etc/datadog-agent/conf.d/selena_be.d/conf.yaml**.

```
sudo mkdir -p /etc/datadog-agent/conf.d/selena_be.d
sudo cp contrib/datadog-
connector/selena_be/datadog_checks/selena_be/data/conf.yaml.example
/etc/datadog-agent/conf.d/selena_be.d/conf.yaml
```

3. Измените конфигурационный файл интеграции BE **/etc/datadog-agent/conf.d/selena_be.d/conf.yaml**.

   Примеры некоторых важных параметров конфигурации:

Config	Пример	Описание
be_metric_url	`http://localhost:8040/metrics`	URL для доступа к метрикам Selena BE.
metrics	`- selena_be_*`	Метрики для мониторинга на BE. Вы можете использовать символы подстановки `*` для соответствия параметрам конфигурации.

1.9.5 Перезапуск Datadog Agent

Перезапустите Datadog Agent, чтобы конфигурация вступила в силу.

```
sudo systemctl stop datadog-agent
sudo systemctl start datadog-agent
```

1.9.6 Проверка интеграции

Инструкции по проверке интеграции см. в Datadog Application.

1.9.7 Удаление комплектов интеграции

Вы можете удалить комплекты интеграции, когда они больше не нужны.

- Чтобы удалить комплект интеграции FE, выполните следующую команду:

```
/opt/datadog-agent/embedded/bin/pip uninstall datadog-selena-fe
```

- Чтобы удалить комплект интеграции BE, выполните следующую команду:

```
/opt/datadog-agent/embedded/bin/pip uninstall datadog-selena-be
```

1.10 Apache Airflow

Обеспечивает оркестрацию и планирование рабочих процессов данных с Selena с использованием DAG (Directed Acyclic Graphs - направленных ациклических графов) и SQL-операторов. Используйте Airflow для загрузки и трансформации данных с помощью `SQLExecuteQueryOperator` и `MySQLHook` без какой-либо реализации или сложной конфигурации. Репозиторий Apache Airflow на GitHub.

1.10.1 Поддерживаемые функции

- Выполнение SQL через протокол MySQL
- Управление подключениями
- Поддержка транзакций
- Параметризованные запросы
- Зависимости задач
- Логика повторных попыток

1.10.2 Установка

Предварительные требования

- Apache Airflow 2.0+ или 3.0+
- Python 3.8+
- Доступ к cluster Selena (см. руководство по быстрому старту)

Установка

Пакет провайдера MySQL требуется для использования Selena, так как Selena использует протокол MySQL.

```
pip install apache-airflow-providers-mysql
```

Проверьте установку, проверив установленные провайдеры:

```
airflow providers list
```

Это должно показать `apache-airflow-providers-mysql` в выводе.

1.10.3 Конфигурация

Создание подключения Selena

Создайте подключение Selena в UI Airflow или через переменную окружения. Имя подключения будет использоваться DAG позже.

Через UI Airflow

1. Перейдите в Admin > Connections
2. Нажмите кнопку + для добавления нового подключения
3. Настройте подключение:

- Connection Id: `selena_default`
- Connection Type: MySQL
- Host: `your-selena-host.com`
- Schema: `your_database`
- Login: `your_username`
- Password: `your_password`
- Port: `9030`

Через CLI Airflow

```
airflow connections add 'selena_default' \
    --conn-type 'mysql' \
    --conn-host 'your-selena-host.com' \
    --conn-schema 'your_database' \
    --conn-login 'your_username' \
    --conn-password 'your_password' \
    --conn-port 9030
```

1.10.4 Примеры использования

Эти примеры демонстрируют общие паттерны интеграции Selena с Airflow. Каждый пример основывается на базовых концепциях, демонстрируя различные подходы к загрузке данных, трансформации и оркестрации рабочих процессов.

Что вы узнаете:

- **Загрузка данных**: Эффективная загрузка данных из CSV-файлов и облачного хранилища в Selena

- **Трансформация данных**: Выполнение SQL-запросов и обработка результатов с помощью Python
- **Продвинутые паттерны**: Реализация инкрементальной загрузки, асинхронных операций и оптимизации запросов
- **Лучшие практики для продакшена**: Корректная обработка ошибок и построение устойчивых конвейеров

Все примеры используют таблицы данных об авариях из Быстрого старта.

Загрузка данных

Потоковая загрузка данных

Эффективная загрузка больших CSV-файлов с использованием API Stream Load Selena. Stream Load - рекомендуемый подход для:

- Высокопроизводительной загрузки данных (поддерживает параллельные загрузки)
- Загрузки данных с трансформациями столбцов и фильтрацией

Stream Load обеспечивает лучшую производительность, чем операторы INSERT INTO VALUES для больших наборов данных, и включает встроенные функции, такие как допуск ошибок. Обратите внимание, что это требует, чтобы CSV-файл был доступен в файловой системе воркера Airflow.

```
from airflow.sdk import dag, task
from airflow.hooks.base import BaseHook
from datetime import datetime
import requests
from requests.auth import HTTPBasicAuth
from urllib.parse import urlparse


class PreserveAuthSession(requests.Session):
    """
    Пользовательская сессия, сохраняющая заголовок Authorization при
редиректах.
    Selena FE может перенаправлять запросы Stream Load на узлы BE.
    """
    def rebuild_auth(self, prepared_request, response):
        old = urlparse(response.request.url)
        new = urlparse(prepared_request.url)

        # Сохранять авторизацию только при редиректе на тот же хост
        if old.hostname == new.hostname:
            prepared_request.headers["Authorization"] =
response.request.headers.get("Authorization")
@dag(
    dag_id="selena_stream_load_example",
    schedule=None,
    start_date=datetime(2024, 1, 1),
    catchup=False,
    tags=["selena", "stream_load", "example"],
)
def selena_stream_load_example():
    @task
    def load_csv_to_selena():
        # Конфигурация
        DATABASE = "quickstart"
        TABLE = "crashdata"
        CSV_PATH = "/path/to/crashdata.csv"

        conn = BaseHook.get_connection("selena_default")
        url = f"http://{conn.host}:
{conn.port}/api/{DATABASE}/{TABLE}/_stream_load"

        # Генерация уникальной метки
        from airflow.sdk import get_current_context
        context = get_current_context()
        execution_date = context['logical_date'].strftime('%Y%m%d_%H%M%S')
```

```
        label = f"{TABLE}_load_{execution_date}"

        headers = {
            "label": label,
            "column_separator": ",",
            "skip_header": "1",
            "max_filter_ratio": "0.1",          # Допускать до 10% ошибок
            "Expect": "100-continue",
            "columns": """
                tmp_CRASH_DATE, tmp_CRASH_TIME,
                CRASH_DATE=str_to_date(concat_ws(' ', tmp_CRASH_DATE,
tmp_CRASH_TIME), '%m/%d/%Y %H:%i'),
                BOROUGH, ZIP_CODE, LATITUDE, LONGITUDE, LOCATION,
                ON_STREET_NAME, CROSS_STREET_NAME, OFF_STREET_NAME,
                NUMBER_OF_PERSONS_INJURED, NUMBER_OF_PERSONS_KILLED,
                NUMBER_OF_PEDESTRIANS_INJURED,
NUMBER_OF_PEDESTRIANS_KILLED,
                NUMBER_OF_CYCLIST_INJURED, NUMBER_OF_CYCLIST_KILLED,
                NUMBER_OF_MOTORIST_INJURED, NUMBER_OF_MOTORIST_KILLED,
                CONTRIBUTING_FACTOR_VEHICLE_1,
CONTRIBUTING_FACTOR_VEHICLE_2,
                CONTRIBUTING_FACTOR_VEHICLE_3,
CONTRIBUTING_FACTOR_VEHICLE_4,
                CONTRIBUTING_FACTOR_VEHICLE_5, COLLISION_ID,
                VEHICLE_TYPE_CODE_1, VEHICLE_TYPE_CODE_2,
                VEHICLE_TYPE_CODE_3, VEHICLE_TYPE_CODE_4,
VEHICLE_TYPE_CODE_5
            """.replace("\n", "").replace("  ", ""),
        }
        session = PreserveAuthSession()
        with open(CSV_PATH, "rb") as f:
            response = session.put(
                url,
                headers=headers,
                data=f,
                auth=HTTPBasicAuth(conn.login, conn.password or ""),
                timeout=3600,
            )

        result = response.json()
        print(f"\nОтвет Stream Load:")
        print(f"  Статус: {result.get('Status')}")
        print(f"  Загружено строк: {result.get('NumberLoadedRows', 0):,}")

        if result.get("Status") == "Success":
            return result
```

```
        else:
            error_msg = result.get("Message", "Unknown error")
            raise Exception(f"Stream Load завершился с ошибкой:
{error_msg}")
    load_csv_to_selena()

selena_stream_load_example()
```

Вставка из файлов

Используйте табличную функцию FILES() Selena для загрузки данных напрямую из файлов. Этот подход идеален для:

- Загрузки данных из S3, HDFS, Google Cloud Storage
- Одноэтапного поглощения данных с применением трансформаций во время загрузки
- Разовых загрузок данных из различных файловых источников

`FILES()` поддерживает несколько форматов файлов и систем хранения, что делает его гибкой альтернативой Stream Load для определённых случаев использования. Данные читаются и вставляются в одном SQL-операторе.

```
from airflow.sdk import dag, task
from airflow.providers.mysql.hooks.mysql import MySqlHook
from datetime import datetime

FILE_PATH = "path_to_file_here"

@dag(
    dag_id='crashdata_dynamic_files_load',
    schedule='@daily',
    start_date=datetime(2024, 1, 1),
    catchup=False,
    tags=['selena', 'files', 'dynamic'],
)
def crashdata_files():
    @task
    def load_file(file_path: str):
        hook = MySqlHook(mysql_conn_id='selena_default')

        sql = f"""
            INSERT INTO crashdata (
                CRASH_DATE, BOROUGH, ZIP_CODE, LATITUDE, LONGITUDE,
                LOCATION, ON_STREET_NAME, CROSS_STREET_NAME,
OFF_STREET_NAME,
                CONTRIBUTING_FACTOR_VEHICLE_1,
CONTRIBUTING_FACTOR_VEHICLE_2,
                COLLISION_ID, VEHICLE_TYPE_CODE_1, VEHICLE_TYPE_CODE_2
            )
            SELECT
                STR_TO_DATE(CONCAT_WS(' ', `CRASH DATE`, `CRASH TIME`),
'%m/%d/%Y %H:%i'),
                BOROUGH,
                `ZIP CODE`,
                CAST(LATITUDE as INT),
                CAST(LONGITUDE as INT),
                LOCATION,
                `ON STREET NAME`,
                `CROSS STREET NAME`,
                `OFF STREET NAME`,
                `CONTRIBUTING FACTOR VEHICLE 1`,
                `CONTRIBUTING FACTOR VEHICLE 2`,
                CAST(`COLLISION_ID` as INT),
                `VEHICLE TYPE CODE 1`,
                `VEHICLE TYPE CODE 2`
            FROM FILES(
                "path" = "s3://{file_path}",
```

```
                "format" = "parquet",
                "aws.s3.access_key" = "XXXXXXXXXX",
                "aws.s3.secret_key" = "YYYYYYYYYY",
                "aws.s3.region" = "us-west-2"
            )
        """
        result = hook.run(sql)

        return file_path

    load_file(FILE_PATH)

crashdata_files()
```

Трансформация данных

Выполняйте SQL-запросы к Selena для создания таблиц и вставки данных. Это полезно для:

- Настройки схемы базы данных
- Загрузки небольших наборов данных
- Выполнения разовых запросов

```
from airflow.sdk import dag, chain
from airflow.providers.common.sql.operators.sql import
SQLExecuteQueryOperator
from datetime import datetime

@dag(
    dag_id='crashdata_basic_setup',
    schedule='@daily',
    start_date=datetime(2024, 1, 1),
    catchup=False,
    tags=['selena', 'crashdata'],
)
def crashdata_basic_pipeline():
    """Создание таблицы crashdata и вставка примерных данных об авариях в
NYC."""

    create_table = SQLExecuteQueryOperator(
        task_id='create_crashdata_table',
        conn_id='selena_default',
        sql="""
            CREATE TABLE IF NOT EXISTS crashdata (
                CRASH_DATE DATETIME,
                BOROUGH STRING,
                ZIP_CODE STRING,
                LATITUDE INT,
                LONGITUDE INT,
                LOCATION STRING,
                ON_STREET_NAME STRING,
                CROSS_STREET_NAME STRING,
                OFF_STREET_NAME STRING,
                CONTRIBUTING_FACTOR_VEHICLE_1 STRING,
                CONTRIBUTING_FACTOR_VEHICLE_2 STRING,
                COLLISION_ID INT,
                VEHICLE_TYPE_CODE_1 STRING,
                VEHICLE_TYPE_CODE_2 STRING
            )
            DUPLICATE KEY(CRASH_DATE)
            DISTRIBUTED BY HASH(COLLISION_ID) BUCKETS 10
            PROPERTIES (
                "replication_num" = "1"
            )
        """,
    )

    insert_data = SQLExecuteQueryOperator(
```

```
        task_id='insert_sample_data',
        conn_id='selena_default',
        sql="""
            INSERT INTO crashdata VALUES
                ('2024-01-15 08:30:00', 'MANHATTAN', '10001', 40748817,
-73985428,
                 '(40.748817, -73.985428)', '5 AVENUE', 'WEST 34 STREET',
NULL,
                 'Driver Inattention/Distraction', 'Unspecified', 4567890,
'Sedan', 'Taxi'),
                ('2024-01-15 14:20:00', 'BROOKLYN', '11201', 40693139,
-73987664,
                 '(40.693139, -73.987664)', 'FLATBUSH AVENUE', 'ATLANTIC
AVENUE', NULL,
                 'Failure to Yield Right-of-Way', 'Unspecified', 4567891,
'SUV', 'Sedan'),
                ('2024-01-15 18:45:00', 'QUEENS', '11354', 40767689,
-73827426,
                 '(40.767689, -73.827426)', 'NORTHERN BOULEVARD', 'MAIN
STREET', NULL,
                 'Following Too Closely', 'Driver Inattention/Distraction',
4567892, 'Sedan', 'Sedan'),
                ('2024-01-16 09:15:00', 'BRONX', '10451', 40820679,
-73925300,
                 '(40.820679, -73.925300)', 'GRAND CONCOURSE', 'EAST 161
STREET', NULL,
                 'Unsafe Speed', 'Unspecified', 4567893, 'Truck', 'Sedan')
        """,
    )

    create_table >> insert_data

crashdata_basic_pipeline()
```

Более сложные операции с MySqlHook

Используйте MySqlHook для продвинутого анализа данных и обработки в задачах Python. Этот подход полезен для:

- Выполнения аналитических запросов и обработки результатов в Python
- Комбинирования запросов Selena с библиотеками Python (pandas, numpy и т.д.)
- Реализации сложной бизнес-логики, требующей как SQL, так и Python
- Создания проверок качества данных и рабочих процессов валидации

MySqlHook предоставляет полный программный доступ к результатам запросов, позволяя выполнять сложные трансформации и анализ данных в вашем DAG.

```python
from airflow.sdk import dag, task
from airflow.providers.mysql.hooks.mysql import MySqlHook
from datetime import datetime

@dag(
    dag_id='crashdata_python_analysis',
    schedule='@daily',
    start_date=datetime(2024, 1, 1),
    catchup=False,
    tags=['selena', 'python', 'analytics'],
)
def crashdata_python_pipeline():
    @task
    def analyze_crash_hotspots():
        """Определение горячих точек аварий по районам и улицам."""
        hook = MySqlHook(mysql_conn_id='selena_default')

        # Запрос для поиска локаций с высокой частотой аварий
        sql = """
            SELECT
                BOROUGH,
                ON_STREET_NAME,
                COUNT(*) as crash_count,
                COUNT(DISTINCT DATE(CRASH_DATE)) as days_with_crashes
            FROM crashdata
            WHERE ON_STREET_NAME IS NOT NULL
            GROUP BY BOROUGH, ON_STREET_NAME
            HAVING crash_count >= 3
            ORDER BY crash_count DESC
            LIMIT 10
        """

        results = hook.get_records(sql)

        print("Топ-10 горячих точек аварий:")
        for row in results:
            borough, street, count, days = row
            print(f"{borough:15} | {street:40} | {count:3} аварий за {days}
дней")

        return len(results)

    @task
    def calculate_contributing_factors():
        """Расчёт процентного распределения факторов, способствующих
```

```
авариям."""
        hook = MySqlHook(mysql_conn_id='selena_default')

        sql = """
            SELECT
                CONTRIBUTING_FACTOR_VEHICLE_1 as factor,
                COUNT(*) as count,
                ROUND(COUNT(*) * 100.0 / SUM(COUNT(*)) OVER (), 2) as
percentage
            FROM crashdata
            WHERE CONTRIBUTING_FACTOR_VEHICLE_1 != 'Unspecified'
            GROUP BY CONTRIBUTING_FACTOR_VEHICLE_1
            ORDER BY count DESC
        """

        results = hook.get_records(sql)

        print("\nАнализ факторов, способствующих авариям:")
        for factor, count, percentage in results:
            print(f"{factor:50} | {count:4} ({percentage}%)")

        return results

    # Определение порядка выполнения задач
    hotspots = analyze_crash_hotspots()
    factors = calculate_contributing_factors()

    hotspots >> factors

crashdata_python_pipeline()
```

Продвинутые паттерны

Инкрементальная загрузка данных

Загружайте данные инкрементально, чтобы избежать повторной обработки существующих записей. Инкрементальная загрузка необходима для:

- Эффективного обновления таблиц только новыми данными
- Сокращения времени обработки и использования ресурсов
- Управления большими наборами данных, растущими со временем
- Поддержания актуальности данных без полных перезагрузок

Этот паттерн использует промежуточные таблицы и фильтрацию на основе временных меток, чтобы гарантировать загрузку только новых записей, что делает его идеальным для запланированных пакетных обновлений.

© ООО «Платформа Данных Селена», 2026

```python
from airflow.sdk import dag, chain
from airflow.providers.common.sql.operators.sql import
SQLExecuteQueryOperator
from datetime import datetime

@dag(
    dag_id='crashdata_incremental_load',
    schedule='@hourly',
    start_date=datetime(2024, 1, 1),
    catchup=False,
    tags=['selena', 'incremental'],
)
def crashdata_incremental_pipeline():
    """Инкрементальная загрузка новых отчётов об авариях из промежуточной
таблицы."""

    create_staging = SQLExecuteQueryOperator(
        task_id='create_staging_table',
        conn_id='selena_default',
        sql="""
            CREATE TABLE IF NOT EXISTS crashdata_staging (
                CRASH_DATE DATETIME,
                BOROUGH STRING,
                ZIP_CODE STRING,
                LATITUDE INT,
                LONGITUDE INT,
                LOCATION STRING,
                ON_STREET_NAME STRING,
                CROSS_STREET_NAME STRING,
                OFF_STREET_NAME STRING,
                CONTRIBUTING_FACTOR_VEHICLE_1 STRING,
                CONTRIBUTING_FACTOR_VEHICLE_2 STRING,
                COLLISION_ID INT,
                VEHICLE_TYPE_CODE_1 STRING,
                VEHICLE_TYPE_CODE_2 STRING,
                loaded_at DATETIME
            )
            DUPLICATE KEY(CRASH_DATE)
            DISTRIBUTED BY HASH(COLLISION_ID) BUCKETS 10
            PROPERTIES ("replication_num" = "1")
        """,
    )

    incremental_load = SQLExecuteQueryOperator(
        task_id='load_new_crashes',
```

```
        conn_id='selena_default',
        sql="""
            INSERT INTO crashdata
            SELECT
                CRASH_DATE,
                BOROUGH,
                ZIP_CODE,
                LATITUDE,
                LONGITUDE,
                LOCATION,
                ON_STREET_NAME,
                CROSS_STREET_NAME,
                OFF_STREET_NAME,
                CONTRIBUTING_FACTOR_VEHICLE_1,
                CONTRIBUTING_FACTOR_VEHICLE_2,
                COLLISION_ID,
                VEHICLE_TYPE_CODE_1,
                VEHICLE_TYPE_CODE_2
            FROM crashdata_staging
            WHERE loaded_at >= '{{ data_interval_start }}'
              AND loaded_at < '{{ data_interval_end }}'
              AND COLLISION_ID NOT IN (SELECT COLLISION_ID FROM crashdata)
        """,
    )

    create_staging >> incremental_load

crashdata_incremental_pipeline()
```

Асинхронные крупномасштабные задания с `SUBMIT TASK`

Используйте `SUBMIT TASK` для долго выполняющихся запросов, которые не должны блокировать задачу Airflow. Этот паттерн полезен для:

- Сложных аналитических запросов, выполняющихся минуты или часы
- Крупномасштабных трансформаций данных (копирование таблиц, агрегации)
- Ресурсоёмких операций, которые могут завершиться по таймауту в синхронном режиме
- Параллельного выполнения нескольких тяжёлых запросов
- Разделения отправки задания и мониторинга завершения

`SUBMIT TASK` позволяет Airflow отслеживать долго выполняющиеся операции Selena без удержания открытых подключений к базе данных, улучшая эффективность использования ресурсов и надёжность.

```python
from airflow.sdk import dag, chain, task
from airflow.providers.mysql.hooks.mysql import MySqlHook
from datetime import datetime
import time

@dag(
    dag_id='crashdata_submit_task',
    schedule='@daily',
    start_date=datetime(2024, 1, 1),
    catchup=False,
    tags=['selena', 'submit-task'],
)
def crashdata_submit_task_pipeline():
    """
    Пример использования SUBMIT TASK для долго выполняющихся запросов.
    Требуется Selena 3.4+ для поддержки SUBMIT TASK.
    """

    @task
    def submit_long_running_query():
        """Отправка долго выполняющегося запроса как асинхронной задачи."""
        hook = MySqlHook(mysql_conn_id='selena_default')

        submit_sql = """
            SUBMIT TASK backup_crashdata AS
            CREATE TABLE crash_data_backup AS
            SELECT * FROM crashdata
        """
        conn = hook.get_conn()
        cursor = conn.cursor()
        cursor.execute(submit_sql)

        # Получение имени задачи из результата
        result = cursor.fetchone()
        task_name = result[0] if result else None

        cursor.close()
        conn.close()

        if task_name:
            print(f"Задача успешно отправлена: {task_name}")
            return task_name
        else:
            raise Exception("Не удалось отправить задачу")
```

```
    @task
    def monitor_task_completion(task_name: str):
        """Мониторинг отправленной задачи до завершения."""
        hook = MySqlHook(mysql_conn_id='selena_default')

        max_wait_time = 600  # 10 минут
        poll_interval = 10   # Проверка каждые 10 секунд
        elapsed_time = 0

        while elapsed_time < max_wait_time:
            conn = hook.get_conn()
            cursor = conn.cursor()

            # Проверка статуса задачи в information_schema
            check_sql = f"""
                SELECT STATE, ERROR_MESSAGE
                FROM information_schema.task_runs
                WHERE TASK_NAME = '{task_name}'
            """
            cursor.execute(check_sql)
            result = cursor.fetchone()

            if result:
                state, error_msg = result
                print(f"[{elapsed_time}c] Статус задачи: {state}")

                if state == 'SUCCESS':
                    cursor.close()
                    conn.close()
                    return {'status': 'SUCCESS', 'task_name': task_name}
                elif state == 'FAILED':
                    cursor.close()
                    conn.close()
                    raise Exception(f"Задача завершилась с ошибкой:
{error_msg}")

            cursor.close()
            conn.close()

            time.sleep(poll_interval)
            elapsed_time += poll_interval

        raise Exception(f"Задача не завершилась за {max_wait_time} секунд")

    @task
    def process_results():
```

```
        """Обработка или проверка результатов завершённой задачи."""
        print("Задача успешно завершена - результаты теперь доступны")
        return "Обработка завершена"

    # Определение потока задач
    task_name = submit_long_running_query()
    monitor_result = monitor_task_completion(task_name)
    result = process_results()

    chain(task_name, monitor_result, result)

crashdata_submit_task_pipeline()
```

Обратите внимание, что имя задачи уникально в Selena, поэтому будущим запускам может потребоваться квалификатор (например, uuid).

Материализованные представления

Создавайте и управляйте материализованными представлениями для ускорения производительности запросов. Материализованные представления идеальны для:

- Предварительного вычисления сложных агрегаций для дашбордов
- Ускорения часто выполняемых аналитических запросов
- Поддержания сводных таблиц, которые обновляются автоматически
- Сокращения вычислительных затрат путём избежания повторных вычислений
- Обслуживания аналитики реального времени из предагрегированных данных

Материализованные представления в Selena обновляются автоматически или по запросу, поддерживая актуальность агрегированных данных при значительном улучшении производительности запросов.

```python
from airflow.sdk import dag, task
from airflow.providers.mysql.hooks.mysql import MySqlHook
from datetime import datetime, timedelta


@dag(
    dag_id="selena_materialized_view_example",
    schedule="0 2 * * *",  # Запуск ежедневно в 2 часа ночи
    start_date=datetime(2024, 1, 1),
    catchup=False,
    tags=["selena", "materialized_view", "example"],
    doc_md=__doc__,
)
def selena_materialized_view_example():
    @task
    def create_materialized_view():
        hook = MySqlHook(mysql_conn_id="selena_conn")

        drop_sql = """
        DROP MATERIALIZED VIEW IF EXISTS quickstart.mv_daily_crash_stats
        """

        create_sql = """
        CREATE MATERIALIZED VIEW IF NOT EXISTS
quickstart.mv_daily_crash_stats
        DISTRIBUTED BY HASH(`crash_date`)
        REFRESH ASYNC
        AS
        SELECT
            DATE(CRASH_DATE) as crash_date,
            BOROUGH,
            COUNT(*) as total_crashes,
            COUNT(DISTINCT COLLISION_ID) as unique_collisions
        FROM quickstart.crashdata
        WHERE CRASH_DATE IS NOT NULL
        GROUP BY DATE(CRASH_DATE), BOROUGH
        """

        hook.run(drop_sql)
        hook.run(create_sql)

        return "mv_daily_crash_stats"

    @task
    def refresh_materialized_view(mv_name: str):
```

```
        hook = MySqlHook(mysql_conn_id="selena_conn")
        refresh_sql = f"REFRESH MATERIALIZED VIEW quickstart.{mv_name}"
        hook.run(refresh_sql)
        return mv_name

    @task
    def check_materialized_view_status(mv_name: str):
        hook = MySqlHook(mysql_conn_id="selena_conn")

        # Получение имени задачи для MV
        task_query = f"""
        SELECT TASK_NAME
        FROM information_schema.tasks
        WHERE `DATABASE` = 'quickstart'
        AND DEFINITION LIKE '%{mv_name}%'
        ORDER BY CREATE_TIME DESC
        LIMIT 1
        """

        task_name = hook.get_first(task_query)[0]

        # Получение последнего состояния запуска задачи
        state_query = f"""
        SELECT STATE
        FROM information_schema.task_runs
        WHERE TASK_NAME = '{task_name}'
        ORDER BY CREATE_TIME DESC
        LIMIT 1
        """

        state = hook.get_first(state_query)[0]

        print(f"MV: {mv_name} | Task: {task_name} | State: {state}")

        if state not in ('SUCCESS', 'RUNNING'):
            raise Exception(f"Обновление материализованного представления в
неожиданном состоянии: {state}")

        return {'task_name': task_name, 'state': state}


    create = create_materialized_view()
    refresh = refresh_materialized_view(create)
    status = check_materialized_view_status(refresh)

    status
```

```
selena_materialized_view_example()
```

Обработка ошибок

Реализуйте надёжную обработку ошибок для производственной надёжности. Правильная обработка ошибок критична для:

- Автоматического восстановления после временных сбоев (сетевые проблемы, таймауты)
- Предотвращения нарушений конвейера данных из-за временных проблем
- Обеспечения видимости паттернов сбоев

Встроенные механизмы повторных попыток Airflow обрабатывают большинство временных ошибок.

```
from airflow.sdk import dag
from airflow.providers.common.sql.operators.sql import
SQLExecuteQueryOperator
from datetime import datetime, timedelta

@dag(
    dag_id='selena_with_retries',
    schedule='@daily',
    start_date=datetime(2024, 1, 1),
    catchup=False,
    default_args={
        'retries': 3,
        'retry_delay': timedelta(minutes=5),
        'retry_exponential_backoff': True,
        'max_retry_delay': timedelta(hours=1),
    },
)
def selena_resilient_pipeline():
    critical_query = SQLExecuteQueryOperator(
        task_id='critical_query',
        conn_id='selena_default',
        sql='SELECT * FROM important_table',
    )
selena_resilient_pipeline()
```

Устранение неполадок

- Убедитесь, что порт 9030 доступен из экземпляра Airflow
- Протестируйте подключение (если включено) из UI Airflow
- При использовании localhost используйте 127.0.0.1 вместо него

1.11 Dbt

`dbt-selena` позволяет использовать `dbt` для преобразования данных в Selena с применением паттернов моделирования и лучших практик dbt.

`dbt-selena` GitHub repo.

> **EXPERIMENTAL FEATURE**
>
> Рекомендации по использованию экспериментальных функций

1.11.1 Поддерживаемые функции

Selena >= 3.1	Selena >= 3.4	Функция
□	□	Материализация таблиц
□	□	Материализация представлений
□	□	Материализация материализованных представлений
□	□	Инкрементальная материализация
□	□	Модель с первичным ключом
□	□	Источники
□	□	Пользовательские тесты данных
□	□	Генерация документации
□	□	Партиционирование по выражениям
×	×	Kafka
×	□	Dynamic Overwrite
*	□	Submit task
□	□	Microbatch (Insert Overwrite)
×	□	Microbatch (Dynamic Overwrite)

`*` Проверьте поддержку `submit task` для вашей версии, см. SUBMIT TASK

1.11.2 Установка

Установите адаптер Selena DBT с помощью pip:

```
pip install dbt-selena
```

1.11.3 Проверка установки

Проверьте установку, проверив версию:

```
dbt --version
```

В списке plugins должен отображаться `selena`.

1.11.4 Конфигурация

Профили

Создайте или обновите `profiles.yml` с настройками для Selena.

```yml
selena_project:
  target: dev
  outputs:
    dev:
      type: selena
      host: your-selena-host.com
      port: 9030
      schema: your_database
      username: your_username
      password: your_password
      catalog: test_catalog
```

Параметры

`type`

Описание: Конкретный адаптер для использования, должен быть установлен в `selena`
Обязательный?: Обязательный
Пример: `selena`

`host`

Описание: Имя хоста для подключения
Обязательный?: Обязательный
Пример: `192.168.100.28`

`port`

Описание: Порт для использования
Обязательный?: Обязательный
Пример: `9030`

`catalog`

Описание: Указывает catalog для построения моделей
Обязательный?: Необязательный
Пример: `default_catalog`

`schema`

Описание: Указывает schema (база данных в Selena) для построения моделей
Обязательный?: Обязательный
Пример: `analytics`

`username`

Описание: Имя пользователя для подключения к серверу
Обязательный?: Обязательный
Пример: `dbt_admin`

`password`

Описание: Пароль для аутентификации на сервере
Обязательный?: Обязательный
Пример: `correct-horse-battery-staple`

`version`

Описание: Позволяет плагину попытаться использовать совместимую версию selena
Обязательный?: Необязательный
Пример: `3.1.0`

`use_pure`

Описание: установите "true" для использования расширений C
Обязательный?: Необязательный
Пример: `true`

`is_async`

Описание: "true" для отправки подходящих задач как etl задач.
Обязательный?: Необязательный
Пример: `true`

`async_query_timeout`

Описание: Устанавливает значение `query_timeout` при отправке задачи в Selena
Обязательный?: Необязательный
Пример: `300`

Источники

Создайте или обновите `sources.yml`

```yaml
sources:
  - name: your_source
    database: your_sr_catalog
    schema: your_sr_database
    tables:
      - name: your_table
```

Если catalog не указан в schema, будет использован catalog, определенный в профиле. Используя профиль из предыдущего примера, если catalog не определен, модель будет считать, что источник находится в `test_catalog.your_sr_database`.

1.11.5 Материализации

Таблица

Базовая конфигурация таблицы

```
{{ config(
    materialized='table',
    engine='OLAP',
    keys=['id', 'name', 'created_date'],
    table_type='PRIMARY',
    distributed_by=['id'],
    buckets=3,
    partition_by=['created_date'],
    properties=[
        {"replication_num": "1"}
    ]
) }}


SELECT
    id,
    name,
    email,
    created_date,
    last_modified_date
FROM {{ source('your_source', 'users') }}
```

1.11.6 Параметры конфигурации

- engine: Движок хранения (по умолчанию: `OLAP`)
- keys: Столбцы, которые определяют ключ сортировки
- table_type: Тип модели таблицы
  - `PRIMARY`: Модель с первичным ключом (поддерживает upsert и удаления)
  - `DUPLICATE`: Модель с дублирующимся ключом (позволяет дублирующиеся строки)
  - `UNIQUE`: Модель с уникальным ключом (обеспечивает уникальность)
- `distributed_by`: Столбцы для hash-распределения
- `buckets`: Количество buckets для распределения данных (оставьте пустым для автоматического bucketing)
- `partition_by`: Столбцы для партиционирования таблицы
- `partition_by_init`: Начальные определения партиций
- `properties`: Дополнительные свойства таблицы Selena

1.11.7 Таблицы во внешних Catalogs

Чтение из внешнего источника в Selena

Этот пример создает материализованную таблицу в Selena, содержащую агрегированные данные из внешнего Hive catalog.

ПОДСКАЗКА

Настройте внешний catalog, если он еще не существует:

```
CREATE EXTERNAL CATALOG `hive_external`
PROPERTIES (
    "hive.metastore.uris"  =  "thrift://127.0.0.1:8087",
    "type"="hive"
);
```

```
{{ config(
    materialized='table',
    keys=['product_id', 'order_date'],
    distributed_by=['product_id'],
    partition_by=['order_date']
) }}

-- Агрегирование данных из внешнего Hive catalog в таблицу Selena
SELECT
    h.product_id,
    h.order_date,
    COUNT(*) as order_count,
    SUM(h.amount) as total_amount,
    MAX(h.last_updated) as last_updated
FROM {{ source('hive_external', 'orders') }} h
GROUP BY
    h.product_id,
    h.order_date
```

Запись во внешний источник

```
{{
  config(
    materialized='table',
    on_table_exists = 'replace',
    partition_by=['order_date'],
    properties={},
    catalog='external_catalog',
    database='test_db'
  )
}}

SELECT * FROM {{ source('iceberg_external', 'orders') }}
```

Конфигурация для материализации во внешние catalogs поддерживает меньше опций. Поддерживаются `on_table_exists`, `partition_by` и `properties`. Если `catalog` и `database` не установлены, будут использованы значения по умолчанию из профиля.

Инкрементальная

Инкрементальные материализации также поддерживаются в Selena:

```
{{ config(
    materialized='incremental',
    unique_key='id',
    table_type='PRIMARY',
    keys=['id'],
    distributed_by=['id'],
    incremental_strategy='default'
) }}

SELECT
    id,
    user_id,
    event_name,
    event_timestamp,
    properties
FROM {{ source('raw', 'events') }}

{% if is_incremental() %}
    WHERE event_timestamp > (SELECT MAX(event_timestamp) FROM {{ this }})
{% endif %}
```

Стратегии инкрементальной загрузки

`dbt-selena` поддерживает несколько стратегий инкрементальной загрузки:

1. `append` (по умолчанию): Просто добавляет новые записи без дедупликации
2. `insert_overwrite`: Перезаписывает партиции таблицы при вставке
3. `dynamic_overwrite`: Перезаписывает, создает и записывает партиции таблицы

Для получения дополнительной информации о том, какую стратегию перезаписи использовать, см. документацию INSERT.

> **ПРИМЕЧАНИЕ**
>
> В настоящее время инкрементальное слияние не поддерживается.

1.11.8 Устранение неполадок

- Перед использованием внешних catalogs в dbt, вы должны создать их в Selena. Документация по этому вопросу находится здесь.

- Доступ к внешним источникам должен осуществляться с использованием макроса `{{ source('external_source_name', 'table_name' }}`.
- `dbt seed` не тестировался для внешних catalogs и в настоящее время не поддерживается.
- Для того чтобы `dbt` мог создавать модели во внешних базах данных, которые в настоящее время не существуют, расположение моделей должно быть установлено через свойства.
- Внешние модели должны определять расположение, в котором они хранятся. Это расположение будет определено, если целевая база данных существует и устанавливает свойство location. В противном случае расположение необходимо установить.
    - В настоящее время мы поддерживаем создание внешних моделей только в базах данных, которые уже существуют.

1.12 Управление схемами и миграция с помощью SQLAlchemy и Alembic

Это руководство описывает, как управлять схемами Selena с использованием экосистемы Python — включая SQLAlchemy, Alembic и sqlacodegen — через диалект `selena` **SQLAlchemy**. Оно помогает понять, **почему миграция схем полезна** и **как эффективно использовать её с Selena**.

1.12.1 Обзор

Многие пользователи управляют таблицами, представлениями и материализованными представлениями Selena напрямую с помощью SQL DDL. Однако по мере роста проектов ручное сопровождение операторов `ALTER TABLE` становится подверженным ошибкам и сложным для отслеживания.

Диалект **Selena SQLAlchemy (`selena`)** предоставляет:

- Полноценный слой моделей SQLAlchemy для **таблиц**, **представлений** и **материализованных представлений** Selena
- **Декларативные** определения для схем таблиц и свойств таблиц (включая представления и материализованные представления)
- Интеграцию с **Alembic** для автоматического **обнаружения** и **генерации** изменений схемы
- Совместимость с инструментами типа **sqlacodegen** для обратной генерации моделей

Это позволяет пользователям Python поддерживать схемы Selena **декларативным**, **версионируемым** и **автоматизированным** способом.

1.12.2 Ключевые преимущества

Хотя миграция схем традиционно ассоциируется с OLTP базами данных, она также ценна в системах хранилищ данных, таких как Selena. Команды используют Alembic вместе с диалектом Selena благодаря преимуществам, перечисленным ниже.

Декларативное определение схемы

После того как вы определите схему в моделях Python ORM или в стиле SQLAlchemy core, вам больше не требуется вручную писать операторы `ALTER TABLE`.

Автоматическое сравнение и автогенерация

Alembic сравнивает **текущую схему Selena** с **вашими моделями SQLAlchemy** и автоматически генерирует скрипты миграции (`CREATE`/`DROP`/`ALTER`).

Проверяемые, версионируемые миграции

Каждое изменение схемы становится файлом миграции (Python), поэтому пользователи могут отслеживать изменения и откатывать их при необходимости.

Единообразный workflow для всех окружений

Изменения схемы можно применять к development, staging и production с использованием одного и того же процесса.

1.12.3 Установка и подключение

Предварительные требования**

- Python-клиент Selena: 1.3.2 или новее
- `SQLAlchemy`: 1.4 или новее (рекомендуется SQLAlchemy 2.0; требуется для использования `sqlacodegen`)
- `Alembic`: 1.16 или новее

Установка Python-клиента Selena

Выполните следующую команду для установки Python-клиента Selena.

```
pip install selena
```

Подключение к Selena

Подключитесь к вашему cluster Selena с использованием следующего URL.

```
selena://<user>:<password>@<FE_host>:<query_port>/[<catalog>.]<database>
```

- `user`: Имя пользователя для подключения к cluster.
- `password`: Пароль пользователя.
- `FE_host`: IP-адрес FE.
- `query_port`: `query_port` FE (по умолчанию: 9030).
- `catalog`: Имя catalog, в котором находится ваша база данных.
- `database`: Имя базы данных, к которой вы хотите подключиться.

После установки вы можете быстро проверить подключение с помощью следующего примера кода:

```
from sqlalchemy import create_engine, text

# сначала нужно создать `mydatabase`
engine = create_engine("selena://root@localhost:9030/mydatabase")

with engine.connect() as conn:
    conn.execute(text("SELECT 1")).fetchall()
    print("Connection successful!")
```

1.12.4 Определение моделей Selena (декларативный ORM)

Диалект Selena поддерживает:

- Таблицы
- Представления
- Материализованные представления

Он также поддерживает специфичные для Selena атрибуты таблиц, такие как:

- `ENGINE` (OLAP)

- Модели ключей (`DUPLICATE KEY`, `PRIMARY KEY`, `UNIQUE KEY`, `AGGREGATE KEY`)
- Варианты `PARTITION BY` (RANGE / LIST / Expression partitioning)
- Варианты `DISTRIBUTED BY` (HASH / RANDOM)
- `ORDER BY`
- Свойства таблиц (например, `replication_num`, `storage_medium`)

> **IMPORTANT**
>
> - Опции диалекта Selena передаются как именованные аргументы с префиксом `selena_`.
> - **Префикс `selena_` должен быть в нижнем регистре**. Суффикс принимается в любом регистре (например, `PRIMARY_KEY` и `primary_key`).
> - Если вы указываете ключ таблицы (например, `selena_primary_key="id"`), задействованные колонки **также должны** быть помечены `primary_key=True` в `Column(...)`, чтобы метаданные SQLAlchemy и автогенерация Alembic работали корректно.

Примеры ниже отражают реальный публичный API и имена параметров.

Пример таблицы

Опции таблиц Selena могут быть указаны как в стиле ORM (через `__table_args__`), так и в стиле Core (через `Table(..., selena_...=...)`).

Стиль ORM (декларативный)

```
from sqlalchemy import create_engine
from sqlalchemy.orm import Mapped, declarative_base, mapped_column
from selena import INTEGER, STRING

# используем тот же engine, что и в быстром тесте
engine = create_engine("selena://root@localhost:9030/mydatabase")

Base = declarative_base()

class MyTable(Base):
    __tablename__ = 'my_orm_table'
    id: Mapped[int] = mapped_column(INTEGER, primary_key=True)
    name: Mapped[str] = mapped_column(STRING)

    __table_args__ = {
        'comment': 'table comment',

        'selena_primary_key': 'id',
        'selena_distributed_by': 'HASH(id) BUCKETS 10',
        'selena_properties': {'replication_num': '1'}
    }

# Создание таблицы в базе данных
Base.metadata.create_all(engine)
```

Стиль Core

```
from sqlalchemy import Column, MetaData, Table, create_engine
from selena import INTEGER, VARCHAR

# используем тот же engine, что и в быстром тесте
engine = create_engine("selena://root@localhost:9030/mydatabase")

metadata = MetaData()

my_core_table = Table(
    'my_core_table',
    metadata,
    Column('id', INTEGER, primary_key=True),
    Column('name', VARCHAR(50)),

    # Специфичные для Selena аргументы
    selena_primary_key='id',
    selena_distributed_by='HASH(id) BUCKETS 10',
    selena_properties={"replication_num": "1"}
)

# Создание таблицы в базе данных
metadata.create_all(engine)
```

> **ПРИМЕЧАНИЕ**
>
> Для полного справочника по атрибутам таблиц и типам данных см. Справочник [4].

Пример представления

Ниже приведен рекомендуемый стиль определения представления, использующий `columns` как список словарей (`name`/`comment`). Этот пример основан на существующей таблице `my_core_table`.

```python
from selena.schema import View

# Повторно используем metadata из примера Core-таблицы выше
metadata = my_core_table.metadata

user_view = View(
    "user_view",
    metadata,
    definition="SELECT id, name FROM my_core_table WHERE name IS NOT NULL",
    columns=[
        {"name": "id", "comment": "ID"},
        {"name": "name", "comment": "Name"},
    ],
    comment="Active users",
)
```

> **ПРИМЕЧАНИЕ**
>
> Для получения дополнительной информации об опциях View и ограничениях см. Справочник [5].

Пример материализованного представления

Материализованные представления определяются аналогично. Свойство `selena_refresh` — это строка синтаксиса, указывающая стратегию обновления.

```python
from selena.schema import MaterializedView

# Повторно используем metadata из примера Core-таблицы выше
metadata = my_core_table.metadata

# Создание простого материализованного представления (асинхронное
обновление)
user_stats_ = MaterializedView(
    'user_stats_',
    metadata,
    definition='SELECT id, COUNT(*) AS cnt FROM my_core_table GROUP BY id',
    selena_refresh='ASYNC'
)
```

> **ПРИМЕЧАНИЕ**
>
> Для получения дополнительной информации об опциях и ограничениях ALTER см. Справочник [6].

1.12.5 Интеграция с Alembic

Диалект Selena SQLAlchemy обеспечивает полную поддержку:

- Создание / удаление таблицы
- Создание / удаление представления
- Создание / удаление материализованного представления
- Обнаружение поддерживаемых изменений в специфичных для Selena атрибутах (например, свойства таблиц и распределение)

Это позволяет корректно работать **автогенерации** Alembic.

Инициализация Alembic

1. Инициализируйте Alembic:

   ```
   alembic init migrations
   ```

2. Настройте URL вашей базы данных в `alembic.ini`:

   ```
   # alembic.ini
   sqlalchemy.url = selena://<user>:<password>@<FE_host>:
   <query_port>/[<catalog>.]<database>
   ```

3. Включите логирование диалекта Selena (опционально):

   Вы можете включить логгер `selena` в `alembic.ini`, чтобы отслеживать обнаруженные изменения таблицы через логи. Подробнее см. Справочник [2].

   Отредактируйте `env.py` (настройте оба пути — offline и online):

```
from alembic import context
from selena.alembic import render_column_type, include_object_for_view_
from selena.alembic.selena import SelenaImpl  # noqa: F401
(обеспечиваем регистрацию impl)

from myapp.models import Base  # настройте под ваш проект

target_metadata = Base.metadata


def run_migrations_offline() -> None:
    url = context.config.get_main_option("sqlalchemy.url")
    context.configure(
        url=url,
        target_metadata=target_metadata,
        render_item=render_column_type,
        include_object=include_object_for_view_
    )

    with context.begin_transaction():
        context.run_migrations()


def run_migrations_online() -> None:
    # ... создание engine и подключение как в стандартном env.py из
alembic ...
    with connectable.connect() as connection:
        context.configure(
            connection=connection,
            target_metadata=target_metadata,
            render_item=render_column_type,
            include_object=include_object_for_view_
        )

        with context.begin_transaction():
            context.run_migrations()
```

Автоматическая генерация миграций

```
alembic revision --autogenerate -m "initial schema"
```

Alembic сравнит модели SQLAlchemy с фактической схемой Selena и выведет корректный DDL.

Применение миграций

```
alembic upgrade head
```

Откат также поддерживается (где это возможно).

> **IMPORTANT**
>
> DDL в Selena не транзакционен для нескольких операторов. Если upgrade прерывается, вам может потребоваться проверить, что уже было применено, и **выполнить ручное исправление** (например, написать компенсирующую миграцию или выполнить ручной DDL) перед повторным запуском.

1.12.6 Поддерживаемые операции изменения схемы

Диалект поддерживает автогенерацию Alembic для:

- **Таблиц**: создание / удаление, и сравнение специфичных для Selena атрибутов, объявленных через `selena_*` (в рамках поддержки ALTER в Selena)
- **Представлений**: создание / удаление / изменение (в основном изменения, связанные с определением; некоторые атрибуты неизменяемы)
- **Материализованных представлений**: создание / удаление / изменение (ограничено изменяемыми частями, такими как стратегии обновления и свойства)

Некоторые изменения DDL в Selena не обратимы или не изменяемы. Вы можете внести эти изменения только путем удаления и пересоздания таблицы/представления/материализованного представления. Если вы укажете такие изменения в диалекте, автогенерация **выдаст предупреждение или ошибку**.

1.12.7 Сквозной пример (рекомендуется для начинающих)

Этот раздел показывает выполнимый сквозной workflow, включая места для паузы и проверки сгенерированных файлов.

Шаг 1. Создайте директорию проекта и инициализируйте Alembic

```
mkdir my_sr_alembic_project
cd my_sr_alembic_project

alembic init alembic
```

Шаг 2. Настройте `alembic.ini`

Отредактируйте URL в `alembic.ini`:

```
sqlalchemy.url = selena://root@localhost:9030/mydatabase
```

Шаг 3. Определите ваши модели

Создайте пакет для ваших моделей:

```
mkdir -p myapp
touch myapp/__init__.py
```

Создайте `myapp/models.py` и поместите определения таблиц/представлений/материализованных представлений в пакет:

> **ПРИМЕЧАНИЕ**
>
> При использовании миграций Alembic не вызывайте `metadata.create_all(engine)` в модуле моделей.

```python
from sqlalchemy import Column, Table
from sqlalchemy.orm import Mapped, declarative_base, mapped_column

from selena import INTEGER, STRING, VARCHAR
from selena.schema import MaterializedView, View

Base = declarative_base()


# --- ORM-таблица ---
class MyOrmTable(Base):
    __tablename__ = "my_orm_table"

    id: Mapped[int] = mapped_column(INTEGER, primary_key=True)
    name: Mapped[str] = mapped_column(STRING)

    __table_args__ = {
        "comment": "table comment",
        "selena_primary_key": "id",
        "selena_distributed_by": "HASH(id) BUCKETS 10",
        "selena_properties": {"replication_num": "1"},
    }


# --- Core-таблица в тех же metadata (важно для target_metadata в Alembic)
---
my_core_table = Table(
    "my_core_table",
    Base.metadata,
    Column("id", INTEGER, primary_key=True),
    Column("name", VARCHAR(50)),
    comment="core table comment",
    selena_primary_key="id",
    selena_distributed_by="HASH(id) BUCKETS 10",
    selena_properties={"replication_num": "1"},
)


# --- Представление ---
user_view = View(
    "user_view",
    Base.metadata,
    definition="SELECT id, name FROM my_core_table WHERE name IS NOT NULL",
    columns=[
        {"name": "id", "comment": "ID"},
```

```
        {"name": "name", "comment": "Name"},
    ],
    comment="Active users",
)


# --- Материализованное представление ---
user_stats_mv = MaterializedView(
    "user_stats_mv",
    Base.metadata,
    definition="SELECT id, COUNT(*) AS cnt FROM my_core_table GROUP BY id",
    selena_refresh="ASYNC",
)
```

Шаг 4. Настройте `env.py` для автогенерации

Отредактируйте `alembic/env.py`:

1. Импортируйте `myapp.models` для установки `target_metadata`.
2. Импортируйте `render_column_type` и `include_object_for_view_mv` для их установки как в `run_migrations_offline()`, так и в `run_migrations_online()`, чтобы корректно обрабатывать представления и материализованные представления, а также корректно отображать типы колонок Selena.

> **ПРИМЕЧАНИЕ**
>
> Вам нужно добавить или изменить эти строки в `env.py`, а не заменять весь сгенерированный файл `env.py`.

```
from alembic import context
from selena.alembic import render_column_type, include_object_for_view_mv
from selena.alembic.selena import SelenaImpl  # noqa: F401

from myapp.models import Base

target_metadata = Base.metadata

# Опционально: установите репликацию таблицы версий для dev-clusters с
одним BE
version_table_kwargs = {"selena_properties": {"replication_num": "1"}}

# В обеих функциях run_migrations_offline() и run_migrations_online()
убедитесь:
def run_migrations_offline() -> None:
    url = context.config.get_main_option("sqlalchemy.url")
    context.configure(
        url=url,
        target_metadata=target_metadata,
        literal_binds=True,
        render_item=render_column_type,
        include_object=include_object_for_view_mv,
        version_table_kwargs=version_table_kwargs,
    )


def run_migrations_online() -> None:
    # ... создание engine и подключение как в стандартном env.py из alembic
...
    with connectable.connect() as connection:
        context.configure(
            connection=connection,
            target_metadata=target_metadata,
            render_item=render_column_type,
            include_object=include_object_for_view_mv,
            version_table_kwargs=version_table_kwargs,
        )
```

Шаг 5. Автогенерация первой ревизии

```
alembic revision --autogenerate -m "create initial schema"
```

Пауза и проверка:

1. Проверьте сгенерированный файл миграции в `alembic/versions/`.
2. Убедитесь, что он содержит ожидаемые операции (например, `create_table`, `create_view`, `create_materialized_view`).
3. Убедитесь, что он не содержит неожиданных удалений или изменений.

Шаг 6. Предпросмотр SQL и применение

Предпросмотр SQL:

```
alembic upgrade head --sql
```

Пауза и проверка:

1. Убедитесь, что DDL в ожидаемом порядке.
2. Определите любые потенциально тяжелые операции и рассмотрите разбиение миграций при необходимости.

Применение:

```
alembic upgrade head
```

> **IMPORTANT**
>
> DDL в Selena не транзакционен для нескольких операторов. Если upgrade прерывается, вам может потребоваться проверить, что уже было применено, и выполнить ручное исправление перед повторным запуском.

Шаг 7. Внесите изменение и повторите автогенерацию

Обновите `myapp/models.py`:

- **Измените существующую таблицу** (`my_core_table`): добавьте колонку или обновите комментарий таблицы, и измените одно свойство таблицы.
- **Добавьте новую таблицу** (`my_new_table`).

> **ПРИМЕЧАНИЕ**
>
> Добавление колонки может быть длительным изменением схемы. Selena допускает только **одну выполняющуюся задачу изменения схемы на таблицу** одновременно. На практике рекомендуется разделять изменения "добавление/удаление/изменение колонок" от других тяжелых изменений (например, дополнительные добавления/удаления колонок или массовые изменения свойств) и разбивать их на несколько ревизий Alembic при необходимости.

```
from sqlalchemy import Column, Table
from selena import INTEGER, VARCHAR

# Изменение существующей таблицы (добавление колонки)
# (Обновите существующее определение my_core_table на месте.)
my_core_table = Table(
    "my_core_table",
    Base.metadata,
    Column("id", INTEGER, primary_key=True),
    Column("name", VARCHAR(50)),
    Column("age", INTEGER),  # только добавленная колонка

    selena_primary_key='id',
    selena_distributed_by='HASH(id) BUCKETS 10',
    selena_properties={"replication_num": "1"},
)

my_new_table = Table(
    "my_new_table",
    Base.metadata,
    Column("id", INTEGER, primary_key=True),
    Column("name", VARCHAR(50)),
    selena_primary_key="id",
    selena_distributed_by="HASH(id) BUCKETS 10",
    selena_properties={"replication_num": "1"},
)
```

```
alembic revision --autogenerate -m "add a new table, change a old table"
```

Пауза и проверка:

Убедитесь, что новая миграция содержит:

- `create_table(...)` для `my_new_table`, и
- ожидаемые операции для изменений `my_core_table` (например, добавление колонки / установка комментария / установка свойств).

Предпросмотр SQL и применение:

```
alembic upgrade head --sql
alembic upgrade head
```

1.12.8 Использование sqlacodegen

`sqlacodegen` может выполнить обратную генерацию моделей SQLAlchemy напрямую из Selena:

```
sqlacodegen --options include_dialect_options,keep_dialect_types \
  --generator tables \
  selena://<user>:<password>@<FE_host>:<query_port>/[catalog.]<database> >
models.py
```

Поддерживаемые объекты:

- Таблицы
- Представления
- Материализованные представления
- Секционирование, распределение, order-by и свойства

Это полезно при подключении существующей схемы Selena к Alembic.

Вы можете напрямую использовать приведенную выше команду для генерации Python-скрипта для таблиц/представлений/материализованных представлений, определенных в разделе **Сквозной пример**.

> **ПРИМЕЧАНИЕ**
>
> - Рекомендуется добавлять `--generator tables` при генерации моделей в стиле Core (ORM-генераторы могут переупорядочивать колонки согласно атрибуту `NOT NULL` / `NULL`).
> - Ключевые колонки могут быть сгенерированы как `NOT NULL`. Если вы хотите сделать их nullable, настройте сгенерированную модель вручную.

1.12.9 Ограничения и лучшие практики

- Некоторые операции DDL в Selena требуют удаления и пересоздания таблицы; автогенерация выдаст предупреждение или ошибку, а не молча создаст недоступный SQL.
- Изменения моделей ключей (например, изменение DUPLICATE KEY на PRIMARY KEY) не поддерживаются через `ALTER TABLE`; используйте явный план (обычно удаление и пересоздание с заполнением данных).
- Selena не предоставляет транзакционный DDL для нескольких операторов; проверяйте сгенерированные миграции и применяйте их операционно. Если миграция прерывается, вам может потребоваться обработать откат **вручную**.
- Для распределения, если вы опускаете часть `BUCKETS`, Selena может автоматически назначить количество buckets; диалект спроектирован так, чтобы избегать лишних различий в этом случае.

1.12.10 Резюме

С диалектом Selena SQLAlchemy и интеграцией с Alembic вы можете:

- ✔ Использовать декларативные модели для определения схем Selena
- ✔ Автоматически обнаруживать и генерировать скрипты миграции схем
- ✔ Использовать контроль версий для эволюции схемы
- ✔ Управлять представлениями и материализованными представлениями декларативно
- ✔ Выполнять обратную инженерию существующих схем с помощью sqlacodegen

Это приводит управление схемами Selena в современную экосистему Python data engineering и значительно упрощает согласованность схем между окружениями.

1.12.11 Справочники

[1]: selena-python-client README

[2]: Интеграция с Alembic

[3]: Детали SQLAlchemy

[4]: Поддержка таблиц

[5]: Поддержка представлений

[6]: Поддержка материализованных представлений

2 Администрирование

2.1 Конфигурация

Параметры конфигурации для узлов FE и BE.

2.1.1 Конфигурация FE

Параметры FE делятся на динамические и статические.

- Динамические параметры можно настраивать и изменять с помощью SQL-команд, что очень удобно. Однако эти настройки становятся недействительными при перезапуске FE. Поэтому мы рекомендуем также изменять параметры конфигурации в файле **fe.conf**, чтобы предотвратить потерю изменений.

- Статические параметры можно настраивать и изменять только в конфигурационном файле FE **fe.conf**. **После изменения этого файла необходимо перезапустить FE, чтобы изменения вступили в силу.**

Является ли параметр динамическим, указано в столбце `IsMutable` в выводе ADMIN SHOW CONFIG. `TRUE` означает динамический параметр.

Обратите внимание, что как динамические, так и статические параметры FE можно настраивать в файле **fe.conf**.

Просмотр параметров конфигурации FE

После запуска FE вы можете выполнить команду ADMIN SHOW FRONTEND CONFIG в вашем MySQL-клиенте для проверки конфигурации параметров. Если вы хотите запросить конфигурацию конкретного параметра, выполните следующую команду:

```
ADMIN SHOW FRONTEND CONFIG [LIKE "pattern"];
```

Для подробного описания возвращаемых полей см. ADMIN SHOW CONFIG.

> **ПРИМЕЧАНИЕ**
>
> У вас должны быть права администратора для выполнения команд, связанных с администрированием cluster.

Настройка параметров FE

Настройка динамических параметров FE

Вы можете настроить или изменить настройки динамических параметров FE с помощью ADMIN SET FRONTEND CONFIG.

```
ADMIN SET FRONTEND CONFIG ("key" = "value");
```

> **ПРИМЕЧАНИЕ**
>
> Изменения конфигурации, сделанные с помощью `ADMIN SET FRONTEND`, будут сброшены к значениям по умолчанию из файла `fe.conf` после перезапуска FE. Поэтому мы рекомендуем также изменить параметры конфигурации в `fe.conf`, если вы хотите сделать изменения постоянными.

Настройка статических параметров FE

> **ПРИМЕЧАНИЕ**
>
> Статические параметры FE задаются путём их изменения в конфигурационном файле **fe.conf** и перезапуска FE для вступления изменений в силу.

Описание параметров FE

Logging

log_roll_size_mb

- Default: 1024
- Type: Int
- Unit: MB
- Is mutable: No
- Description: Максимум size of a system файл лога or an audit файл лога.

- Introduced in: -

sys_log_dir

- Default: SelenaFE.SELENA_HOME_DIR + "/log"
- Type: String
- Unit: -
- Is mutable: No
- Description: Директория которая хранит system файл логаs.
- Introduced in: -

sys_log_level

- Default: INFO
- Type: String
- Unit: -
- Is mutable: No
- Description: The severity levels into which system записей лога are classified. Допустимые значения: `INFO`, `WARN`, `ERROR`, and `FATAL`.
- Introduced in: -

sys_log_roll_num

- Default: 10
- Type: Int
- Unit: -
- Is mutable: No
- Description: Максимальное количество system файл логаs that может сохраняться within каждый период хранения указанного параметром the `sys_log_roll_interval` parameter.
- Introduced in: -

sys_log_verbose_modules

- Default: Empty string
- Type: String[]
- Unit: -
- Is mutable: No
- Description: The modules for which Selena генерирует системный логs. Если этот параметр установлен в `org.apache.selena.catalog`, Selena генерирует системный логs only for the catalog module. Разделите the имена модулей запятой (,) и пробелом.

- Introduced in: -

sys_log_roll_interval

- Default: DAY
- Type: String
- Unit: -
- Is mutable: No
- Description: Интервал времени с которым Selena ротирует system записей лога. Допустимые значения: `DAY` and `HOUR`.
  - Если этот параметр установлен в `DAY`, суффикс в формате `yyyyMMdd` добавляется к именам of system файл логаs.
  - Если этот параметр установлен в `HOUR`, суффикс в формате `yyyyMMddHH` добавляется к именам of system файл логаs.
- Introduced in: -

sys_log_delete_age

- Default: 7d
- Type: String
- Unit: -
- Is mutable: No
- Description: The период хранения of system файл логаs. The default value `7d` указывает, что каждый system файл лога может сохраняться for 7 days. Selena проверяет каждый system файл лога and удаляет те которые были сгенерированы 7 дней назад.
- Introduced in: -

audit_log_dir

- Default: SelenaFE.SELENA_HOME_DIR + "/log"
- Type: String
- Unit: -
- Is mutable: No
- Description: Директория которая хранит audit файл логаs.
- Introduced in: -

audit_log_roll_num

- Default: 90
- Type: Int

- Unit: -
- Is mutable: No
- Description: Максимальное количество audit файл логas that может сохраняться within каждый период хранения указанного параметром the `audit_log_roll_interval` parameter.
- Introduced in: -

audit_log_modules

- Default: slow_query, query
- Type: String[]
- Unit: -
- Is mutable: No
- Description: The modules for which Selena генерирует audit записей лога. По умолчанию, Selena генерирует лог аудитas for the `slow_query` module and the `query` module. The `connection` module поддерживается начиная с версии v1.5.2. Разделите the имена модулей запятой (,) и пробелом.
- Introduced in: -

qe_slow_log_ms

- Default: 5000
- Type: Long
- Unit: Milliseconds
- Is mutable: Yes
- Description: Порог используется для determine whether a query is a медленный запрос. If the время ответа of a query exceeds этот порог, it записывается как a медленный запрос in **fe.audit.log**.
- Introduced in: -

audit_log_roll_interval

- Default: DAY
- Type: String
- Unit: -
- Is mutable: No
- Description: Интервал времени с которым Selena ротирует audit записей лога. Допустимые значения: `DAY` and `HOUR`.
  - Если этот параметр установлен в `DAY`, суффикс в формате `yyyyMMdd` добавляется к именам of audit файл логas.

  - Если этот параметр установлен в `HOUR`, суффикс в формате `yyyyMMddHH` добавляется к именам of audit файл логаs.
- Introduced in: -

audit_log_delete_age

- Default: 30d
- Type: String
- Unit: -
- Is mutable: No
- Description: The период хранения of audit файл логаs. The default value `30d` указывает, что каждый audit файл лога может сохраняться for 30 days. Selena проверяет каждый audit файл лога and удаляет те которые были сгенерированы 30 дней назад.
- Introduced in: -

dump_log_dir

- Default: SelenaFE.SELENA_HOME_DIR + "/log"
- Type: String
- Unit: -
- Is mutable: No
- Description: Директория которая хранит dump файл логаs.
- Introduced in: -

dump_log_roll_num

- Default: 10
- Type: Int
- Unit: -
- Is mutable: No
- Description: Максимальное количество dump файл логаs that может сохраняться within каждый период хранения указанного параметром the `dump_log_roll_interval` parameter.
- Introduced in: -

dump_log_modules

- Default: query
- Type: String[]
- Unit: -
- Is mutable: No

- Description: The modules for which Selena генерирует dump записей лога. По умолчанию, Selena генерирует лог дампas for the query module. Разделите the имена модулей запятой (,) и пробелом.
- Introduced in: -

dump_log_roll_interval

- Default: DAY
- Type: String
- Unit: -
- Is mutable: No
- Description: Интервал времени с которым Selena ротирует dump записей лога. Допустимые значения: `DAY` and `HOUR`.
  - Если этот параметр установлен в `DAY`, суффикс в формате `yyyyMMdd` добавляется к именам of dump файл логas.
  - Если этот параметр установлен в `HOUR`, суффикс в формате `yyyyMMddHH` добавляется к именам of dump файл логas.
- Introduced in: -

dump_log_delete_age

- Default: 7d
- Type: String
- Unit: -
- Is mutable: No
- Description: The период хранения of dump файл логas. The default value `7d` указывает, что каждый dump файл лога может сохраняться for 7 days. Selena проверяет каждый dump файл лога and удаляет те которые были сгенерированы 7 дней назад.
- Introduced in: -

log_cleaner_disk_util_based_enable

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Enable automatic log cleaning based on disk usage. When enabled, logs будет cleaned when disk usage exceeds the threshold. The log cleaner runs as a background daemon on the узел FE and helps prevent disk space exhaustion from файл лога accumulation.

- Introduced in: -

log_cleaner_disk_usage_threshold

- Default: 80
- Type: Int
- Unit: Percentage
- Is mutable: Yes
- Description: Disk usage threshold (нacentage) to trigger log cleaning. When disk usage exceeds этот порог, log cleaning will start. The cleaner проверяет каждый configured log directory independently and обрабатывает directories that exceed этот порог.
- Introduced in: -

log_cleaner_disk_usage_target

- Default: 60
- Type: Int
- Unit: Percentage
- Is mutable: Yes
- Description: Target disk usage (нacentage) after log cleaning. Log cleaning will continue until disk usage drops below этот порог. The cleaner deletes the oldest файл лoгas one by one until the target is гкаждыйed.
- Introduced in: -

log_cleaner_audit_log_min_retention_days

- Default: 3
- Type: Int
- Unit: Days
- Is mutable: Yes
- Description: Minimum retention days for audit файл лoгas. Audit файл лoгas newer than this will not be deleted even if disk usage is high. This гарантирует, что лог аудитas are preserved for compliance and troubleshooting purposes.
- Introduced in: -

log_cleaner_check_interval_second

- Default: 300
- Type: Int
- Unit: Seconds
- Is mutable: Yes

- Description: Interval в секундах для проверки disk usage and clean logs. The cleaner наiodicвсеу проверяет каждый log directory's disk usage and triggers cleaning when necessary. Default is 300 seconds (5 minutes).
- Introduced in: -

Server

frontend_address

- Default: 0.0.0.0
- Type: String
- Unit: -
- Is mutable: No
- Description: The IP address of the узел FE.
- Introduced in: -

dns_cache_ttl_seconds

- Default: 60
- Type: Int
- Unit: Seconds
- Is mutable: No
- Description: DNS cache TTL (Time-To-Live) в секундах for successful DNS lookups. This sets the Java security pronaty `networkaddress.cache.ttl` which controls how long the JVM caches successful DNS lookups. Set to -1 to cache forever, or 0 чтобы отключить caching. This is particularly useful in environments where IP addresses change frequently, such as Kubernetes deployments or when using dynamic DNS. This is a static configuration and requires FE restart to take effect.
- Introduced in: v1.5.2, v2.0.0

priority_networks

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: Declares a selection strategy for servers that иметь multiple IP addresses. Note that at most one IP address must match the list указанного параметром this parameter. The value of this parameter is a list that consists of entries, which are separated with semicolons (;) in CIDR notation, such as 10.10.10.0/24. If no IP address matches the

entries in this list, an available IP address of the server будет randomly selected. From v1.5.2, Selena supports deployment based on IPv6. If the server имеет both IPv4 and IPv6 addresses, and this parameter is not specified, the system uses an IPv4 address по умолчанию. You can change this behavior by настройка `net_use_ipv6_when_priority_networks_empty` to `true`.

- Introduced in: -

net_use_ipv6_when_priority_networks_empty

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: A boolean value чтобы управлять whether to use IPv6 addresses preferentiвсey when `priority_networks` is not specified. `true` означает to всеow the system to use an IPv6 address preferentiвсey when the server that hosts the node имеет both IPv4 and IPv6 addresses and `priority_networks` is not specified.
- Introduced in: v1.5.2

http_port

- Default: 8030
- Type: Int
- Unit: -
- Is mutable: No
- Description: The port on which the HTTP server in the узел FE listens.
- Introduced in: -

http_worker_threads_num

- Default: 0
- Type: Int
- Unit: -
- Is mutable: No
- Description: Number of worker threads for http server to deal with http requests. For a negative or 0 value, the number of threads будет twice the number of cpu cores.
- Introduced in: v1.5.2

http_backlog_num

- Default: 1024

- Type: Int
- Unit: -
- Is mutable: No
- Description: The length of the backlog queue held by the HTTP server in the узел FE.
- Introduced in: -

enable_http_async_handler

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, всеow the system to process HTTP-запрос asynchronously. If this feature включён, an HTTP-запрос received by Netty worker threads will then be submitted to a separate thread pool for service logic handling to avoid blocking the HTTP server. If disabled, Netty workers will handle the service logic.
- Introduced in: 4.0.0

enable_https

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Определяет, enable HTTPS server alongside HTTP server in узел FEs.
- Introduced in: v2.0.0

https_port

- Default: 8443
- Type: Int
- Unit: -
- Is mutable: No
- Description: The port on which the HTTPS server in the узел FE listens.
- Introduced in: v2.0.0

http_async_threads_num

- Default: 4096
- Type: Int
- Unit: -

- Is mutable: Yes
- Description: Size of the thread pool for asynchronous HTTP-запрос processing. The alias is `max_http_sql_service_task_threads_num`.
- Introduced in: 4.0.0

cluster_name

- Default: Selena Cluster
- Type: String
- Unit: -
- Is mutable: No
- Description: The name of the Selena cluster to which the FE belongs. The cluster name is displayed for `Title` on the web page.
- Introduced in: -

rpc_port

- Default: 9020
- Type: Int
- Unit: -
- Is mutable: No
- Description: The port on which the Thrift server in the узел FE listens.
- Introduced in: -

thrift_server_max_worker_threads

- Default: 4096
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество worker threads that поддерживаются by the Thrift server in the узел FE.
- Introduced in: -

thrift_server_queue_size

- Default: 4096
- Type: Int
- Unit: -
- Is mutable: No

- Description: The length of queue where requests are ожидающий. If the number of threads that are being processed in the thrift server exceeds the value specified in `thrift_server_max_worker_threads`, new requests are added to the ожидающий queue.
- Introduced in: -

thrift_client_timeout_ms

- Default: 5000
- Type: Int
- Unit: Milliseconds
- Is mutable: No
- Description: The length of time after which idle client connections time out.
- Introduced in: -

thrift_backlog_num

- Default: 1024
- Type: Int
- Unit: -
- Is mutable: No
- Description: The length of the backlog queue held by the Thrift server in the узел FE.
- Introduced in: -

brpc_idle_wait_max_time

- Default: 10000
- Type: Int
- Unit: ms
- Is mutable: No
- Description: Максимум length of time for which bRPC clients wait as in the idle state.
- Introduced in: -

query_port

- Default: 9030
- Type: Int
- Unit: -
- Is mutable: No
- Description: The port on which the MySQL server in the узел FE listens.
- Introduced in: -

mysql_nio_backlog_num

- Default: 1024
- Type: Int
- Unit: -
- Is mutable: No
- Description: The length of the backlog queue held by the MySQL server in the узел FE.
- Introduced in: -

mysql_service_nio_enable_keep_alive

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Enable TCP Keep-Alive for MySQL connections. Useful for long-idled connections behind load balancers.
- Introduced in: -

mysql_service_io_threads_num

- Default: 4
- Type: Int
- Unit: -
- Is mutable: No
- Description: Максимальное количество threads that может быть run by the MySQL server in the узел FE to process I/O events.
- Introduced in: -

max_mysql_service_task_threads_num

- Default: 4096
- Type: Int
- Unit: -
- Is mutable: No
- Description: Максимальное количество threads that может быть run by the MySQL server in the узел FE to process tasks.
- Introduced in: -

mysql_server_version

- Default: 8.0.33

- Type: String
- Unit: -
- Is mutable: Yes
- Description: The MySQL server version returned to the client. Modifying this parameter will affect the version information in the following situations:
  i. `select version();`
  ii. Handshake packet version
  iii. Value of the global variable `version` (`show variables like 'version';`)
- Introduced in: -

qe_max_connection

- Default: 4096
- Type: Int
- Unit: -
- Is mutable: No
- Description: Максимальное количество connections that может быть established by все users to the узел FE. From v1.5.2 onwards, the default value имеет been changed from `1024` to `4096`.
- Introduced in: -

Metadata and cluster management

cluster_id

- Default: -1
- Type: Int
- Unit: -
- Is mutable: No
- Description: The ID of the Selena cluster to which the FE belongs. FEs or BEs that иметь the same cluster ID belong to the same Selena cluster. Допустимые значения: любой positive integer. The default value `-1` указывает, что Selena will generate a random cluster ID for the Selena cluster at the time when the leader FE of the cluster is started for the first time.
- Introduced in: -

meta_dir

- Default: SelenaFE.SELENA_HOME_DIR + "/meta"
- Type: String

- Unit: -
- Is mutable: No
- Description: Директория которая хранит metadata.
- Introduced in: -

edit_log_type

- Default: BDB
- Type: String
- Unit: -
- Is mutable: No
- Description: The type of edit log that может быть generated. Set the value to `BDB`.
- Introduced in: -

edit_log_port

- Default: 9010
- Type: Int
- Unit: -
- Is mutable: No
- Description: The port that is используется для communication among the Leader, Follower, and Observer FEs in the cluster.
- Introduced in: -

edit_log_roll_num

- Default: 50000
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество metadata записей лога that может быть written before a файл лога is created for these записей лога. Этот параметр is используется для control the size of файл логаs. The new файл лога is written to the BDBJE database.
- Introduced in: -

metadata_ignore_unknown_онаation_type

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes

- Description: Определяет, ignore an unknown log ID. When an FE is rolled back, the FEs of the earlier version may be unable to recognize some log IDs. Если значение is `TRUE`, the FE ignores unknown log IDs. Если значение is `FALSE`, the FE exits.
- Introduced in: -

meta_delay_toleration_second

- Default: 300
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Максимум duration by which the metadata on the follower and observer FEs can lag behind that on the leader FE. Unit: seconds. If this duration is exceeded, the non-leader FEs stops providing services.
- Introduced in: -

master_sync_policy

- Default: SYNC
- Type: String
- Unit: -
- Is mutable: No
- Description: The policy based on which the leader FE flushes logs to disk. Этот параметр is valid только когда the current FE is a leader FE. Valid values:
  - `SYNC`: When a transaction is committed, a log entry is generated and flushed to disk simultaneously.
  - `NO_SYNC`: The generation and flushing of a log entry do not occur at the same time when a transaction is committed.
  - `WRITE_NO_SYNC`: When a transaction is committed, a log entry is generated simultaneously but is not flushed to disk.

  If you иметь deployed only one follower FE, we recommend that you set this parameter to `SYNC`. If you иметь deployed three or more follower FEs, we recommend that you set this parameter and the `replica_sync_policy` both to `WRITE_NO_SYNC`.
- Introduced in: -

replica_sync_policy

- Default: SYNC
- Type: String
- Unit: -
- Is mutable: No
- Description: The policy based on which the follower FE flushes logs to disk. Этот параметр is valid только когда the current FE is a follower FE. Valid values:
  - `SYNC`: When a transaction is committed, a log entry is generated and flushed to disk simultaneously.
  - `NO_SYNC`: The generation and flushing of a log entry do not occur at the same time when a transaction is committed.
  - `WRITE_NO_SYNC`: When a transaction is committed, a log entry is generated simultaneously but is not flushed to disk.
- Introduced in: -

replica_ack_policy

- Default: SIMPLE_MAJORITY
- Type: String
- Unit: -
- Is mutable: No
- Description: The policy based on which a log entry is considered valid. The default value `SIMPLE_MAJORITY` указывает, что a log entry is considered valid if a majority of follower FEs return ACK messages.
- Introduced in: -

bdbje_heartbeat_timeout_second

- Default: 30
- Type: Int
- Unit: Seconds
- Is mutable: No
- Description: The amount of time after which the heartbeats among the leader, follower, and observer FEs in the Selena cluster time out.
- Introduced in: -

bdbje_replica_ack_timeout_second

- Default: 10
- Type: Int

- Unit: Seconds
- Is mutable: No
- Description: Максимум amount of time for which the leader FE can wait for ACK messages from a specified number of follower FEs when metadata is written from the leader FE to the follower FEs. Unit: second. If a large amount of metadata is being written, the follower FEs require a long time before they can return ACK messages to the leader FE, causing ACK timeout. In this situation, metadata writes fail, and the FE process exits. We recommend that you increase the value of this parameter to prevent this situation.
- Introduced in: -

bdbje_lock_timeout_second

- Default: 1
- Type: Int
- Unit: Seconds
- Is mutable: No
- Description: The amount of time after which a lock in the BDB JE-based FE times out.
- Introduced in: -

bdbje_reset_election_group

- Default: false
- Type: String
- Unit: -
- Is mutable: No
- Description: Определяет, reset the BDBJE replication group. Если этот параметр установлен в `TRUE`, the FE will reset the BDBJE replication group (that is, remove the information of все electable узел FEs) and start as the leader FE. After the reset, this FE будет the only member in the cluster, and other FEs can rejoin this cluster by using `ALTER SYSTEM ADD/DROP FOLLOWER/OBSERVER 'xxx'`. Use this настройка только когда no leader FE может быть elected because the data of most follower FEs иметь been damaged. `reset_election_group` is используется для replace `metadata_failure_recovery`.
- Introduced in: -

max_bdbje_clock_delta_ms

- Default: 5000
- Type: Long
- Unit: Milliseconds

- Is mutable: No
- Description: Максимум clock offset that is всеowed between the leader FE and the follower or observer FEs in the Selena cluster.
- Introduced in: -

txn_rollback_limit

- Default: 100
- Type: Int
- Unit: -
- Is mutable: No
- Description: Максимальное количество transactions that может быть rolled back.
- Introduced in: -

heartbeat_mgr_threads_num

- Default: 8
- Type: Int
- Unit: -
- Is mutable: No
- Description: Количество threads that может быть run by the Heartbeat Manager to run heartbeat tasks.
- Introduced in: -

heartbeat_mgr_blocking_queue_size

- Default: 1024
- Type: Int
- Unit: -
- Is mutable: No
- Description: The size of the blocking queue которая хранит heartbeat tasks run by the Heartbeat Manager.
- Introduced in: -

catalog_try_lock_timeout_ms

- Default: 5000
- Type: Long
- Unit: Milliseconds
- Is mutable: Yes
- Description: The timeout duration to obtain the global lock.

- Introduced in: -

ignore_materialized_view_error

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, FE ignores the metadata exception caused by materialized view errors. If FE fails to start due to the metadata exception caused by materialized view errors, you can set this parameter to `true` to всеow FE to ignore the exception.
- Introduced in: v1.5.2

ignore_meta_check

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, non-Leader FEs ignore the metadata gap from the Leader FE. Если значение is TRUE, non-Leader FEs ignore the metadata gap from the Leader FE and continue providing data reading services. Этот параметр ensures continuous data reading services even when you stop the Leader FE for a long наiod of time. Если значение is FALSE, non-Leader FEs do not ignore the metadata gap from the Leader FE and stop providing data reading services.
- Introduced in: -

drop_backend_after_decommission

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, delete a BE after the BE is decommissioned. `TRUE` означает that the BE is deleted immediately after it is decommissioned. `FALSE` означает that the BE is not deleted after it is decommissioned.
- Introduced in: -

enable_collect_query_detail_info

- Default: false

- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, collect the profile of a query. Если этот параметр установлен в `TRUE`, the system collects the profile of the query. Если этот параметр установлен в `FALSE`, the system does not collect the profile of the query.
- Introduced in: -

profile_info_format

- Default: default
- Type: String
- Unit: -
- Is mutable: Yes
- Description: The format of the Profile output by the system. Допустимые значения: `default` and `json`. When set to `default`, Profile is of the default format. When set to `json`, the system outputs Profile in JSON format.
- Introduced in: v1.5.2

enable_background_refresh_connector_metadata

- Default: true in v1.5.2 and later and false in v1.5.2
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable the haiodic Hive metadata cache refresh. After it включён, Selena polls the metastore (Hive Metastore or AWS Glue) of your Hive cluster, and refreshes the cached metadata of the frequently accessed Hive catalogs to haceive data changes. `true` означает чтобы включить the Hive metadata cache refresh, and `false` означает чтобы отключить it.
- Introduced in: v1.5.2

background_refresh_metadata_interval_millis

- Default: 600000
- Type: Int
- Unit: Milliseconds
- Is mutable: Yes
- Description: The interval between two consecutive Hive metadata cache refreshes.
- Introduced in: v1.5.2

background_refresh_metadata_time_secs_since_last_access_secs

- Default: 3600 * 24
- Type: Long
- Unit: Seconds
- Is mutable: Yes
- Description: The expiration time of a Hive metadata cache refresh task. For the Hive catalog that имеет been accessed, if it имеет not been accessed for more than the specified time, Selena stops refreshing its cached metadata. For the Hive catalog that имеет not been accessed, Selena will not refresh its cached metadata.
- Introduced in: v1.5.2

enable_statistics_collect_profile

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, generate profiles for statistics queries. You can set this item to `true` to всеow Selena to generate query profiles for queries on system statistics.
- Introduced in: v1.5.2

metadata_enable_recovery_mode

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Определяет, enable the metadata recovery mode. When this mode включён, if part of the cluster metadata is lost, it может быть restored based on the information from BE. Currently, only the version information of partitions может быть restored.
- Introduced in: v1.5.2

black_host_history_sec

- Default: 2 * 60
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: The time duration for retaining historical connection failures of узел BEs in the BE Blacklist. If a узел BE is added to the BE Blacklist automaticвсey, Selena will assess its

connectivity and judge whether it может быть removed from the BE Blacklist. Within `black_host_history_sec`, only if a blacklisted узел BE имеет fewer connection failures than the threshold set in `black_host_connect_failures_within_time`, it может быть removed from the BE Blacklist.

- Introduced in: v1.5.2

black_host_connect_failures_within_time

- Default: 5
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Порог of connection failures всеowed for a blacklisted узел BE. If a узел BE is added to the BE Blacklist automaticвсey, Selena will assess its connectivity and judge whether it может быть removed from the BE Blacklist. Within `black_host_history_sec`, only if a blacklisted узел BE имеет fewer connection failures than the threshold set in `black_host_connect_failures_within_time`, it может быть removed from the BE Blacklist.
- Introduced in: v1.5.2

lock_manager_enabled

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Определяет, enable the lock manager. The lock manager выполняет central management for locks. For example, it can control whether to refine the granularity of metadata locks from the database level to the table level.
- Introduced in: v1.5.2

lock_manager_enable_using_fine_granularity_lock

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Определяет, refine the granularity of metadata locks from the database level to the table level. After metadata locks are refined to the table level, lock conflicts and

contentions может быть reduced, which improves load and query concurrency. Этот параметр only вступает в силу when `lock_manager_enabled` включён.
- Introduced in: v1.5.2

enable_legacy_compatibility_for_replication

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable the Legacy Compatibility for Replication. Selena may beиметь differently between the old and new versions, causing problems during cross-cluster data migration. Therefore, you must enable Legacy Compatibility for the target cluster before data migration and disable it after data migration is завершён. `true` означает enabling this mode.
- Introduced in: v1.5.2

automated_cluster_snapshot_interval_seconds

- Default: 600
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: The interval с которым the Automated Cluster Snapshot tasks срабатывают.
- Introduced in: v1.5.2

enable_table_name_case_insensitive

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Определяет, enable case-insensitive processing on catalog names, database names, table names, view names, and materialized view names. Currently, table names are case-sensitive по умолчанию.
    - After enabling this feature, все related names будет stored in lowercase, and все SQL commands containing these names will automaticвсey convert them to lowercase.
    - You can enable this feature только когда creating a cluster. **After the cluster is started, the value of this configuration cannot be modified by любой means**. Any attempt для изменения it will result in an error. FE will fail to start when it detects that

the value of this параметр конфигурации is inconsistent with that when the cluster was first started.
  - Currently, this feature does not support JDBC catalog and table names. Do not enable this feature if you want to наform case-insensitive processing on JDBC or ODBC data sources.
- Introduced in: v2.0.0

txn_latency_metric_report_groups

- Default: An empty string
- Type: String
- Unit: -
- Is mutable: Yes
- Description: A comma-separated list of transaction latency metric groups to report. Load types are categorized into logical groups for monitoring. When a group включён, its name is added as a 'type' label to transaction metrics. Допустимые значения: `stream_load`, `routine_load`, `broker_load`, `insert`, and `compaction` (availabl only for shared-data clusters). Example: `"stream_load,routine_load"`.
- Introduced in: v2.0.0

User, role, and privilege

privilege_max_total_roles_на_user

- Default: 64
- Type: Int
- Unit:
- Is mutable: Yes
- Description: Максимальное количество roles a user can have.
- Introduced in: v1.5.2

privilege_max_role_depth

- Default: 16
- Type: Int
- Unit:
- Is mutable: Yes
- Description: Максимум role depth (level of inheritance) of a role.
- Introduced in: v1.5.2

Query engine

publish_version_interval_ms

- Default: 10
- Type: Int
- Unit: Milliseconds
- Is mutable: No
- Description: Интервал времени с которым release validation tasks are issued.
- Introduced in: -

statistic_cache_columns

- Default: 100000
- Type: Long
- Unit: -
- Is mutable: No
- Description: Количество rows that может быть cached for the statistics table.
- Introduced in: -

statistic_cache_thread_pool_size

- Default: 10
- Type: Int
- Unit: -
- Is mutable: No
- Description: The size of the thread-pool which будет используется для refresh statistic caches.
- Introduced in: -

max_всеowed_in_element_num_of_delete

- Default: 10000
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество elements всеowed for the IN predicate in a DELETE statement.
- Introduced in: -

enable_materialized_view

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable the creation of materialized views.
- Introduced in: -

enable_materialized_view_spill

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable Intermediate Result Spilling for materialized view refresh tasks.
- Introduced in: v1.5.2

enable_backup_materialized_view

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable the BACKUP and RESTORE of asynchronous materialized views when backing up or restoring a specific database. If this item установлен в `false`, Selena will skip backing up asynchronous materialized views.
- Introduced in: v1.5.2

enable_exнaimental_mv

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable the asynchronous materialized view feature. TRUE означает this feature включён. From v1.5.2 onwards, this feature включён по умолчанию. For versions earlier than v1.5.2, this feature отключён по умолчанию.
- Introduced in: v1.5.2

enable_colocate_mv_index

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, support colocating the synchronous materialized view index with the base table when creating a synchronous materialized view. If this item установлен в `true`, tablet sink will speed up the write наformance of synchronous materialized views.
- Introduced in: v1.5.2

default_mv_refresh_immediate

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, refresh an asynchronous materialized view immediately after creation. When this item установлен в `true`, newly created materialized view будет refreshed immediately.
- Introduced in: v1.5.2

enable_materialized_view_metrics_collect

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, collect monitoring metrics for asynchronous materialized views по умолчанию.
- Introduced in: v1.5.2

enable_materialized_view_text_based_rewrite

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable text-based query rewrite по умолчанию. If this item установлен в `true`, the system builds the abstract syntax tree while creating an asynchronous materialized view.

- Introduced in: v1.5.2

enable_mv_automatic_active_check

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable the system to automaticвсey check and re-activate the asynchronous materialized views that are set inactive because their base tables (views) had undergone Schema Change or had been dropped and re-created. Please note that this feature will not re-activate the materialized views that are manuвсey set inactive by users.
- Introduced in: v1.5.2

enable_active_materialized_view_schema_strict_check

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, strictly check the length consistency of тип данныхs when activating an inactive materialized view. When this item установлен в `false`, the activation of the materialized view is not affected if the length of the тип данныхs имеет changed in the base table.
- Introduced in: v1.5.2

mv_active_checker_interval_seconds

- Default: 60
- Type: Long
- Unit: Seconds
- Is mutable: Yes
- Description: When the background active_checker thread включён, the system will наiodicвсey detect and automaticвсey reactivate materialized views that became Inactive due to schema changes or rebuilds of their base tables (or views). Этот параметр controls the scheduling interval of the checker thread, в секундах. Значение по умолчанию — system-defined.
- Introduced in: v1.5.2

default_mv_partition_refresh_number

- Default: 1
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: When a materialized view refresh involves multiple partitions, this parameter controls how млюбой partitions are refreshed in a single batch по умолчанию. Starting начиная с версии version 3.3.0, the system defaults to refreshing one partition at a time to avoid potential out-of-memory (OOM) issues. In earlier versions, все partitions were refreshed at once по умолчанию, which could lead to memory exhaustion and task failure. However, note that when a materialized view refresh involves a large number of partitions, refreshing only one partition at a time may lead to excessive scheduling overhead, longer overвсе refresh time, and a large number of refresh records. In such cases, it рекомендуется to adjust this parameter appropriately to improve refresh efficiency and reduce scheduling costs.
- Introduced in: v1.5.2

enable_udf

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Определяет, enable UDF.
- Introduced in: -

enable_decimal_v3

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, support the DECIMAL V3 тип данных.
- Introduced in: -

enable_sql_blacklist

- Default: false
- Type: Boolean
- Unit: -

- Is mutable: Yes
- Description: Определяет, enable blacklist check for SQL queries. When this feature включён, queries in the blacklist cannot be executed.
- Introduced in: -

dynamic_partition_enable

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable the dynamic partitioning feature. When this feature включён, Selena dynamicвсey creates partitions for new data and automaticвсey deletes expired partitions to ensure the freshness of data.
- Introduced in: -

dynamic_partition_check_interval_seconds

- Default: 600
- Type: Long
- Unit: Seconds
- Is mutable: Yes
- Description: The interval с которым new data is checked. If new data is detected, Selena automaticвсey creates partitions for the data.
- Introduced in: -

max_query_retry_time

- Default: 2
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество query retries on an FE.
- Introduced in: -

max_create_table_timeout_second

- Default: 600
- Type: Int
- Unit: Seconds
- Is mutable: Yes

- Description: Максимум timeout duration for creating a table.
- Introduced in: -

create_table_max_serial_replicas

- Default: 128
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество replicas to create seriвсey. If actual replica count exceeds this value, replicas будет created concurrently. Try to reduce this value if table creation is taking a long time to complete.
- Introduced in: -

http_slow_request_threshold_ms

- Default: 5000
- Type: Int
- Unit: Milliseconds
- Is mutable: Yes
- Description: If the время ответа for an HTTP-запрос exceeds the value указанного параметром this parameter, a log is generated to track this request.
- Introduced in: v1.5.2

max_partitions_in_one_batch

- Default: 4096
- Type: Long
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество partitions that может быть created when you bulk create partitions.
- Introduced in: -

max_running_rollup_job_num_на_table

- Default: 1
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество rollup jobs can run in parвceel for a table.

- Introduced in: -

expr_children_limit

- Default: 10000
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество child expressions всеowed in an expression.
- Introduced in: -

max_planner_scalar_rewrite_num

- Default: 100000
- Type: Long
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество times that the optimizer can rewrite a scalar онаator.
- Introduced in: -

max_scalar_онаator_optimize_depth

- Default：256
- Type：Int
- Unit：-
- Is mutable: Yes
- Description: Максимум depth that ScalarОнаator optimization может быть applied.
- Introduced in: -

max_scalar_онаator_flat_children

- Default：10000
- Type：Int
- Unit：-
- Is mutable: Yes
- Description：Максимальное количество flat children for ScalarОнаator. You can set this limit to prevent the optimizer from using too much memory.
- Introduced in: -

enable_statistic_collect

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, collect statistics for the CBO. This feature включён по умолчанию.
- Introduced in: -

enable_statistic_collect_on_first_load

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Controls automatic statistics collection and maintenance triggered by data loading онаations. This includes:
  - Statistics collection when data is first loaded into a partition (partition version equals 2).
  - Statistics collection when data is loaded into empty partitions of multi-partition tables.
  - Statistics copying and updating for INSERT OVERWRITE онаations.

  Decision Policy for Statistics Collection Type:
  - For INSERT OVERWRITE: `deltaRatio = |targetRows - sourceRows| / (sourceRows + 1)`
    - If `deltaRatio < statistic_sample_collect_ratio_threshold_of_first_load` (Default: 0.1), statistics collection will not be haformed. Only the existing statistics будет copied.
    - Else, if `targetRows > statistic_sample_collect_rows` (Default: 200000), SAMPLE statistics collection используется.
    - Else, FULL statistics collection используется.
  - For First Load: `deltaRatio = loadRows / (totalRows + 1)`

    - If `deltaRatio < statistic_sample_collect_ratio_threshold_of_first_load` (Default: 0.1), statistics collection will not be наformed.
    - Else, if `loadRows > statistic_sample_collect_rows` (Default: 200000), SAMPLE statistics collection используется.
    - Else, FULL statistics collection используется.

  Synchronization Behavior:

  - For DML statements (INSERT INTO/INSERT OVERWRITE): Synchronous mode with table lock. The load онаation waits for statistics collection to complete (up to `semi_sync_collect_statistic_await_seconds`).
  - For Stream Load and Broker Load: Asynchronous mode without lock. Statistics collection runs in background without blocking the load онаation.

  > **ПРИМЕЧАНИЕ**
  >
  > Disabling this configuration will prevent все loading-triggered statistics онаations, including statistics maintenance for INSERT OVERWRITE, which may result in tables lacking statistics. If new tables are frequently created and data is frequently loaded, enabling this feature will increase memory and CPU overhead.

- Introduced in: v1.5.2

enable_statistic_collect_on_update

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Controls whether UPDATE statements can trigger automatic statistics collection. When enabled, UPDATE онаations that modify table data may schedule statistics collection through the same ingestion-based statistics framework controlled by `enable_statistic_collect_on_first_load`. Disabling this configuration skips statistics collection for UPDATE statements while keeping load-triggered statistics collection behavior unchanged.
- Introduced in: v1.5.2, v2.0.0

semi_sync_collect_statistic_await_seconds

- Default: 30

- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Maximum wait time for semi-synchronous statistics collection during DML онаations (INSERT INTO and INSERT OVERWRITE statements). Stream Load and Broker Load use asynchronous mode and are not affected by this configuration. If statistics collection time exceeds this value, the load онаation continues without waiting for collection to complete. This configuration works in conjunction with `enable_statistic_collect_on_first_load`.
- Introduced in: v1.5.2

statistic_auto_analyze_start_time

- Default: 00:00:00
- Type: String
- Unit: -
- Is mutable: Yes
- Description: The start time of automatic collection. Value range: `00:00:00` - `23:59:59`.
- Introduced in: -

statistic_auto_analyze_end_time

- Default: 23:59:59
- Type: String
- Unit: -
- Is mutable: Yes
- Description: The end time of automatic collection. Value range: `00:00:00` - `23:59:59`.
- Introduced in: -

statistic_analyze_status_keep_second

- Default: 3 * 24 * 3600
- Type: Long
- Unit: Seconds
- Is mutable: Yes
- Description: The duration to retain the history of collection tasks. Значение по умолчанию — 3 days.
- Introduced in: -

statistic_collect_interval_sec

- Default: 5 * 60
- Type: Long
- Unit: Seconds
- Is mutable: Yes
- Description: The interval for checking data updates during automatic collection.
- Introduced in: -

statistic_update_interval_sec

- Default: 24 * 60 * 60
- Type: Long
- Unit: Seconds
- Is mutable: Yes
- Description: The interval с которым the cache of statistical information is updated.
- Introduced in: -

enable_collect_full_statistic

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable automatic full statistics collection. This feature включён по умолчанию.
- Introduced in: -

statistic_auto_collect_ratio

- Default: 0.8
- Type: Double
- Unit: -
- Is mutable: Yes
- Description: Порог for determining whether the statistics for automatic collection are healthy. If statistics health is below этот порог, automatic collection срабатывает.
- Introduced in: -

statistic_auto_collect_smвсе_table_rows

- Default: 10000000
- Type: Long

- Unit: -
- Is mutable: Yes
- Description: Threshold для определения whether a table in an external data source (Hive, Iceberg, Hudi) is a smвce table during automatic collection. If the table имеет rows меньше this value, the table is considered a smвce table.
- Introduced in: v1.5.2

statistic_max_full_collect_data_size

- Default: 100 * 1024 * 1024 * 1024
- Type: Long
- Unit: bytes
- Is mutable: Yes
- Description: The data size threshold for the automatic collection of statistics. If the total size exceeds this value, then sampled collection is наformed instead of full.
- Introduced in: -

statistic_collect_max_row_count_на_query

- Default: 5000000000
- Type: Long
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество rows для запроса for a single analyze task. An analyze task будет split into multiple queries if this value is exceeded.
- Introduced in: -

statistic_sample_collect_rows

- Default: 200000
- Type: Long
- Unit: -
- Is mutable: Yes
- Description: The row count threshold for deciding between SAMPLE and FULL statistics collection during loading-triggered statistics онаations. If the number of loaded or changed rows exceeds этот порог (default 200,000), SAMPLE statistics collection используется; otherwise, FULL statistics collection используется. This настройка works in conjunction with `enable_statistic_collect_on_first_load` and `statistic_sample_collect_ratio_threshold_of_first_load`.
- Introduced in: -

enable_predicate_columns_collection

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable predicate columns collection. If disabled, predicate columns will not be recorded during query optimization.
- Introduced in: -

histogram_buckets_size

- Default: 64
- Type: Long
- Unit: -
- Is mutable: Yes
- Description: The default bucket number for a histogram.
- Introduced in: -

histogram_mcv_size

- Default: 100
- Type: Long
- Unit: -
- Is mutable: Yes
- Description: Количество most common values (MCV) for a histogram.
- Introduced in: -

histogram_sample_ratio

- Default: 0.1
- Type: Double
- Unit: -
- Is mutable: Yes
- Description: The sampling ratio for a histogram.
- Introduced in: -

histogram_max_sample_row_count

- Default: 10000000
- Type: Long
- Unit: -

- Is mutable: Yes
- Description: Максимальное количество rows to collect for a histogram.
- Introduced in: -

connector_table_query_trigger_task_schedule_interval

- Default: 30
- Type: Int
- Unit: Second
- Is mutable: Yes
- Description: The interval с которым the Scheduler thread schedules the query-trigger background tasks. Этот параметр is to replace `connector_table_query_trigger_analyze_schedule_interval` introduced in v1.5.2. Here, the background tasks refer `ANALYZE` tasks in v1.5.2，and the collection task of low-cardinality columns' dictionary in versions later than v1.5.2.
- Introduced in: v1.5.2

connector_table_query_trigger_analyze_smвce_table_rows

- Default: 10000000
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Порог for determining whether a table is a smвce table for query-trigger ANALYZE tasks.
- Introduced in: v1.5.2

connector_table_query_trigger_analyze_smвce_table_interval

- Default: 2 * 3600
- Type: Int
- Unit: Second
- Is mutable: Yes
- Description: The interval for query-trigger ANALYZE tasks of smвce tables.
- Introduced in: v1.5.2

connector_table_query_trigger_analyze_large_table_interval

- Default: 12 * 3600
- Type: Int
- Unit: Second

- Is mutable: Yes
- Description: The interval for query-trigger ANALYZE tasks of large tables.
- Introduced in: v1.5.2

connector_table_query_trigger_analyze_max_ожидающий_task_num

- Default: 100
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Maximum number of query-trigger ANALYZE tasks that are in Pending state on the FE.
- Introduced in: v1.5.2

connector_table_query_trigger_analyze_max_running_task_num

- Default: 2
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Maximum number of query-trigger ANALYZE tasks that are in Running state on the FE.
- Introduced in: v1.5.2

enable_local_replica_selection

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, select local replicas for queries. Local replicas reduce the network transmission cost. Если этот параметр установлен в TRUE, the CBO preferentiвсey selects tablet replicas on BEs that иметь the same IP address as the current FE. Если этот параметр установлен в `FALSE`, both local replicas and non-local replicas может быть selected.
- Introduced in: -

max_distribution_pruner_recursion_depth

- Default: 100
- Type: Int

- Unit: -
- Is mutable: Yes
- Description:: Максимум recursion depth всеowed by the partition pruner. Increasing the recursion depth can prune more elements but also increases CPU consumption.
- Introduced in: -

slow_query_analyze_threshold

- Default: 5
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description:: The execution time threshold for queries to trigger the analysis of Query Feedback.
- Introduced in: v1.5.2

low_cardinality_threshold

- Default: 255
- Type: Int
- Unit: -
- Is mutable: No
- Description: Threshold of low cardinality dictionary.
- Introduced in: v1.5.2

enable_manual_collect_array_ndv

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable manual collection for the NDV information of the ARRAY type.
- Introduced in: v2.0.0

enable_auto_collect_array_ndv

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes

- Description: Определяет, enable automatic collection for the NDV information of the ARRAY type.
- Introduced in: v2.0.0

Loading and unloading

load_straggler_wait_second

- Default: 300
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Максимум loading lag that может быть tolerated by a BE replica. Если это значение is exceeded, cloning is наformed to clone data from other replicas.
- Introduced in: -

load_checker_interval_second

- Default: 5
- Type: Int
- Unit: Seconds
- Is mutable: No
- Description: Интервал времени с которым load jobs are processed on a rolling basis.
- Introduced in: -

broker_load_default_timeout_second

- Default: 14400
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: The timeout duration for a Broker Load job.
- Introduced in: -

min_bytes_на_broker_scanner

- Default: 67108864
- Type: Long
- Unit: Bytes
- Is mutable: Yes

- Description: Минимум всеowed amount of data that может быть processed by a Broker Load instance.
- Introduced in: -

insert_load_default_timeout_second

- Default: 3600
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: The timeout duration for the INSERT INTO statement that is используется для load data.
- Introduced in: -

stream_load_default_timeout_second

- Default: 600
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: The default timeout duration for каждый Stream Load job.
- Introduced in: -

max_stream_load_timeout_second

- Default: 259200
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Максимум всеowed timeout duration for a Stream Load job.
- Introduced in: -

max_load_timeout_second

- Default: 259200
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Максимум timeout duration всеowed for a load job. The load job fails if this limit is exceeded. This limit applies to все types of load jobs.
- Introduced in: -

min_load_timeout_second

- Default: 1
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Минимум timeout duration всеowed for a load job. This limit applies to все types of load jobs.
- Introduced in: -

prepared_transaction_default_timeout_second

- Default: 86400
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: The default timeout duration for a prepared transaction.
- Introduced in: -

finish_transaction_default_lock_timeout_ms

- Default: 1000
- Type: Int
- Unit: MilliSeconds
- Is mutable: Yes
- Description: The default timeout for acquiring the db and table lock during finishing transaction.
- Introduced in: v2.0.0, v1.5.2

spark_dpp_version

- Default: 1.0.0
- Type: String
- Unit: -
- Is mutable: No
- Description: The version of Spark Dynamic Partition Pruning (DPP) used.
- Introduced in: -

spark_load_default_timeout_second

- Default: 86400
- Type: Int

- Unit: Seconds
- Is mutable: Yes
- Description: The timeout duration for каждый Spark Load job.
- Introduced in: -

spark_home_default_dir

- Default: SelenaFE.SELENA_HOME_DIR + "/lib/spark2x"
- Type: String
- Unit: -
- Is mutable: No
- Description: The root directory of a Spark client.
- Introduced in: -

spark_resource_path

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The root directory of the Spark dependency package.
- Introduced in: -

spark_launcher_log_dir

- Default: sys_log_dir + "/spark_launcher_log"
- Type: String
- Unit: -
- Is mutable: No
- Description: Директория которая хранит Spark файл логаs.
- Introduced in: -

yarn_client_path

- Default: SelenaFE.SELENA_HOME_DIR + "/lib/yarn-client/hadoop/bin/yarn"
- Type: String
- Unit: -
- Is mutable: No
- Description: The root directory of the Yarn client package.
- Introduced in: -

yarn_config_dir

- Default: SelenaFE.SELENA_HOME_DIR + "/lib/yarn-config"
- Type: String
- Unit: -
- Is mutable: No
- Description: Директория которая хранит the Yarn configuration file.
- Introduced in: -

desired_max_waiting_jobs

- Default: 1024
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество ожидающий jobs in an FE. The number refers to все jobs, such as table creation, loading, and schema change jobs. If the number of ожидающий jobs in an FE гкаждыйes this value, the FE will reject new load requests. Этот параметр вступает в силу only for asynchronous loading. From v1.5.2 onwards, the default value is changed from 100 to 1024.
- Introduced in: -

max_running_txn_num_на_db

- Default: 1000
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество load transactions всеowed to be running for каждый database within a Selena cluster. Значение по умолчанию — `1000`. From v1.5.2 onwards, the default value изменён на `1000` from `100`. When the actual number of load transactions running for a database exceeds the value of this parameter, new load requests will not be processed. New requests for synchronous load jobs будет denied, and new requests for asynchronous load jobs будет placed in queue. We do not recommend you increase the value of this parameter because this will increase system load.
- Introduced in: -

max_broker_load_job_concurrency

- Default: 5
- Alias: async_load_task_pool_size

- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество concurrent Broker Load jobs всеowed within the Selena cluster. Этот параметр is valid only for Broker Load. The value of this parameter должен быть меньше the value of `max_running_txn_num_на_db`. From v1.5.2 onwards, the default value is changed from `10` to `5`.
- Introduced in: -

load_parвceel_instance_num (Deprecated)

- Default: 1
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество concurrent loading instances for каждый load job on a BE. Этот параметр is deprecated начиная с версии v1.5.2 onwards.
- Introduced in: -

disable_load_job

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, disable loading when the cluster encounters an error. This prevents любой loss caused by cluster errors. Значение по умолчанию — `FALSE`, indicating that loading is not disabled. `TRUE` означает loading отключён and the cluster is in read-only state.
- Introduced in: -

history_job_keep_max_second

- Default: 7 * 24 * 3600
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Максимум duration a historical job может сохраняться, such as schema change jobs.
- Introduced in: -

label_keep_max_second

- Default: 3 * 24 * 3600
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Максимум duration в секундах to keep the labels of load jobs that иметь been завершён and are in the FINISHED or CANCELLED state. Значение по умолчанию — 3 days. After this duration expires, the labels будет deleted. Этот параметр applies to все types of load jobs. A value too large consumes a lot of memory.
- Introduced in: -

label_keep_max_num

- Default: 1000
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество load jobs that может сохраняться within a haiod of time. If this number is exceeded, the information of historical jobs будет deleted.
- Introduced in: -

max_routine_load_task_concurrent_num

- Default: 5
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество concurrent tasks for каждый Routine Load job.
- Introduced in: -

max_routine_load_task_num_на_be

- Default: 16
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество concurrent Routine Load tasks on каждый BE. Since v1.5.2, the default value for this parameter is increased to 16 from 5, and no longer needs to be меньше or равно the value of BE static parameter `routine_load_thread_pool_size` (deprecated).

- Introduced in: -

max_routine_load_batch_size

- Default: 4294967296
- Type: Long
- Unit: Bytes
- Is mutable: Yes
- Description: Максимум amount of data that может быть loaded by a Routine Load task.
- Introduced in: -

routine_load_task_consume_second

- Default: 15
- Type: Long
- Unit: Seconds
- Is mutable: Yes
- Description: Максимум time for каждый Routine Load task within the cluster to consume data. Since v1.5.2, Routine Load job supports a new parameter `task_consume_second` in job_pronaties. Этот параметр applies to individual load tasks within a Routine Load job, which is more flexible.
- Introduced in: -

routine_load_task_timeout_second

- Default: 60
- Type: Long
- Unit: Seconds
- Is mutable: Yes
- Description: The timeout duration for каждый Routine Load task within the cluster. Since v1.5.2, Routine Load job supports a new parameter `task_timeout_second` in job_pronaties. Этот параметр applies to individual load tasks within a Routine Load job, which is more flexible.
- Introduced in: -

routine_load_unstable_threshold_second

- Default: 3600
- Type: Long
- Unit: Seconds
- Is mutable: Yes

- Description: Routine Load job установлен в the UNSTABLE state if любой task within the Routine Load job lags. To be specific, the difference between the timestamp of the message being consumed and the current time exceeds этот порог, and unconsumed messages exist in the data source.
- Introduced in: -

enable_routine_load_lag_metrics

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, collect Routine Load Kafka partition offset lag metrics. Please note that set this item to `true` will свсе the Kafka API to get the partition's latest offset.
- Introduced in: -

min_routine_load_lag_for_metrics

- Default: 10000
- Type: INT
- Unit: -
- Is mutable: Yes
- Description: Минимум offset lag of Routine Load jobs to be shown in monitoring metrics. Routine Load jobs whose offset lags are больше this value будет displayed in the metrics.
- Introduced in: -

max_tolerable_backend_down_num

- Default: 0
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество faulty узел BEs всеowed. If this number is exceeded, Routine Load jobs cannot be automaticвсey recovered.
- Introduced in: -

наiod_of_auto_resume_min

- Default: 5
- Type: Int
- Unit: Minutes

- Is mutable: Yes
- Description: The interval с которым Routine Load jobs are automaticвсey recovered.
- Introduced in: -

export_task_default_timeout_second

- Default: 2 * 3600
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: The timeout duration for a data exporting task.
- Introduced in: -

export_max_bytes_на_be_на_task

- Default: 268435456
- Type: Long
- Unit: Bytes
- Is mutable: Yes
- Description: Максимум amount of data that может быть exported from a single BE by a single data unload task.
- Introduced in: -

export_task_pool_size

- Default: 5
- Type: Int
- Unit: -
- Is mutable: No
- Description: The size of the unload task thread pool.
- Introduced in: -

export_checker_interval_second

- Default: 5
- Type: Int
- Unit: Seconds
- Is mutable: No
- Description: Интервал времени с которым load jobs are scheduled.
- Introduced in: -

export_running_job_num_limit

- Default: 5
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество data exporting tasks that can run in parвceel.
- Introduced in: -

empty_load_as_error

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, return an error message "все partitions иметь no load data" if no data is loaded. Valid values:
  - `true`: If no data is loaded, the system displays a failure message and returns an error "все partitions иметь no load data".
  - `false`: If no data is loaded, the system displays a success message and returns OK, instead of an error.
- Introduced in: -

external_table_commit_timeout_ms

- Default: 10000
- Type: Int
- Unit: Milliseconds
- Is mutable: Yes
- Description: The timeout duration for committing (publishing) a write transaction to a Selena external table. The default value `10000` означает a 10-second timeout duration.
- Introduced in: -

enable_sync_publish

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, synchronously execute the apply task at the publish phase of a load transaction. Этот параметр is applicable only to Primary Key tables. Valid values:

  - `TRUE` (default): The apply task is synchronously executed at the publish phase of a load transaction. It означает, что the load transaction is reported as successful only after the apply task is завершён, and the loaded data can truly be queried. When a task loads a large volume of data at a time or loads data frequently, настройка this parameter to `true` can improve query наformance and stability, but may increase load latency.
  - `FALSE`: The apply task is asynchronously executed at the publish phase of a load transaction. It означает, что the load transaction is reported as successful after the apply task is submitted, but the loaded data cannot be immediately queried. In this case, concurrent queries need to wait for the apply task to complete or time out before they can continue. When a task loads a large volume of data at a time or loads data frequently, настройка this parameter to `false` may affect query наformance and stability.
- Introduced in: v1.5.2

label_clean_interval_second

- Default: 4 * 3600
- Type: Int
- Unit: Seconds
- Is mutable: No
- Description: Интервал времени с которым labels are cleaned up. Unit: second. We recommend that you specify a short time interval to ensure that historical labels может быть cleaned up in a timely manner.
- Introduced in: -

transaction_clean_interval_second

- Default: 30
- Type: Int
- Unit: Seconds
- Is mutable: No
- Description: Интервал времени с которым finished transactions are cleaned up. Unit: second. We recommend that you specify a short time interval to ensure that finished transactions может быть cleaned up in a timely manner.
- Introduced in: -

transaction_stream_load_coordinator_cache_capacity

- Default: 4096

- Type: Int
- Unit: -
- Is mutable: Yes
- Description: The capacity of the cache которая хранит the mapping from transaction label to coordinator node.
- Introduced in: -

transaction_stream_load_coordinator_cache_expire_seconds

- Default: 900
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: The time to keep the coordinator mapping in the cache before it's evicted(TTL).
- Introduced in: -

enable_file_bundling

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable the File Bundling optimization for the cloud-native table. When this feature включён (set to `true`), the system automaticвсеу bundles the файл данныхs generated by loading, Compaction, or Publish онаations, thereby reducing the API cost caused by high-frequency access to the external storage system. You can also control this behavior on the table level using the CREATE TABLE proнaty `file_bundling`. For detailed instructions, see CREATE TABLE.
- Introduced in: v2.0.0

Storage

default_replication_num

- Default: 3
- Type: Short
- Unit: -
- Is mutable: Yes
- Description: Sets the default number of replicas for каждый data partition when creating a table in Selena. This настройка может быть overridden when creating a table by

specifying `replication_num=x` in the CREATE TABLE DDL.
- Introduced in: -

enable_strict_storage_medium_check

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, the FE strictly checks the storage medium of BEs when users create tables. Если этот параметр установлен в `TRUE`, the FE checks the storage medium of BEs when users create tables and returns an error if the storage medium of the BE is different from the `storage_medium` parameter specified in the CREATE TABLE statement. For example, the storage medium specified in the CREATE TABLE statement is SSD but the actual storage medium of BEs is HDD. As a result, the table creation fails. Если этот параметр is `FALSE`, the FE does not check the storage medium of BEs when users create a table.
- Introduced in: -

catalog_trash_expire_second

- Default: 86400
- Type: Long
- Unit: Seconds
- Is mutable: Yes
- Description: The longest duration the metadata может сохраняться after a database, table, or partition is dropped. If this duration expires, the data будет deleted and cannot be recovered through the RECOVER command.
- Introduced in: -

partition_recycle_retention_наiod_secs

- Default: 1800
- Type: Long
- Unit: Seconds
- Is mutable: Yes
- Description: The metadata retention time for the partition that is dropped by INSERT OVERWRITE or materialized view refresh онаations. Note that such metadata cannot be recovered by executing RECOVER.
- Introduced in: v1.5.2

enable_auto_tablet_distribution

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, automaticвсey set the number of buckets.
  - Если этот параметр установлен в `TRUE`, you don't need чтобы указать the number of buckets when you create a table or add a partition. Selena automaticвсey determines the number of buckets.
  - Если этот параметр установлен в `FALSE`, you need to manuвсey specify the number of buckets when you create a table or add a partition. If you do not specify the bucket count when adding a new partition to a table, the new partition inherits the bucket count set at the creation of the table. However, you can also manuвсey specify the number of buckets for the new partition.
- Introduced in: v1.5.2

enable_exнaimental_rowstore

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable the hybrid row-column storage feature.
- Introduced in: v1.5.2

enable_exнaimental_gin

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable the full-text inverted index feature.
- Introduced in: v1.5.2

storage_usage_soft_limit_нacent

- Default: 90
- Alias: storage_high_watermark_usage_нacent
- Type: Int
- Unit: -

- Is mutable: Yes
- Description: Soft limit of the storage usage нacentage in a BE directory. If the storage usage (in нacentage) of the BE storage directory exceeds this value and the remaining storage space is меньше `storage_usage_soft_limit_reserve_bytes`, tablets cannot be cloned into this directory.
- Introduced in: -

storage_usage_soft_limit_reserve_bytes

- Default: 200 * 1024 * 1024 * 1024
- Alias: storage_min_left_capacity_bytes
- Type: Long
- Unit: Bytes
- Is mutable: Yes
- Description: Soft limit of the remaining storage space in a BE directory. If the remaining storage space in the BE storage directory is меньше this value and the storage usage (in нacentage) exceeds `storage_usage_soft_limit_нacent`, tablets cannot be cloned into this directory.
- Introduced in: -

storage_usage_hard_limit_нacent

- Default: 95
- Alias: storage_flood_stage_usage_нacent
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Hard limit of the storage usage нacentage in a BE directory. If the storage usage (in нacentage) of the BE storage directory exceeds this value and the remaining storage space is меньше `storage_usage_hard_limit_reserve_bytes`, Load and Restore jobs are rejected. You need to set this item together with the BE параметр конфигурации `storage_flood_stage_usage_нacent` to всеow the configurations to take effect.
- Introduced in: -

storage_usage_hard_limit_reserve_bytes

- Default: 100 * 1024 * 1024 * 1024
- Alias: storage_flood_stage_left_capacity_bytes
- Type: Long

- Unit: Bytes
- Is mutable: Yes
- Description: Hard limit of the remaining storage space in a BE directory. If the remaining storage space in the BE storage directory is меньше this value and the storage usage (in нacentage) exceeds `storage_usage_hard_limit_нacent`, Load and Restore jobs are rejected. You need to set this item together with the BE параметр конфигурации `storage_flood_stage_left_capacity_bytes` to всеow the configurations to take effect.
- Introduced in: -

alter_table_timeout_second

- Default: 86400
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: The timeout duration for the schema change онаation (ALTER TABLE).
- Introduced in: -

enable_fast_schema_evolution

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable fast schema evolution for все tables within the Selena cluster. Допустимые значения: `TRUE` and `FALSE` (default). Enabling fast schema evolution can increase the speed of schema changes and reduce resource usage when columns are added or dropped.
- Introduced in: v1.5.2

> **NOTE**
>
> - Selena shared-data clusters supports this parameter начиная с версии v1.5.2.
> - If you need для настройки the fast schema evolution for a specific table, such as disabling fast schema evolution for a specific table, you can set the table pronaty `fast_schema_evolution` at table creation.

recover_with_empty_tablet

- Default: false

- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, replace a lost or corrupted tablet replica with an empty one. If a tablet replica is lost or corrupted, data queries on this tablet or other healthy tablets may fail. Replacing the lost or corrupted tablet replica with an empty tablet гарантирует, что the query can still be executed. However, the result may be incorrect because data is lost. Значение по умолчанию — `FALSE`, which means lost or corrupted tablet replicas are not replaced with empty ones, and the query fails.
- Introduced in: -

tablet_checker_lock_time_на_cycle_ms

- Default: 1000
- Type: Int
- Unit: Milliseconds
- Is mutable: Yes
- Description: Максимум lock hold time на cycle for tablet checker before releasing and reacquiring the table lock. Values меньше 100 будет treated as 100.
- Introduced in: v1.5.2, v2.0.0

tablet_create_timeout_second

- Default: 10
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: The timeout duration for creating a tablet. Значение по умолчанию — changed from 1 to 10 начиная с версии v1.5.2 onwards.
- Introduced in: -

tablet_delete_timeout_second

- Default: 2
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: The timeout duration for deleting a tablet.
- Introduced in: -

check_consistency_default_timeout_second

- Default: 600
- Type: Long
- Unit: Seconds
- Is mutable: Yes
- Description: The timeout duration for a replica consistency check. You can set this parameter based on the size of your tablet.
- Introduced in: -

tablet_sched_slot_num_на_path

- Default: 8
- Alias: schedule_slot_num_на_path
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество tablet-related tasks that can run concurrently in a BE storage directory. From v1.5.2 onwards, the default value of this parameter is changed from `4` to `8`.
- Introduced in: -

tablet_sched_max_scheduling_tablets

- Default: 10000
- Alias: max_scheduling_tablets
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество tablets that может быть scheduled at the same time. Если значение is exceeded, tablet balancing and repair checks будет skipped.
- Introduced in: -

tablet_sched_disable_balance

- Default: false
- Alias: disable_balance
- Type: Boolean
- Unit: -
- Is mutable: Yes

- Description: Определяет, disable tablet balancing. `TRUE` означает that tablet balancing отключён. `FALSE` означает that tablet balancing включён.
- Introduced in: -

tablet_sched_disable_colocate_balance

- Default: false
- Alias: disable_colocate_balance
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, disable replica balancing for Colocate Table. `TRUE` означает replica balancing отключён. `FALSE` означает replica balancing включён.
- Introduced in: -

tablet_sched_be_down_tolerate_time_s

- Default: 900
- Type: Long
- Unit: Seconds
- Is mutable: Yes
- Description: Максимум duration the scheduler всеows for a узел BE to remain inactive. After the time threshold is rкаждыйed, tablets on that узел BE будет migrated to other active узел BEs.
- Introduced in: v1.5.2

tablet_sched_max_balancing_tablets

- Default: 500
- Alias: max_balancing_tablets
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество tablets that может быть balanced at the same time. Если это значение is exceeded, tablet re-balancing будет skipped.
- Introduced in: -

tablet_sched_storage_cooldown_second

- Default: -1
- Alias: storage_cooldown_second

- Type: Long
- Unit: Seconds
- Is mutable: Yes
- Description: The latency of automatic cooling starting from the time of table creation. The default value `-1` указывает, что automatic cooling отключён. If you want чтобы включить automatic cooling, set this parameter to a value больше `-1`.
- Introduced in: -

tablet_sched_max_not_being_scheduled_interval_ms

- Default: 15 * 60 * 1000
- Type: Long
- Unit: Milliseconds
- Is mutable: Yes
- Description: When the tablet clone tasks are being scheduled, if a tablet имеет not been scheduled for the specified time in this parameter, Selena gives it a higher priority to schedule it as soon as possible.
- Introduced in: -

tablet_sched_balance_load_score_threshold

- Default: 0.1
- Alias: balance_load_score_threshold
- Type: Double
- Unit: -
- Is mutable: Yes
- Description: The нacentage threshold for determining whether the load of a BE is balanced. If a BE имеет a lower load than the average load of все BEs and the difference is больше this value, this BE is in a low load state. On the contrary, if a BE имеет a higher load than the average load and the difference is больше this value, this BE is in a high load state.
- Introduced in: -

tablet_sched_num_based_balance_threshold_ratio

- Default: 0.5
- Alias: -
- Type: Double
- Unit: -
- Is mutable: Yes

- Description: Doing num based balance may break the disk size balance, but the maximum gap between disks cannot exceed tablet_sched_num_based_balance_threshold_ratio * tablet_sched_balance_load_score_threshold. If there are tablets in the cluster that are constantly balancing from A to B and B to A, reduce this value. If you want the tablet distribution to be more balanced, increase this value.
- Introduced in: - 3.1

tablet_sched_balance_load_disk_safe_threshold

- Default: 0.5
- Alias: balance_load_disk_safe_threshold
- Type: Double
- Unit: -
- Is mutable: Yes
- Description: The нacentage threshold for determining whether the disk usage of BEs is balanced. If the disk usage of все BEs is ниже this value, it is considered balanced. If the disk usage is больше this value and the difference between the highest and lowest BE disk usage is больше 10%, the disk usage is considered unbalanced and a tablet re-balancing срабатывает.
- Introduced in: -

tablet_sched_repair_delay_factor_second

- Default: 60
- Alias: tablet_repair_delay_factor_second
- Type: Long
- Unit: Seconds
- Is mutable: Yes
- Description: The interval с которым replicas are repaired, в секундах.
- Introduced in: -

tablet_sched_min_clone_task_timeout_sec

- Default: 3 * 60
- Alias: min_clone_task_timeout_sec
- Type: Long
- Unit: Seconds
- Is mutable: Yes
- Description: Минимум timeout duration for cloning a tablet.
- Introduced in: -

tablet_sched_max_clone_task_timeout_sec

- Default: 2 * 60 * 60
- Alias: max_clone_task_timeout_sec
- Type: Long
- Unit: Seconds
- Is mutable: Yes
- Description:Максимум timeout duration for cloning a tablet.
- Introduced in: -

tablet_stat_update_interval_second

- Default: 300
- Type: Int
- Unit: Seconds
- Is mutable: No
- Description: Интервал времени с которым the FE retrieves tablet statistics from каждый BE.
- Introduced in: -

max_automatic_partition_number

- Default: 4096
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество automaticвсey created partitions.
- Introduced in: v1.5.2

auto_partition_max_creation_number_на_load

- Default: 4096
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество partitions может быть created in a table (with Expression Partitioning strategy) by a loading task.
- Introduced in: v1.5.2

max_partition_number_на_table

- Default: 100000

- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество partitions может быть created in a table.
- Introduced in: v1.5.2

max_bucket_number_на_partition

- Default: 1024
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество buckets может быть created in a partition.
- Introduced in: v1.5.2

max_column_number_на_table

- Default: 10000
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество columns может быть created in a table.
- Introduced in: v1.5.2

Shared-data

run_mode

- Default: shared_nothing
- Type: String
- Unit: -
- Is mutable: No
- Description: The running mode of the Selena cluster. Допустимые значения: `shared_data` and `shared_nothing` (Default).
  - `shared_data` означает running Selena in shared-data mode.
  - `shared_nothing` означает running Selena in shared-nothing mode.

  > **CAUTION**

> - You cannot adopt the `shared_data` and `shared_nothing` modes simultaneously for a Selena cluster. Mixed deployment is not supported.
> - DO NOT change `run_mode` after the cluster is deployed. Otherwise, the cluster fails to restart. The transformation from a shared-nothing cluster to a shared-data cluster or vice versa is not supported.

- Introduced in: -

cloud_native_meta_port

- Default: 6090
- Type: Int
- Unit: -
- Is mutable: No
- Description: FE cloud-native metadata server RPC listen port.
- Introduced in: -

enable_load_volume_from_conf

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Определяет, всеow Selena to create the built-in storage volume by using the object storage-related pronaties specified in the FE configuration file. Значение по умолчанию — changed from `true` to `false` начиная с версии v1.5.2 onwards.
- Introduced in: v1.5.2

cloud_native_storage_type

- Default: S3
- Type: String
- Unit: -
- Is mutable: No
- Description: The type of object storage you use. In shared-data mode, Selena supports storing data in HDFS, Azure Blob (supported начиная с версии v1.5.2 onwards), Azure Data Lake Storage Gen2 (supported начиная с версии v1.5.2 onwards), Google Storage (with native SDK, supported начиная с версии v1.5.2 onwards), and object storage systems that are compatible with the S3 protocol (such as AWS S3, and MinIO). Valid value: `S3` (Default), `HDFS`, `AZBLOB`, `ADLS2`, and `GS`. If you specify this parameter as `S3`,

you must add the parameters prefixed by `aws_s3`. If you specify this parameter as `AZBLOB`, you must add the parameters prefixed by `azure_blob`. If you specify this parameter as `ADLS2`, you must add the parameters prefixed by `azure_adls2`. If you specify this parameter as `GS`, you must add the parameters prefixed by `gcp_gcs`. If you specify this parameter as `HDFS`, you only need чтобы указать `cloud_native_hdfs_url`.
- Introduced in: -

cloud_native_hdfs_url

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The URL of the HDFS storage, for example, `hdfs://127.0.0.1:9000/user/xxx/selena/`.
- Introduced in: -

aws_s3_path

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The S3 path используется для store data. It consists of the name of your S3 bucket and the sub-path (if любой) under it, for example, `testbucket/subpath`.
- Introduced in: v1.5.2

aws_s3_region

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The region in which your S3 bucket resides, for example, `us-west-2`.
- Introduced in: v1.5.2

aws_s3_endpoint

- Default: Empty string
- Type: String
- Unit: -

- Is mutable: No
- Description: The endpoint используется для access your S3 bucket, for example, `https://s3.us-west-2.amazonaws.com`.
- Introduced in: v1.5.2

aws_s3_use_aws_sdk_default_behavior

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Определяет, use the default authentication credential of AWS SDK. Допустимые значения: true and false (Default).
- Introduced in: v1.5.2

aws_s3_use_instance_profile

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Определяет, use Instance Profile and Assumed Role as credential methods for accessing S3. Допустимые значения: true and false (Default).
  - If you use IAM user-based credential (Access Key and Secret Key) to access S3, you must specify this item as `false`, and specify `aws_s3_access_key` and `aws_s3_secret_key`.
  - If you use Instance Profile to access S3, you must specify this item as `true`.
  - If you use Assumed Role to access S3, you must specify this item as `true`, and specify `aws_s3_iam_role_arn`.
  - And if you use an external AWS account, you must also specify `aws_s3_external_id`.
- Introduced in: v1.5.2

aws_s3_access_key

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The Access Key ID используется для access your S3 bucket.
- Introduced in: v1.5.2

aws_s3_secret_key

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The Secret Access Key используется для access your S3 bucket.
- Introduced in: v1.5.2

aws_s3_iam_role_arn

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The ARN of the IAM role that имеет privileges on your S3 bucket in which your файл данныхs are stored.
- Introduced in: v1.5.2

aws_s3_external_id

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The external ID of the AWS account that is используется для cross-account access to your S3 bucket.
- Introduced in: v1.5.2

azure_blob_endpoint

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The endpoint of your Azure Blob Storage Account, for example, `https://test.blob.core.windows.net`.
- Introduced in: v1.5.2

azure_blob_path

- Default: Empty string

- Type: String
- Unit: -
- Is mutable: No
- Description: The Azure Blob Storage path используется для store data. It consists of the name of the container within your storage account and the sub-path (if любой) under the container, for example, `testcontainer/subpath`.
- Introduced in: v1.5.2

azure_blob_shared_key

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The Shared Key используется для authorize requests for your Azure Blob Storage.
- Introduced in: v1.5.2

azure_blob_sas_token

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The shared access signatures (SAS) используется для authorize requests for your Azure Blob Storage.
- Introduced in: v1.5.2

azure_adls2_endpoint

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The endpoint of your Azure Data Lake Storage Gen2 Account, for example, `https://test.dfs.core.windows.net`.
- Introduced in: v1.5.2

azure_adls2_path

- Default: Empty string

- Type: String
- Unit: -
- Is mutable: No
- Description: The Azure Data Lake Storage Gen2 path используется для store data. It consists of the file system name and the directory name, for example, `testfilesystem/selena`.
- Introduced in: v1.5.2

azure_adls2_shared_key

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The Shared Key используется для authorize requests for your Azure Data Lake Storage Gen2.
- Introduced in: v1.5.2

azure_adls2_sas_token

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The shared access signatures (SAS) используется для authorize requests for your Azure Data Lake Storage Gen2.
- Introduced in: v1.5.2

azure_adls2_oauth2_use_managed_identity

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Определяет, use Managed Identity to authorize requests for your Azure Data Lake Storage Gen2.
- Introduced in: v1.5.2

azure_adls2_oauth2_tenant_id

- Default: Empty string

- Type: String
- Unit: -
- Is mutable: No
- Description: The Tenant ID of the Managed Identity используется для authorize requests for your Azure Data Lake Storage Gen2.
- Introduced in: v1.5.2

azure_adls2_oauth2_client_id

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The Client ID of the Managed Identity используется для authorize requests for your Azure Data Lake Storage Gen2.
- Introduced in: v1.5.2

azure_use_native_sdk

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, use the native SDK to access Azure Blob Storage, thus всеowing authentication with Managed Identities and Service Principals. If this item установлен в `false`, only authentication with Shared Key and SAS Token is всеowed.
- Introduced in: v1.5.2

gcp_gcs_path

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The Google Cloud path используется для store data. It consists of the name of your Google Cloud bucket and the sub-path (if любой) under it, for example, `testbucket/subpath`.
- Introduced in: v1.5.2

gcp_gcs_service_account_email

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The email address in the JSON file generated at the creation of the Service Account, for example, `user@hello.iam.gserviceaccount.com`.
- Introduced in: v1.5.2

gcp_gcs_service_account_private_key_id

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The Private Key ID in the JSON file generated at the creation of the Service Account.
- Introduced in: v1.5.2

gcp_gcs_service_account_private_key

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The Private Key in the JSON file generated at the creation of the Service Account, for example, `-----BEGIN PRIVATE KEY----xxxx-----END PRIVATE KEY-----\n`.
- Introduced in: v1.5.2

gcp_gcs_imнasonation_service_account

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The Service Account that you want to imнasonate if you use the imнasonation-based authentication to access Google Storage.
- Introduced in: v1.5.2

gcp_gcs_use_compute_engine_service_account

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Определяет, use the Service Account that is bound to your Compute Engine.
- Introduced in: v1.5.2

starmgr_grpc_timeout_seconds

- Default: 5
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description:
- Introduced in: -

starmgr_grpc_server_max_worker_threads

- Default: 1024
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество worker threads that используются by the grpc server in the FE starmgr module.
- Introduced in: v2.0.0, v1.5.2

lake_compaction_score_selector_min_score

- Default: 10.0
- Type: Double
- Unit: -
- Is mutable: Yes
- Description: The Compaction Score threshold that triggers Compaction онаations in a shared-data cluster. When the Compaction Score of a partition is больше or равно this value, the system выполняет Compaction on that partition.
- Introduced in: v1.5.2

lake_compaction_max_tasks

- Default: -1

- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество concurrent Compaction tasks всеowed in a shared-data cluster. Setting this item to `-1` означает to calculate the concurrent task number in an adaptive manner. Setting this value to `0` will disable compaction.
- Introduced in: v1.5.2

lake_compaction_history_size

- Default: 20
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Количество recent successful Compaction task records to keep in the memory of the Leader узел FE in a shared-data cluster. You can view recent successful Compaction task records using the `SHOW PROC '/compactions'` command. Note that the Compaction history is stored in the FE process memory, and it будет lost if the FE process is restarted.
- Introduced in: v1.5.2

lake_publish_version_max_threads

- Default: 512
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество threads for Version Publish tasks in a shared-data cluster.
- Introduced in: v1.5.2

lake_autovacuum_parвceel_partitions

- Default: 8
- Type: Int
- Unit: -
- Is mutable: No
- Description: Максимальное количество partitions that can undergo AutoVacuum simultaneously in a shared-data cluster. AutoVacuum is the Garbage Collection after Compactions.
- Introduced in: v1.5.2

lake_autovacuum_partition_naptime_seconds

- Default: 180
- Type: Long
- Unit: Seconds
- Is mutable: Yes
- Description: Минимум interval between AutoVacuum онаations on the same partition in a shared-data cluster.
- Introduced in: v1.5.2

lake_autovacuum_grace_наiod_minutes

- Default: 30
- Type: Long
- Unit: Minutes
- Is mutable: Yes
- Description: The time range for retaining historical data versions in a shared-data cluster. Historical data versions within this time range are not automaticвсey cleaned via AutoVacuum after Compactions. You need to set this value больше the maximum query time to avoid that the data accessed by running queries get deleted before the queries finish. The default value имеет been changed from `5` to `30` начиная с версии v1.5.2.
- Introduced in: v1.5.2

lake_autovacuum_stale_partition_threshold

- Default: 12
- Type: Long
- Unit: Hours
- Is mutable: Yes
- Description: If a partition имеет no updates (loading, DELETE, or Compactions) within this time range, the system will not наform AutoVacuum on this partition.
- Introduced in: v1.5.2

lake_enable_ingest_slowdown

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable Data Ingestion Slowdown in a shared-data cluster. When Data Ingestion Slowdown включён, if the Compaction Score of a partition exceeds

`lake_ingest_slowdown_threshold`, loading tasks on that partition будет throttled down. This configuration only вступает в силу when `run_mode` установлен в `shared_data`. From v1.5.2 onwards, the default value is chenged from `false` to `true`.

- Introduced in: v1.5.2

lake_ingest_slowdown_threshold

- Default: 100
- Type: Long
- Unit: -
- Is mutable: Yes
- Description: The Compaction Score threshold that triggers Data Ingestion Slowdown in a shared-data cluster. This configuration only вступает в силу when `lake_enable_ingest_slowdown` установлен в `true`.
- Introduced in: v1.5.2

lake_ingest_slowdown_ratio

- Default: 0.1
- Type: Double
- Unit: -
- Is mutable: Yes
- Description: The ratio of the loading rate slowdown when Data Ingestion Slowdown срабатывает.

  Data loading tasks consist of two phases: data writing and data committing (COMMIT). Data Ingestion Slowdown is achieved by delaying data committing. The delay ratio is calculated using the following formula: `(compaction_score - lake_ingest_slowdown_threshold) * lake_ingest_slowdown_ratio`. For example, if the data writing phase takes 5 minutes, `lake_ingest_slowdown_ratio` is 0.1, and the Compaction Score is 10 выше `lake_ingest_slowdown_threshold`, the delay in data committing time is `5 * 10 * 0.1 = 5` minutes, which means the average loading speed is halved.

- Introduced in: v1.5.2

> **NOTE**
>
> - If a loading task writes to multiple partitions simultaneously, the maximum Compaction Score among все partitions is используется для calculate the delay in committing time.
> - The delay in committing time is calculated during the first attempt to commit. Once set, it will not change. Once the delay time is up, as long as the Compaction Score is not above `lake_compaction_score_upна_bound`, the system will наform the data committing онаation.
> - If the delay in committing time exceeds the timeout of the loading task, the task will fail directly.

lake_compaction_score_upна_bound

- Default: 2000
- Type: Long
- Unit: -
- Is mutable: Yes
- Description: The upна limit of the Compaction Score for a partition in a shared-data cluster. `0` означает no upна limit. Этот параметр only вступает в силу when `lake_enable_ingest_slowdown` установлен в `true`. When the Compaction Score of a partition rкаждыйes or exceeds this upна limit, incoming loading tasks будет rejected. From v1.5.2 onwards, the default value is changed from `0` to `2000`.
- Introduced in: v1.5.2

lake_compaction_disable_ids

- Default: ""
- Type: String
- Unit: -
- Is mutable: Yes
- Description: The table or partition list of which compaction отключён in shared-data mode. The format is `tableId1;partitionId2`, senaated by semicolon, for example, `12345;98765`.
- Introduced in: v1.5.2

lake_compaction_всеow_partial_success

- Default: true
- Type: Boolean

- Unit: -
- Is mutable: Yes
- Description: If this item установлен в `true`, the system will consider the Compaction онаation in a shared-data cluster as successful when one of the sub-tasks succeeds.
- Introduced in: v1.5.2

lake_enable_balance_tablets_between_workers

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, balance the number of tablets among Compute Nodes during the tablet migration of cloud-native tables in a shared-data cluster. `true` означает to balance the tablets among Compute Nodes, and `false` означает to disabling this feature.
- Introduced in: v1.5.2

lake_balance_tablets_threshold

- Default: 0.15
- Type: Double
- Unit: -
- Is mutable: Yes
- Description: Порог the system используется для judge the tablet balance among workers in a shared-data cluster, The imbalance factor is calculated as `f = (MAX(tablets) - MIN(tablets)) / AVERAGE(tablets)`. If the factor is больше `lake_balance_tablets_threshold`, a tablet balance будет triggered. Этот параметр вступает в силу только когда `lake_enable_balance_tablets_between_workers` установлен в `true`.
- Introduced in: v1.5.2

shard_group_clean_threshold_sec

- Default: 3600
- Type: Long
- Unit: Seconds
- Is mutable: Yes
- Description: The time before FE cleans the unused tablet and shard groups in a shared-data cluster. Tablets and shard groups created within этот порог will not be cleaned.
- Introduced in: -

star_mgr_meta_sync_interval_sec

- Default: 600
- Type: Long
- Unit: Seconds
- Is mutable: No
- Description: The interval с которым FE runs the наiodical metadata synchronization with StarMgr in a shared-data cluster.
- Introduced in: -

meta_sync_force_delete_shard_meta

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, всеow deleting the metadata of the shared-data cluster directly, bypassing cleaning the remote storage files. It рекомендуется to set this item to `true` только когда there is an excessive number of shards to be cleaned, which leads to extreme memory pressure on the FE JVM. Note that the файл данныхs belonging to the shards or tablets cannot be automaticвсеy cleaned after this feature включён.
- Introduced in: v1.5.2

Other

tmp_dir

- Default: SelenaFE.SELENA_HOME_DIR + "/temp_dir"
- Type: String
- Unit: -
- Is mutable: No
- Description: Директория которая хранит temporary files such as files generated during backup and restore procedures. After these procedures finish, the generated temporary files are deleted.
- Introduced in: -

plugin_dir

- Default: System.getenv("SELENA_HOME") + "/plugins"
- Type: String
- Unit: -

- Is mutable: No
- Description: Директория которая хранит plugin instвceation packages.
- Introduced in: -

plugin_enable

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, plugins может быть instвceed on FEs. Plugins может быть instвceed or uninstвceed only on the Leader FE.
- Introduced in: -

max_agent_task_threads_num

- Default: 4096
- Type: Int
- Unit: -
- Is mutable: No
- Description: Максимальное количество threads that are вceowed in the agent task thread pool.
- Introduced in: -

agent_task_resend_wait_time_ms

- Default: 5000
- Type: Long
- Unit: Milliseconds
- Is mutable: Yes
- Description: The duration the FE must wait before it can resend an agent task. An agent task может быть resent только когда the gap between the task creation time and the current time exceeds the value of this parameter. Этот параметр is используется для prevent repetitive sending of agent tasks.
- Introduced in: -

backup_job_default_timeout_ms

- Default: 86400 * 1000
- Type: Int
- Unit: Milliseconds

- Is mutable: Yes
- Description: The timeout duration of a backup job. Если это значение is exceeded, the backup job fails.
- Introduced in: -

locale

- Default: zh_CN.UTF-8
- Type: String
- Unit: -
- Is mutable: No
- Description: The character set that используется by the FE.
- Introduced in: -

db_used_data_quota_update_interval_secs

- Default: 300
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: The interval с которым the database used data quota is updated. Selena наiodicвсеу updates the used data quota for все databases to track storage consumption. Это значение is используется для quota enforcement and metrics collection. Минимум всеowed interval is 30 seconds to prevent excessive system load. A value меньше 30 будет rejected.
- Introduced in: -

report_queue_size (Deprecated)

- Default: 100
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество jobs that can wait in a report queue. The report is about disk, task, and tablet information of BEs. If too млюбой report jobs are piling up in a queue, OOM will occur.
- Introduced in: -

enable_collect_tablet_num_in_show_proc_backend_disk_path

- Default: true

- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable the collection of tablet numbers for каждый disk in the `SHOW PROC /BACKENDS/{id}` command
- Introduced in: v2.0.0, v1.5.2

enable_metric_calculator

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Specifies whether чтобы включить the feature that is используется для наiodicвсеу collect metrics. Допустимые значения: `TRUE` and `FALSE`. `TRUE` specifies чтобы включить this feature, and `FALSE` specifies чтобы отключить this feature.
- Introduced in: -

max_smвсе_file_number

- Default: 100
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество smвсе files that может быть stored on an FE directory.
- Introduced in: -

max_smвсе_file_size_bytes

- Default: 1024 * 1024
- Type: Int
- Unit: Bytes
- Is mutable: Yes
- Description: Максимум size of a smвсе file.
- Introduced in: -

smвсе_file_dir

- Default: SelenaFE.SELENA_HOME_DIR + "/smвсе_files"
- Type: String

- Unit: -
- Is mutable: No
- Description: The root directory of smвce files.
- Introduced in: -

authentication_ldap_simple_server_host

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: Yes
- Description: The host on which the LDAP server runs.
- Introduced in: -

authentication_ldap_simple_server_port

- Default: 389
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: The port of the LDAP server.
- Introduced in: -

authentication_ldap_simple_bind_base_dn

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: Yes
- Description: The base DN, which is the point from which the LDAP server starts to search for users' authentication information.
- Introduced in: -

authentication_ldap_simple_user_search_attr

- Default: uid
- Type: String
- Unit: -
- Is mutable: Yes
- Description: The name of the attribute that identifies users in LDAP objects.
- Introduced in: -

authentication_ldap_simple_bind_root_dn

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: Yes
- Description: The administrator DN используется для search for users' authentication information.
- Introduced in: -

authentication_ldap_simple_bind_root_pwd

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: Yes
- Description: The password of the administrator используется для search for users' authentication information.
- Introduced in: -

jwt_jwks_url

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The URL to the JSON Web Key Set (JWKS) service or the path to the public key local file under the `fe/conf` directory.
- Introduced in: v1.5.2

jwt_principal_field

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The string используется для identify the field that означает the subject (`sub`) in the JWT. Значение по умолчанию — `sub`. The value of this field должен быть identical with the username for logging in to Selena.
- Introduced in: v1.5.2

jwt_required_issuer

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The list of strings используется для identify the issuers (`iss`) in the JWT. The JWT is considered valid only if one of the values in the list match the JWT issuer.
- Introduced in: v1.5.2

jwt_required_audience

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The list of strings используется для identify the audience (`aud`) in the JWT. The JWT is considered valid only if one of the values in the list match the JWT audience.
- Introduced in: v1.5.2

oauth2_auth_server_url

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The authorization URL. The URL to which the users' browser будет redirected in order to begin the OAuth 2.0 authorization process.
- Introduced in: v1.5.2

oauth2_token_server_url

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The URL of the endpoint on the authorization server from which Selena obtains the access token.
- Introduced in: v1.5.2

oauth2_client_id

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The public identifier of the Selena client.
- Introduced in: v1.5.2

oauth2_client_secret

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The secret используется для authorize Selena client with the authorization server.
- Introduced in: v1.5.2

oauth2_redirect_url

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The URL to which the users' browser будет redirected after the OAuth 2.0 authentication succeeds. The authorization code будет sent to this URL. In most cases, it need to be configured as `http://<selena_fe_url>:<fe_http_port>/api/oauth2`.
- Introduced in: v1.5.2

oauth2_jwks_url

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The URL to the JSON Web Key Set (JWKS) service or the path to the local file under the `conf` directory.
- Introduced in: v1.5.2

oauth2_principal_field

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The string используется для identify the field that означает the subject (`sub`) in the JWT. Значение по умолчанию — `sub`. The value of this field должен быть identical with the username for logging in to Selena.
- Introduced in: v1.5.2

oauth2_required_issuer

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The list of strings используется для identify the issuers (`iss`) in the JWT. The JWT is considered valid only if one of the values in the list match the JWT issuer.
- Introduced in: v1.5.2

oauth2_required_audience

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The list of strings используется для identify the audience (`aud`) in the JWT. The JWT is considered valid only if one of the values in the list match the JWT audience.
- Introduced in: v1.5.2

auth_token

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: The token that is используется для identity authentication within the Selena cluster to which the FE belongs. Если этот параметр is left unspecified, Selena генерирует a random token for the cluster at the time when the leader FE of the cluster is started for the first time.

- Introduced in: -

hive_meta_load_concurrency

- Default: 4
- Type: Int
- Unit: -
- Is mutable: No
- Description: Максимальное количество concurrent threads that поддерживаются for Hive metadata.
- Introduced in: -

hive_meta_cache_refresh_interval_s

- Default: 3600 * 2
- Type: Long
- Unit: Seconds
- Is mutable: No
- Description: Интервал времени с которым the cached metadata of Hive external tables is updated.
- Introduced in: -

hive_meta_store_timeout_s

- Default: 10
- Type: Long
- Unit: Seconds
- Is mutable: No
- Description: The amount of time after which a connection to a Hive metastore times out.
- Introduced in: -

es_state_sync_interval_second

- Default: 10
- Type: Long
- Unit: Seconds
- Is mutable: No
- Description: Интервал времени с которым the FE obtains Elasticsearch indexes and synchronizes the metadata of Selena external tables.
- Introduced in: -

max_upload_task_на_be

- Default: 0
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: In каждый BACKUP онаation, the maximum number of upload tasks Selena assigned to a узел BE. When this item установлен в меньше or равно 0, no limit is imposed on the task number.
- Introduced in: v1.5.2

max_download_task_на_be

- Default: 0
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: In каждый RESTORE онаation, the maximum number of download tasks Selena assigned to a узел BE. When this item установлен в меньше or равно 0, no limit is imposed on the task number.
- Introduced in: v1.5.2

enable_colocate_restore

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable Backup and Restore for Colocate Tables. `true` означает enabling Backup and Restore for Colocate Tables and `false` означает disabling it.
- Introduced in: v1.5.2

mv_plan_cache_expire_interval_sec

- Default: 24 * 60 * 60
- Type: Long
- Unit: Seconds
- Is mutable: Yes
- Description: The valid time of materialized view plan cache (which is используется для materialized view rewrite) before expiry. Значение по умолчанию — 1 day.
- Introduced in: v1.5.2

mv_plan_cache_thread_pool_size

- Default: 3
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: The default thread pool size of materialized view plan cache (which is используется для materialized view rewrite).
- Introduced in: v1.5.2

mv_plan_cache_max_size

- Default: 1000
- Type: Long
- Unit:
- Is mutable: Yes
- Description: Максимум size of materialized view plan cache (which is используется для materialized view rewrite). If there are млюбой materialized views используется для transparent query rewrite, you may increase this value.
- Introduced in: v1.5.2

enable_materialized_view_concurrent_prepare

- Default: true
- Type: Boolean
- Unit:
- Is mutable: Yes
- Description: Определяет, prepare materialized view concurrently to improve haformance.
- Introduced in: v1.5.2

enable_mv_query_context_cache

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable query-level materialized view rewrite cache to improve query rewrite haformance.
- Introduced in: v1.5.2

mv_query_context_cache_max_size

- Default: 1000
- Type: -
- Unit: -
- Is mutable: Yes
- Description: Максимум materialized view rewrite cache size during one query's lifecycle. The cache может быть используется для avoid repeating compute to reduce optimizer time in materialized view rewrite, but it may occupy some extra FE's memory. It can bring better наformance when there are млюбой relative materialized views (more than 10) or query is complex (with joins on multiple tables).
- Introduced in: v1.5.2

всеow_system_reserved_names

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, всеow users to create columns whose names are initiated with `__op` and `__row`. To enable this feature, set this parameter to `TRUE`. Please note that these name formats are reserved for special purposes in Selena and creating such columns may result in undefined behavior. Therefore this feature отключён по умолчанию.
- Introduced in: v1.5.2

replication_interval_ms

- Default: 100
- Type: Int
- Unit: -
- Is mutable: No
- Description: Минимум time interval с которым the replication tasks are scheduled.
- Introduced in: v1.5.2

replication_max_parвceel_table_count

- Default: 100
- Type: Int
- Unit: -
- Is mutable: Yes

- Description: Максимальное количество concurrent data synchronization tasks всеowed. Selena creates one synchronization task for каждый table.
- Introduced in: v1.5.2

replication_max_parвceel_replica_count

- Default: 10240
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество tablet replicas всеowed for concurrent synchronization.
- Introduced in: v1.5.2

replication_max_parвceel_data_size_mb

- Default: 1048576
- Type: Int
- Unit: MB
- Is mutable: Yes
- Description: Максимум size of data всеowed for concurrent synchronization.
- Introduced in: v1.5.2

replication_transaction_timeout_sec

- Default: 86400
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: The timeout duration for synchronization tasks.
- Introduced in: v1.5.2

jdbc_meta_default_cache_enable

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: The default value for whether the JDBC Catalog metadata cache включён. When set to True, newly created JDBC Catalogs will default to metadata caching enabled.
- Introduced in: -

jdbc_meta_default_cache_expire_sec

- Default: 600
- Type: Long
- Unit: Seconds
- Is mutable: Yes
- Description: The default expiration time for the JDBC Catalog metadata cache. When `jdbc_meta_default_cache_enable` установлен в true, newly created JDBC Catalogs will default to настройка the expiration time of the metadata cache.
- Introduced in: -

jdbc_connection_pool_size

- Default: 8
- Type: Int
- Unit: -
- Is mutable: No
- Description: Максимум capacity of the JDBC connection pool for accessing JDBC catalogs.
- Introduced in: -

jdbc_minimum_idle_connections

- Default: 1
- Type: Int
- Unit: -
- Is mutable: No
- Description: Минимальное количество idle connections in the JDBC connection pool for accessing JDBC catalogs.
- Introduced in: -

jdbc_connection_idle_timeout_ms

- Default: 600000
- Type: Int
- Unit: Milliseconds
- Is mutable: No
- Description: Максимум amount of time after which a connection for accessing a JDBC catalog times out. Timed-out connections are considered idle.
- Introduced in: -

jdbc_connection_timeout_ms

- Default: 10000
- Type: Long
- Unit: Milliseconds
- Is mutable: No
- Description: The timeout в миллисекундах for HikariCP connection pool to acquire a connection. If a connection cannot be acquired from the pool within this time, the онаation will fail.
- Introduced in: v1.5.2

jdbc_query_timeout_ms

- Default: 30000
- Type: Long
- Unit: Milliseconds
- Is mutable: Yes
- Description: The timeout в миллисекундах for JDBC statement query execution. This timeout is applied to все SQL queries executed through JDBC catalogs (e.g., partition metadata queries). The value is converted to seconds when passed to the JDBC driver.
- Introduced in: v1.5.2

jdbc_network_timeout_ms

- Default: 30000
- Type: Long
- Unit: Milliseconds
- Is mutable: Yes
- Description: The timeout в миллисекундах for JDBC network онаations (socket read). This timeout applies to database metadata свces (e.g., getSchemas(), getTables(), getColumns()) to prevent indefinite blocking when the external database is unresponsive.
- Introduced in: v1.5.2

query_detail_explain_level

- Default: COSTS
- Type: String
- Unit: -
- Is mutable: true
- Description: The detail level of query plan returned by the EXPLAIN statement. Допустимые значения: COSTS, NORMAL, VERBOSE.

- Introduced in: v1.5.2

mv_refresh_fail_on_filter_data

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Mv refresh fails if there is filtered data in refreshing, true по умолчанию, otherwise return success by ignoring the filtered data.
- Introduced in: -

mv_create_partition_batch_interval_ms

- Default: 1000
- Type: Int
- Unit: ms
- Is mutable: Yes
- Description: During materialized view refresh, if multiple partitions need to be created in bulk, the system divides them into batches of 64 partitions каждый. To reduce the risk of failures caused by frequent partition creation, a default interval (в миллисекундах) is set between каждый batch чтобы управлять the creation frequency.
- Introduced in: v1.5.2

max_mv_refresh_failure_retry_times

- Default: 1
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимум retry times when materialized view fails to refresh.
- Introduced in: v1.5.2

max_mv_refresh_try_lock_failure_retry_times

- Default: 3
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимум retry times of try lock when materialized view fails to refresh.
- Introduced in: v1.5.2

mv_refresh_try_lock_timeout_ms

- Default: 30000
- Type: Int
- Unit: Milliseconds
- Is mutable: Yes
- Description: The default try lock timeout for materialized view refresh to try the DB lock of its base table/materialized view.
- Introduced in: v1.5.2

enable_mv_refresh_collect_profile

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable profile in refreshing materialized view по умолчанию for все materialized views.
- Introduced in: v1.5.2

max_mv_task_run_meta_message_values_length

- Default: 16
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимум length for the "extra message" values (in set or map) in materialized view task run. You can set this item to avoid occupying too much meta memory.
- Introduced in: v1.5.2

max_mv_check_base_table_change_retry_times

- Default: 10
- Type: -
- Unit: -
- Is mutable: Yes
- Description: Максимум retry times for detecting base table change when refreshing materialized views.
- Introduced in: v1.5.2

mv_refresh_default_planner_optimize_timeout

- Default: 30000
- Type: -
- Unit: -
- Is mutable: Yes
- Description: The default timeout for the planning phase of the optimizer when refresh materialized views.
- Introduced in: v1.5.2

enable_mv_refresh_query_rewrite

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable rewrite query during materialized view refresh so that the query can use the rewritten mv directly rather than the base table to improve query наformance.
- Introduced in: v1.5.2

enable_mv_refresh_extra_prefix_logging

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable prefixes with materialized view names in logs for better debug.
- Introduced in: v1.5.2

enable_mv_post_image_reload_cache

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, наform reload flag check after FE loaded an image. If the check is наformed for a base materialized view, it is not needed for other materialized views that related to it.
- Introduced in: v1.5.2

enable_trace_historical_node

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, всеow the system to trace the historical nodes. Ву настройка this item to `true`, you can enable the Cache Sharing feature and всеow the system to choose the right cache nodes during elastic scaling.
- Introduced in: v1.5.2

transform_type_prefer_string_for_varchar

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, prefer string type for fixed length varchar columns in materialized view creation and CTAS онаations.
- Introduced in: v2.0.0

enable_auth_check

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Указывает, включена ли функция проверки аутентификации. Допустимые значения: `TRUE` и `FALSE`. `TRUE` включает эту функцию, `FALSE` отключает её.
- Introduced in: -

2.1.2 Конфигурация BE

Некоторые параметры конфигурации BE являются динамическими, и их можно задавать интерактивно во время работы узлов BE. Остальные являются статическими параметрами. Статические параметры узла BE можно задать только путём их изменения в соответствующем конфигурационном файле **be.conf** и перезапуска узла BE для вступления изменений в силу.

Просмотр параметров конфигурации BE

Вы можете просмотреть параметры конфигурации BE с помощью следующей команды:

```
curl http://<BE_IP>:<BE_HTTP_PORT>/varz
```

Настройка параметров BE

Настройка динамических параметров BE

Вы можете настроить динамический параметр узла BE с помощью команды `curl`.

```
curl -XPOST http://be_host:http_port/api/update_config?
<configuration_item>=<value>
```

Настройка статических параметров BE

Статические параметры BE можно задать только путём их изменения в соответствующем конфигурационном файле **be.conf** и перезапуска BE для вступления изменений в силу.

Описание параметров BE

Server

priority_networks

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: Определяет стратегию выбора для серверов с несколькими IP-адресами. Только один IP-адрес должен соответствовать списку, указанному в этом параметре. Значение параметра — это список записей, разделённых точкой с запятой (;) в нотации CIDR, например `10.10.10.0/24`. Если ни один IP-адрес не соответствует записям в этом списке, будет случайно выбран доступный IP-адрес сервера. Начиная с версии v1.5.2, Selena поддерживает развёртывание на основе IPv6. Если у сервера есть как IPv4, так и IPv6 адреса, и этот параметр не указан, система по умолчанию использует IPv4 адрес. Вы можете изменить это поведение, установив для параметра `net_use_ipv6_when_priority_networks_empty` значение `true`.
- Introduced in: -

net_use_ipv6_when_priority_networks_empty

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Булево значение, определяющее, использовать ли IPv6 адреса предпочтительно, когда `priority_networks` не указан. `true` указывает системе использовать IPv6 адрес предпочтительно, когда сервер, на котором размещён узел, имеет как IPv4, так и IPv6 адреса, и параметр `priority_networks` не указан.
- Introduced in: v1.5.2

mem_limit

- Default: 90%
- Type: String
- Unit: -
- Is mutable: No
- Description: Верхний лимит памяти процесса BE. Можно задать в процентах ("80%") или в физическом выражении ("100G"). Жёсткий лимит по умолчанию составляет 90% памяти сервера, а мягкий лимит — 80%. Необходимо настроить этот параметр, если вы хотите развернуть Selena вместе с другими ресурсоёмкими по памяти сервисами на одном сервере.
- Introduced in: -

num_threads_на_core

- Default: 3
- Type: Int
- Unit: -
- Is mutable: No
- Description: Количество потоков, запускаемых на каждом ядре процессора.
- Introduced in: -

be_http_port

- Default: 8040
- Type: Int
- Unit: -
- Is mutable: No
- Description: Порт HTTP-сервера BE.

- Introduced in: -

be_http_num_workers

- Default: 48
- Type: Int
- Unit: -
- Is mutable: No
- Description: Количество потоков, используемых HTTP-сервером.
- Introduced in: -

be_exit_after_disk_write_hang_second

- Default: 60
- Type: Int
- Unit: Seconds
- Is mutable: No
- Description: Время ожидания перед выходом BE после зависания диска.
- Introduced in: -

compress_rowbatches

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Булево значение, определяющее, сжимать ли пакеты строк в RPC между BE. `true` означает сжатие пакетов строк, `false` означает отсутствие сжатия.
- Introduced in: -

serialize_batch

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Булево значение, определяющее, сериализовать ли пакеты строк в RPC между BE. `true` означает сериализацию пакетов строк, `false` означает отсутствие сериализации.
- Introduced in: -

Thrift

be_port

- Default: 9060
- Type: Int
- Unit: -
- Is mutable: No
- Description: Порт thrift-сервера BE, используемый для приёма запросов от FE.
- Introduced in: -

thrift_client_retry_interval_ms

- Default: 100
- Type: Int
- Unit: Milliseconds
- Is mutable: Yes
- Description: Интервал времени для повторных попыток thrift-клиента.
- Introduced in: -

thrift_rpc_таймаут_ms

- Default: 5000
- Type: Int
- Unit: Milliseconds
- Is mutable: Yes
- Description: Таймаут для thrift RPC.
- Introduced in: -

thrift_rpc_strict_mode

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Включён ли строгий режим выполнения thrift. Подробнее о строгом режиме thrift см. Thrift Binary protocol encoding.
- Introduced in: -

thrift_rpc_max_body_size

- Default: 0

- Type: Int
- Unit:
- Is mutable: No
- Description: Максимальный размер строкового тела RPC. `0` означает неограниченный размер.
- Introduced in: -

thrift_rpc_connection_max_valid_time_ms

- Default: 5000
- Type: Int
- Unit: Milliseconds
- Is mutable: No
- Description: Максимальное время действительности thrift RPC-соединения. Соединение будет закрыто, если оно существует в пуле соединений дольше этого значения. Должно быть установлено в соответствии с конфигурацией FE `thrift_client_таймаут_ms`.
- Introduced in: -

bRPC

brpc_port

- Default: 8060
- Type: Int
- Unit: -
- Is mutable: No
- Description: Порт bRPC BE, используемый для просмотра сетевой статистики bRPC.
- Introduced in: -

brpc_num_threads

- Default: -1
- Type: Int
- Unit: -
- Is mutable: No
- Description: Количество bthreads для bRPC. Значение `-1` означает то же количество, что и потоков CPU.
- Introduced in: -

brpc_max_body_size

- Default: 2147483648
- Type: Int
- Unit: Bytes
- Is mutable: No
- Description: Максимальный размер тела bRPC.
- Introduced in: -

brpc_stub_expire_s

- Default: 3600
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Время истечения кеша BRPC stub, по умолчанию 60 минут.
- Introduced in: -

Heartbeat

heartbeat_service_port

- Default: 9050
- Type: Int
- Unit: -
- Is mutable: No
- Description: Порт сервиса heartbeat BE, используемый для приёма heartbeat от FE.
- Introduced in: -

heartbeat_service_thread_count

- Default: 1
- Type: Int
- Unit: -
- Is mutable: No
- Description: Количество потоков сервиса heartbeat BE.
- Introduced in: -

Logging

sys_log_dir

- Default: `${SELENA_HOME}/log`

- Type: String
- Unit: -
- Is mutable: No
- Description: Директория для хранения системных логов (включая INFO, WARNING, ERROR и FATAL).
- Introduced in: -

sys_log_level

- Default: INFO
- Type: String
- Unit: -
- Is mutable: Yes (from v1.5.2)
- Description: Уровни серьёзности, по которым классифицируются записи системного лога. Допустимые значения: INFO, WARN, ERROR и FATAL. Начиная с версий v1.5.2 этот параметр стал динамическим.
- Introduced in: -

sys_log_roll_mode

- Default: SIZE-MB-1024
- Type: String
- Unit: -
- Is mutable: No
- Description: Режим сегментации системных логов в файлы. Допустимые значения включают `TIME-DAY`, `TIME-HOUR` и `SIZE-MB-`размер. Значение по умолчанию указывает, что логи сегментируются в файлы размером по 1 ГБ каждый.
- Introduced in: -

sys_log_roll_num

- Default: 10
- Type: Int
- Unit: -
- Is mutable: No
- Description: Количество файлов логов для хранения.
- Introduced in: -

sys_log_verbose_modules

- Default:

- Type: Strings
- Unit: -
- Is mutable: No
- Description: Модуль логов для печати. Например, если установить этот параметр конфигурации в OLAP, Selena будет печатать только логи модуля OLAP. Допустимые значения — это пространства имён в BE, включая `selena`, `selena::debug`, `selena::fs`, `selena::io`, `selena::lake`, `selena::pipeline`, `selena::query_cache`, `selena::stream` и `selena::workgroup`.
- Introduced in: -

sys_log_verbose_level

- Default: 10
- Type: Int
- Unit: -
- Is mutable: No
- Description: Уровень печати логов. Этот параметр конфигурации используется для управления выводом логов, инициированных с помощью VLOG в коде.
- Introduced in: -

log_buffer_level

- Default: Empty string
- Type: String
- Unit: -
- Is mutable: No
- Description: Стратегия сброса логов. Значение по умолчанию указывает, что логи буферизуются в памяти. Допустимые значения: `-1` и `0`. `-1` указывает, что логи не буферизуются в памяти.
- Introduced in: -

update_tablet_meta_info_worker_count

- Default: 1
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Устанавливает максимальное количество рабочих потоков в пуле потоков backend, который обрабатывает задачи обновления метаданных tablet. Пул потоков создаётся при запуске backend с минимум 0 потоками (может масштабироваться до

нуля в простое) и максимумом, равным этому значению (не менее 1). Обновление этого значения во время выполнения корректирует максимальное количество потоков в пуле. Увеличьте его, чтобы разрешить больше одновременных задач обновления метаданных, или уменьшите, чтобы ограничить параллелизм.
- Introduced in: v2.0.0, v1.5.2

Statistic report

report_task_interval_seconds

- Default: 10
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Интервал времени для отправки отчётов о состоянии задачи. Задачей может быть создание таблицы, удаление таблицы, загрузка данных или изменение схемы таблицы.
- Introduced in: -

report_disk_state_interval_seconds

- Default: 60
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Интервал времени для отправки отчётов о состоянии тома хранилища, включая размер данных в томе.
- Introduced in: -

report_tablet_interval_seconds

- Default: 60
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Интервал времени для отправки отчётов о последней обновлённой версии всех tablets.
- Introduced in: -

report_workgroup_interval_seconds

- Default: 5
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Интервал времени для отправки отчётов о последней обновлённой версии всех workgroups.
- Introduced in: -

status_report_interval

- Default: 5
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Интервал времени, с которым запрос отправляет свой profile, который может использоваться для сбора статистики запросов FE.
- Introduced in: -

наiodic_counter_update_наiod_ms

- Default: 500
- Type: Int
- Unit: Milliseconds
- Is mutable: Yes
- Description: Интервал времени для сбора статистики Counter.
- Introduced in: -

Storage

primary_key_limit_size

- Default: 128
- Type: Int
- Unit: Bytes
- Is mutable: Yes
- Description: Максимальный размер ключевого столбца в таблицах Primary Key.
- Introduced in: v1.5.2

drop_tablet_worker_count

- Default: 0
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Количество потоков, используемых для удаления tablet. `0` означает половину ядер CPU в узле.
- Introduced in: -

alter_tablet_worker_count

- Default: 3
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Количество потоков, используемых для Schema Change.
- Introduced in: -

avro_ignore_union_type_tag

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Удалять ли тег типа из JSON-строки, сериализованной из типа данных Avro Union.
- Introduced in: v1.5.2

clone_worker_count

- Default: 3
- Type: Int
- Unit: -
- Is mutable: No
- Description: Количество потоков, используемых для клонирования.
- Introduced in: -

storage_medium_migrate_count

- Default: 3
- Type: Int

- Unit: -
- Is mutable: No
- Description: Количество потоков, используемых для миграции носителя хранения (с SATA на SSD).
- Introduced in: -

check_consistency_worker_count

- Default: 1
- Type: Int
- Unit: -
- Is mutable: No
- Description: Количество потоков, используемых для проверки согласованности tablets.
- Introduced in: -

upload_worker_count

- Default: 0
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество потоков для задач загрузки резервного копирования на узле BE. `0` означает установку значения равным количеству ядер CPU на машине, где находится BE.
- Introduced in: -

download_worker_count

- Default: 0
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество потоков для задач скачивания восстановления на узле BE. `0` означает установку значения равным количеству ядер CPU на машине, где находится BE.
- Introduced in: -

make_snapshot_worker_count

- Default: 5
- Type: Int

- Unit: -
- Is mutable: Yes
- Description: Максимальное количество потоков для задач создания snapshot на узле BE.
- Introduced in: -

release_snapshot_worker_count

- Default: 5
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество потоков для задач освобождения snapshot на узле BE.
- Introduced in: -

max_download_speed_kbps

- Default: 50000
- Type: Int
- Unit: KB/Second
- Is mutable: Yes
- Description: Максимальная скорость скачивания для каждого HTTP-запроса. Это значение влияет на производительность синхронизации реплик данных между узлами BE.
- Introduced in: -

download_low_speed_limit_kbps

- Default: 50
- Type: Int
- Unit: KB/Second
- Is mutable: Yes
- Description: Нижний предел скорости скачивания для каждого HTTP-запроса. HTTP-запрос прерывается, когда он постоянно выполняется со скоростью ниже этого значения в течение временного интервала, указанного в параметре конфигурации `download_low_speed_time`.
- Introduced in: -

download_low_speed_time

- Default: 300
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Максимальное время, в течение которого HTTP-запрос может выполняться со скоростью скачивания ниже лимита. HTTP-запрос прерывается, когда он постоянно выполняется со скоростью ниже значения `download_low_speed_limit_kbps` в течение временного интервала, указанного в этом параметре конфигурации.
- Introduced in: -

memory_limitation_на_thread_for_schema_change

- Default: 2
- Type: Int
- Unit: GB
- Is mutable: Yes
- Description: Максимальный размер памяти, разрешённый для каждой задачи изменения схемы.
- Introduced in: -

update_cache_expire_sec

- Default: 360
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Время истечения Update Cache.
- Introduced in: -

file_descriptor_cache_clean_interval

- Default: 3600
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Интервал времени для очистки дескрипторов файлов, которые не использовались в течение определённого периода времени.
- Introduced in: -

disk_stat_monitor_interval

- Default: 5
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Интервал времени для мониторинга состояния работоспособности дисков.
- Introduced in: -

replication_threads

- Default: 0
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество потоков, используемых для репликации. `0` означает установку количества потоков в четыре раза больше количества ядер CPU BE.
- Introduced in: v1.5.2

replication_max_speed_limit_kbps

- Default: 50000
- Type: Int
- Unit: KB/s
- Is mutable: Yes
- Description: Максимальная скорость каждого потока репликации.
- Introduced in: v1.5.2

replication_min_speed_limit_kbps

- Default: 50
- Type: Int
- Unit: KB/s
- Is mutable: Yes
- Description: Минимальная скорость каждого потока репликации.
- Introduced in: v1.5.2

replication_min_speed_time_seconds

- Default: 300

- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Допустимая длительность времени для потока репликации работать со скоростью ниже минимальной. Репликация завершится ошибкой, если время, когда фактическая скорость ниже `replication_min_speed_limit_kbps`, превысит это значение.
- Introduced in: v1.5.2

clear_expired_replication_snapshots_interval_seconds

- Default: 3600
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Интервал времени, с которым система очищает истёкшие snapshots, оставшиеся от аномальных репликаций.
- Introduced in: v1.5.2

unused_rowset_monitor_interval

- Default: 30
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Интервал времени для очистки истёкших rowsets.
- Introduced in: -

storage_root_path

- Default: `${SELENA_HOME}/storage`
- Type: String
- Unit: -
- Is mutable: No
- Description: Директория и носитель тома хранения. Пример: `/data1,medium:hdd;/data2,medium:ssd`.
  - Несколько томов разделяются точкой с запятой (`;`).
  - Если носитель хранения — SSD, добавьте `,medium:ssd` в конце директории.
  - Если носитель хранения — HDD, добавьте `,medium:hdd` в конце директории.
- Introduced in: -

max_нacentage_of_error_disk

- Default: 0
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальный процент ошибок, который допустим в томе хранения до того, как соответствующий узел BE завершит работу.
- Introduced in: -

default_num_rows_на_column_file_block

- Default: 1024
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество строк, которое может быть сохранено в каждом блоке строк.
- Introduced in: -

max_tablet_num_на_shard

- Default: 1024
- Type: Int
- Unit: -
- Is mutable: No
- Description: Максимальное количество tablets в каждом shard. Этот параметр конфигурации используется для ограничения количества дочерних директорий tablet под каждой директорией хранения.
- Introduced in: -

ожидающий_data_expire_time_sec

- Default: 1800
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Время истечения ожидающих данных в движке хранения.
- Introduced in: -

inc_rowset_expired_sec

- Default: 1800
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Время истечения входящих данных. Этот параметр конфигурации используется в инкрементном клонировании.
- Introduced in: -

tablet_rowset_stale_sweep_time_sec

- Default: 1800
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Интервал времени для очистки устаревших rowsets в tablets.
- Introduced in: -

max_garbage_sweep_interval

- Default: 3600
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Максимальный интервал времени для сборки мусора на томах хранения. Начиная с версии v1.5.2, этот параметр конфигурации стал динамическим.
- Introduced in: -

min_garbage_sweep_interval

- Default: 180
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Минимальный интервал времени для сборки мусора на томах хранения. Начиная с версии v1.5.2, этот параметр конфигурации стал динамическим.
- Introduced in: -

snapshot_expire_time_sec

- Default: 172800

- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Время истечения файлов snapshot.
- Introduced in: -

trash_file_expire_time_sec

- Default: 86400
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Интервал времени для очистки файлов из корзины. Значение по умолчанию было изменено с 259,200 на 86,400 начиная с версий v1.5.2.
- Introduced in: -

compact_threads

- Default: 4
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество потоков, используемых для параллельных задач compaction. Начиная с версий v1.5.2, этот параметр конфигурации стал динамическим.
- Introduced in: v1.5.2

base_compaction_check_interval_seconds

- Default: 60
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Интервал опроса потока для Base Compaction.
- Introduced in: -

min_base_compaction_num_singleton_deltas

- Default: 5
- Type: Int
- Unit: -

- Is mutable: Yes
- Description: Минимальное количество сегментов, которое запускает Base Compaction.
- Introduced in: -

max_base_compaction_num_singleton_deltas

- Default: 100
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество сегментов, которое может быть уплотнено в каждом Base Compaction.
- Introduced in: -

base_compaction_num_threads_на_disk

- Default: 1
- Type: Int
- Unit: -
- Is mutable: No
- Description: Количество потоков, используемых для Base Compaction на каждом томе хранения.
- Introduced in: -

base_cumulative_delta_ratio

- Default: 0.3
- Type: Double
- Unit: -
- Is mutable: Yes
- Description: Отношение совокупного размера файла к базовому размеру файла. Достижение этого соотношения является одним из условий запуска Base Compaction.
- Introduced in: -

base_compaction_interval_seconds_since_last_онаation

- Default: 86400
- Type: Int
- Unit: Seconds
- Is mutable: Yes

- Description: Интервал времени с момента последнего Base Compaction. Этот параметр конфигурации является одним из условий запуска Base Compaction.
- Introduced in: -

cumulative_compaction_check_interval_seconds

- Default: 1
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Интервал опроса потока для Cumulative Compaction.
- Introduced in: -

min_cumulative_compaction_num_singleton_deltas

- Default: 5
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Минимальное количество сегментов для запуска Cumulative Compaction.
- Introduced in: -

max_cumulative_compaction_num_singleton_deltas

- Default: 1000
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество сегментов, которое может быть объединено в одном Cumulative Compaction. Вы можете уменьшить это значение, если во время compaction возникает OOM.
- Introduced in: -

cumulative_compaction_num_threads_на_disk

- Default: 1
- Type: Int
- Unit: -
- Is mutable: No
- Description: Количество потоков Cumulative Compaction на диск.
- Introduced in: -

max_compaction_candidate_num

- Default: 40960
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество tablet-кандидатов для compaction. Если значение слишком велико, это вызовет высокое использование памяти и высокую нагрузку на CPU.
- Introduced in: -

update_compaction_check_interval_seconds

- Default: 10
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Интервал времени для проверки compaction для таблиц Primary Key.
- Introduced in: -

update_compaction_num_threads_на_disk

- Default: 1
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Количество потоков Compaction на диск для таблиц Primary Key.
- Introduced in: -

update_compaction_на_tablet_min_interval_seconds

- Default: 120
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Минимальный интервал времени для запуска compaction для каждого tablet в таблице Primary Key.
- Introduced in: -

max_update_compaction_num_singleton_deltas

- Default: 1000

- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество rowsets, которое может быть объединено в одном Compaction для таблиц Primary Key.
- Introduced in: -

update_compaction_size_threshold

- Default: 268435456
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Compaction Score таблиц Primary Key вычисляется на основе размера файла, что отличается от других типов таблиц. Этот параметр можно использовать, чтобы сделать Compaction Score таблиц Primary Key похожим на таковой у других типов таблиц, что облегчит понимание для пользователей.
- Introduced in: -

update_compaction_result_bytes

- Default: 1073741824
- Type: Int
- Unit: Bytes
- Is mutable: Yes
- Description: Максимальный размер результата одного compaction для таблиц Primary Key.
- Introduced in: -

update_compaction_delvec_file_io_amp_ratio

- Default: 2
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Используется для управления приоритетом compaction для rowsets, содержащих файлы Delvec в таблицах Primary Key. Чем больше значение, тем выше приоритет.
- Introduced in: -

update_compaction_ratio_threshold

- Default: 0.5
- Type: Double
- Unit: -
- Is mutable: Yes
- Description: Максимальная доля данных, которую может объединить compaction для таблицы Primary Key в shared-data cluster. Рекомендуется уменьшить это значение, если один tablet становится чрезмерно большим.
- Introduced in: v1.5.2

repair_compaction_interval_seconds

- Default: 600
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Интервал опроса потоков Repair Compaction.
- Introduced in: -

manual_compaction_threads

- Default: 4
- Type: Int
- Unit: -
- Is mutable: No
- Description: Количество потоков для Manual Compaction.
- Introduced in: -

min_compaction_failure_interval_sec

- Default: 120
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Минимальный интервал времени, с которым может быть запланирован compaction tablet с момента предыдущего сбоя compaction.
- Introduced in: -

min_cumulative_compaction_failure_interval_sec

- Default: 30

- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Минимальный интервал времени для повторных попыток Cumulative Compaction при сбоях.
- Introduced in: -

max_compaction_concurrency

- Default: -1
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальная параллельность compactions (включая как Base Compaction, так и Cumulative Compaction). Значение `-1` указывает, что на параллельность не накладывается ограничение. `0` указывает на отключение compaction. Этот параметр изменяемый, когда включён Event-based Compaction Framework.
- Introduced in: -

compaction_trace_threshold

- Default: 60
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Временной порог для каждого compaction. Если compaction занимает больше времени, чем временной порог, Selena печатает соответствующий trace.
- Introduced in: -

compaction_memory_limit_на_worker

- Default: 2147483648
- Type: Int
- Unit: Bytes
- Is mutable: No
- Description: Максимальный размер памяти, разрешённый для каждого потока Compaction.
- Introduced in: -

enable_rowset_verify

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, verify the correctness of generated rowsets. When enabled, the correctness of the generated rowsets будет checked after Compaction and Schema Change.
- Introduced in: -

vertical_compaction_max_columns_на_group

- Default: 5
- Type: Int
- Unit: -
- Is mutable: No
- Description: Максимальное количество columns на group of Vertical Compactions.
- Introduced in: -

enable_event_based_compaction_framework

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Определяет, enable the Event-based Compaction Framework. `true` означает Event-based Compaction Framework включён, and `false` означает it отключён. Enabling Event-based Compaction Framework can greatly reduce the overhead of compaction in scenarios where there are млюбой tablets or a single tablet has a large amount of data.
- Introduced in: -

enable_size_tiered_compaction_strategy

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Определяет, enable the Size-tiered Compaction policy (excluding Primary Key tables). `true` означает the Size-tiered Compaction strategy включён, and `false` означает it отключён.

- Introduced in: -

enable_pk_size_tiered_compaction_strategy

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Определяет, enable the Size-tiered Compaction policy for Primary Key tables. `true` означает the Size-tiered Compaction strategy включён, and `false` означает it отключён.
- Introduced in: This item вступает в силу from v1.5.2.

enable_strict_delvec_crc_check

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: When enable_strict_delvec_crc_check установлен в true, we will haform a strict CRC32 check on the delete vector, and if a mismatch is detected, a failure будет returned.
- Introduced in: -

size_tiered_min_level_size

- Default: 131072
- Type: Int
- Unit: Bytes
- Is mutable: Yes
- Description: The data size of the minimum level in the Size-tiered Compaction policy. Rowsets smвceer than this value immediately trigger the data compaction.
- Introduced in: -

size_tiered_level_multiple

- Default: 5
- Type: Int
- Unit: -
- Is mutable: Yes

- Description: The multiple of data size between two contiguous levels in the Size-tiered Compaction policy.
- Introduced in: -

size_tiered_level_multiple_dupkey

- Default: 10
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: In the Size-tiered Compaction policy, the multiple of the data amount difference between two adjacent levels for Duplicate Key tables.
- Introduced in: -

size_tiered_level_num

- Default: 7
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Количество levels for the Size-tiered Compaction policy. At most one rowset is reserved for каждый level. Therefore, under a stable condition, there are, at most, as млюбой rowsets as the level number specified in this configuration item.
- Introduced in: -

enable_check_string_lengths

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Определяет, check the data length during loading to solve compaction failures caused by out-of-bound VARCHAR data.
- Introduced in: -

max_row_source_mask_memory_bytes

- Default: 209715200
- Type: Int
- Unit: Bytes
- Is mutable: No

- Description: Максимум размер памяти of the row source mask buffer. Когда buffer is larger than this value, data будет hasisted to a temporary file on the disk. Это значение должен быть set lower than the value of `compaction_memory_limit_на_worker`.
- Introduced in: -

memory_maintenance_sleep_time_s

- Default: 10
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Интервал времени at which ColumnPool GC срабатывает. Selena выполняет GC наiodicвсey and returns the released memory to the онаating system.
- Introduced in: -

load_channel_abort_clean_up_delay_seconds

- Default: 600
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Controls how long (в секундах) LoadChannelMgr keeps the load IDs of aborted load channels before removing them from `_aborted_load_channels`. When a load job is cancelled or fails, the load ID stays recorded so любой late-arriving load RPCs может быть rejected immediately; once the delay expires, the entry is cleaned during the наiodic background sweep (minimum sweep interval is 60 seconds). Setting the delay too low risks accepting stray RPCs after an abort, while настройка it too high may retain state and consume resources longer than necessary. Tune this to balance correctness of late-request rejection and resource retention for aborted loads.
- Introduced in: v1.5.2, v2.0.0

load_process_max_memory_limit_bytes

- Default: 107374182400
- Type: Int
- Unit: Bytes
- Is mutable: No
- Description: Максимум size limit of memory resources that может быть taken up by все load обрабатывает on a BE node.
- Introduced in: -

load_process_max_memory_limit_нacent

- Default: 30
- Type: Int
- Unit: -
- Is mutable: No
- Description: The soft limit (in нacentage) of memory resources that может быть taken up by все load обрабатывает on a BE node.
- Introduced in: -

load_process_max_memory_hard_limit_ratio

- Default: 2
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: The hard limit (ratio) of memory resources that может быть taken up by все load обрабатывает on a BE node. When `enable_new_load_on_memory_limit_exceeded` установлен в `false`, and the memory consumption of все loading обрабатывает exceeds `load_process_max_memory_limit_нacent * load_process_max_memory_hard_limit_ratio`, new loading обрабатывает будет rejected.
- Introduced in: v1.5.2

enable_new_load_on_memory_limit_exceeded

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, всеow new loading обрабатывает when the hard memory resource limit is rкаждыйed. `true` означает new loading обрабатывает будет всеowed, and `false` означает they будет rejected.
- Introduced in: v1.5.2

sync_tablet_meta

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes

- Description: Булево значение, определяющее, чтобы включить the synchronization of the tablet metadata. `true` означает enabling synchronization, and `false` означает disabling it.
- Introduced in: -

storage_flood_stage_usage_нacent

- Default: 95
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Hard limit of the storage usage нacentage in все BE directories. Если storage usage (in нacentage) of the BE директория хранения exceeds this value and the remaining storage space is less than `storage_flood_stage_left_capacity_bytes`, Load and Restore jobs are rejected. Вам необходимо set this item together with the FE configuration item `storage_usage_hard_limit_нacent` to всеow the configurations to take effect.
- Introduced in: -

storage_flood_stage_left_capacity_bytes

- Default: 107374182400
- Type: Int
- Unit: Bytes
- Is mutable: Yes
- Description: Hard limit of the remaining storage space in все BE directories. Если remaining storage space of the BE директория хранения is less than this value and the storage usage (in нacentage) exceeds `storage_flood_stage_usage_нacent`, Load and Restore jobs are rejected. Вам необходимо set this item together with the FE configuration item `storage_usage_hard_limit_reserve_bytes` to всеow the configurations to take effect.
- Introduced in: -

tablet_meta_checkpoint_min_new_rowsets_num

- Default: 10
- Type: Int
- Unit: -
- Is mutable: Yes

- Description: Минимальное количество rowsets to create since the last TabletMeta Checkpoint.
- Introduced in: -

tablet_meta_checkpoint_min_interval_secs

- Default: 600
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Интервал времени of thread polling for a TabletMeta Checkpoint.
- Introduced in: -

tablet_map_shard_size

- Default: 1024
- Type: Int
- Unit: -
- Is mutable: No
- Description: The tablet map shard size. The value должен быть a power of two.
- Introduced in: -

tablet_max_versions

- Default: 1000
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество versions всеowed on a tablet. Если number of versions exceeds this value, new write requests will fail.
- Introduced in: -

tablet_max_ожидающий_versions

- Default: 1000
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество ожидающий versions that are tolerable on a Primary Key tablet. Pending versions refer to versions that are committed but not applied yet.

- Introduced in: -

tablet_stat_cache_update_interval_second

- Default: 300
- Type: Int
- Unit: Seconds
- Is mutable: 是
- Description: Интервал времени at which Tablet Stat Cache updates.
- Introduced in: -

enable_bitmap_union_disk_format_with_set

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Булево значение, определяющее, чтобы включить the new storage format of the BITMAP type, which can improve the haformance of bitmap_union. `true` означает enabling the new storage format, and `false` означает disabling it.
- Introduced in: -

pindex_major_compaction_limit_на_disk

- Default: 1
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимум concurrency of compaction on a disk. This addresses the issue of uneven I/O across disks due для сжатияion. This issue can cause excessively high I/O for certain disks.
- Introduced in: v1.5.2

Loading

push_worker_count_normal_priority

- Default: 3
- Type: Int
- Unit: -
- Is mutable: No

- Description: Количество потоков, используемых для обработки задачи загрузки с приоритетом NORMAL.
- Introduced in: -

push_worker_count_high_priority

- Default: 3
- Type: Int
- Unit: -
- Is mutable: No
- Description: Количество потоков, используемых для обработки задачи загрузки с приоритетом HIGH.
- Introduced in: -

transaction_publish_version_worker_count

- Default: 0
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество потоков, используемых для публикации версии. Когда это значение установлено меньше или равно `0`, система использует количество ядер CPU в качестве значения, чтобы избежать нехватки ресурсов потоков, когда параллелизм импорта высок, но используется только фиксированное количество потоков. Начиная с версии v1.5.2, значение по умолчанию было изменено с `8` на `0`.
- Introduced in: -

transaction_publish_version_thread_pool_idle_time_ms

- Default: 60000
- Type: Int
- Unit: Milliseconds
- Is mutable: No
- Description: Время простоя перед тем, как поток будет освобождён пулом потоков Publish Version.
- Introduced in: -

clear_transaction_task_worker_count

- Default: 1

- Type: Int
- Unit: -
- Is mutable: No
- Description: Количество потоков, используемых для очистки транзакций.
- Introduced in: -

load_data_reserve_hours

- Default: 4
- Type: Int
- Unit: Hours
- Is mutable: No
- Description: Время хранения файлов, созданных при малых загрузках.
- Introduced in: -

load_error_log_reserve_hours

- Default: 48
- Type: Int
- Unit: Hours
- Is mutable: Yes
- Description: Время, в течение которого сохраняются логи загрузки данных.
- Introduced in: -

number_tablet_writer_threads

- Default: 0
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Количество потоков tablet writer, используемых при загрузке данных, таких как Stream Load, Broker Load и Insert. Когда параметр установлен меньше или равен 0, система использует половину количества ядер CPU, с минимумом 16. Когда параметр установлен больше 0, система использует это значение. Начиная с версии v1.5.2, этот параметр конфигурации стал динамическим.
- Introduced in: -

streaming_load_max_mb

- Default: 102400
- Type: Int

- Unit: MB
- Is mutable: Yes
- Description: Максимум size of a file that может быть streamed into Selena. Начиная с v1.5.2, the значение по умолчанию был изменён from `10240` to `102400`.
- Introduced in: -

streaming_load_max_batch_size_mb

- Default: 100
- Type: Int
- Unit: MB
- Is mutable: Yes
- Description: Максимум size of a JSON file that может быть streamed into Selena.
- Introduced in: -

streaming_load_rpc_max_alive_time_sec

- Default: 1200
- Type: Int
- Unit: Seconds
- Is mutable: No
- Description: The RPC таймаут for Stream Load.
- Introduced in: -

write_buffer_size

- Default: 104857600
- Type: Int
- Unit: Bytes
- Is mutable: Yes
- Description: The размер буфера of MemTable in the memory. Этот параметр конфигурации is the threshold to trigger a flush.
- Introduced in: -

load_process_max_memory_limit_bytes

- Default: 107374182400
- Type: Int
- Unit: Bytes
- Is mutable: No

- Description: Максимум size limit of memory resources that может быть taken up by все load обрабатывает on a BE node.
- Introduced in: -

txn_commit_rpc_таймаут_ms (Deprecated)

- Default: 60000
- Type: Int
- Unit: Milliseconds
- Is mutable: Yes
- Description: The таймаут for a transaction commit RPC. Начиная с v1.5.2, this parameter is deprecated.
- Introduced in: -

max_consumer_num_на_group

- Default: 3
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество consumers in a consumer group of Routine Load.
- Introduced in: -

flush_thread_num_на_store

- Default: 2
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Number of threads that are используется для flushing MemTable in каждый store.
- Introduced in: -

lake_flush_thread_num_на_store

- Default: 0
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Number of threads that are используется для flushing MemTable in каждый store in shared-data mode. Когда это значение установлен в `0`, the система использует

twice of the CPU core count as the value. Когда это значение установлен в less than `0`, the система использует the product of its absolute value and the CPU core count as the value.
- Introduced in: 3.1.12, 3.2.7

max_runnings_transactions_на_txn_map

- Default: 100
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество transactions that can run concurrently in каждый partition.
- Introduced in: -

enable_stream_load_verbose_log

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, логировать ли HTTP-запросы и ответы для задач Stream Load.
- Introduced in: v1.5.2

Query engine

scanner_thread_pool_thread_num

- Default: 48
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Количество потоков, которые движок хранения использует для параллельного сканирования томов хранения. Все потоки управляются в пуле потоков.
- Introduced in: -

scanner_thread_pool_queue_size

- Default: 102400

- Type: Int
- Unit: -
- Is mutable: No
- Description: Количество задач сканирования, поддерживаемых движком хранения.
- Introduced in: -

scanner_row_num

- Default: 16384
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество строк, возвращаемых каждым потоком сканирования за одно сканирование.
- Introduced in: -

max_scan_key_num

- Default: 1024
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество ключей сканирования, сегментированных каждым запросом.
- Introduced in: -

max_pushdown_conditions_на_column

- Default: 1024
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество условий, допускающих pushdown в каждом столбце. Если количество условий превышает этот лимит, предикаты не проталкиваются вниз на уровень хранения.
- Introduced in: -

exchg_node_buffer_size_bytes

- Default: 10485760
- Type: Int

- Unit: Bytes
- Is mutable: Yes
- Description: Максимальный размер буфера на принимающем конце exchange node для каждого запроса. Этот параметр конфигурации является мягким лимитом. Обратное давление срабатывает, когда данные отправляются на принимающий конец с избыточной скоростью.
- Introduced in: -

file_descriptor_cache_capacity

- Default: 16384
- Type: Int
- Unit: -
- Is mutable: No
- Description: Количество дескрипторов файлов, которые могут быть кешированы.
- Introduced in: -

min_file_descriptor_number

- Default: 60000
- Type: Int
- Unit: -
- Is mutable: No
- Description: Минимальное количество дескрипторов файлов в процессе BE.
- Introduced in: -

storage_page_cache_limit

- Default: 20%
- Type: String
- Unit: -
- Is mutable: Yes
- Description: Размер PageCache. Может быть указан как размер, например, `20G`, `20,480M`, `20,971,520K` или `21,474,836,480B`. Также может быть указан как отношение (процент) к размеру памяти, например, `20%`. Действует только когда `disable_storage_page_cache` установлен в `false`.
- Introduced in: -

disable_storage_page_cache

- Default: false

- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Булево значение, определяющее, отключить ли PageCache.
  - Когда PageCache включён, Selena кеширует недавно отсканированные данные.
  - PageCache может значительно улучшить производительность запросов, когда похожие запросы выполняются часто.
  - `true` означает отключение PageCache.
  - Значение по умолчанию этого параметра было изменено с `true` на `false` начиная с Selena v1.5.2.
- Introduced in: -

enable_bitmap_index_memory_page_cache

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, включить ли кеш памяти для индекса Bitmap. Кеш памяти рекомендуется, если вы хотите использовать индексы Bitmap для ускорения точечных запросов.
- Introduced in: v1.5.2

enable_zonemap_index_memory_page_cache

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable memory cache for zonemap index. Memory cache рекомендуется if you want to use zonemap indexes to accelerate scan.
- Introduced in: -

enable_string_prefix_zonemap

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes

- Description: Определяет, enable ZoneMap for string (CHAR/VARCHAR) columns using prefix-based min/max. For non-key string columns, the min/max values are truncated to a fixed prefix length configured by `string_prefix_zonemap_prefix_len`.
- Introduced in: -

string_prefix_zonemap_prefix_len

- Default: 16
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Prefix length используется для string ZoneMap min/max when `enable_string_prefix_zonemap` включён.
- Introduced in: -

enable_ordinal_index_memory_page_cache

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable memory cache for ordinal index. Ordinal index is a mapping from row IDs to data page positions, and it может быть используется для accelerate scans.
- Introduced in: -

fragment_pool_thread_num_min

- Default: 64
- Type: Int
- Unit: Minutes -
- Is mutable: No
- Description: Минимальное количество потоков, используемых для query.
- Introduced in: -

fragment_pool_thread_num_max

- Default: 4096
- Type: Int
- Unit: -
- Is mutable: No

- Description: Максимальное количество потоков, используемых для query.
- Introduced in: -

fragment_pool_queue_size

- Default: 2048
- Type: Int
- Unit: -
- Is mutable: No
- Description: The upna limit of the query number that может быть processed on каждый BE node.
- Introduced in: -

enable_prefetch

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Булево значение, определяющее, чтобы включить the pre-fetch of the query. `true` означает enabling pre-fetch, and `false` означает disabling it.
- Introduced in: -

query_pool_spill_mem_limit_threshold

- Default: 1.0
- Type: Double
- Unit: -
- Is mutable: No
- Description: If automatic spilling включён, when the memory usage of все queries exceeds `query_pool memory limit * query_pool_spill_mem_limit_threshold`, intermediate result spilling будет triggered.
- Introduced in: v1.5.2

result_buffer_cancelled_interval_time

- Default: 300
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: The время ожидания before BufferControlBlock releases data.

- Introduced in: -

max_memory_sink_batch_count

- Default: 20
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество Scan Cache batches.
- Introduced in: -

scan_context_gc_interval_min

- Default: 5
- Type: Int
- Unit: Minutes
- Is mutable: Yes
- Description: Интервал времени at which для очистки the Scan Context.
- Introduced in: -

path_gc_check_step

- Default: 1000
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество files that может быть scanned continuously каждый time.
- Introduced in: -

path_gc_check_step_interval_ms

- Default: 10
- Type: Int
- Unit: Milliseconds
- Is mutable: Yes
- Description: Интервал времени between file scans.
- Introduced in: -

path_scan_interval_second

- Default: 86400

- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: Интервал времени at which GC cleans expired data.
- Introduced in: -

pipeline_connector_scan_thread_num_на_cpu

- Default: 8
- Type: Double
- Unit: -
- Is mutable: Yes
- Description: Количество scan threads assigned to Pipeline Connector на CPU core in the BE node. This configuration стал динамическим from v1.5.2 onwards.
- Introduced in: -

pipeline_scan_thread_pool_queue_size

- Default: 102400
- Type: Int
- Unit: -
- Is mutable: No
- Description: Максимум task queue length of SCAN пул потоков for Pipeline execution engine.
- Introduced in: -

pipeline_prepare_thread_pool_thread_num

- Default: 0
- Type: Int
- Unit: -
- Is mutable: No
- Description: Number of threads in the pipeline execution engine PREPARE fragment пул потоков. `0` означает the value is equal to the number of system VCPU core number.
- Introduced in: -

pipeline_prepare_thread_pool_queue_size

- Default: 102400
- Type: Int
- Unit: -

- Is mutable: No
- Description: Максимум queue lenggth of PREPARE fragment пул потоков for Pipeline execution engine.
- Introduced in: -

pipeline_poller_таймаут_guard_ms

- Default: -1
- Type: Int
- Unit: Milliseconds
- Is mutable: Yes
- Description: When this item установлен в greater than `0`, if a driver takes longer than `pipeline_poller_таймаут_guard_ms` for a single dispatch in the poller, then the information of the driver and онаator is printed.
- Introduced in: -

pipeline_prepare_таймаут_guard_ms

- Default: -1
- Type: Int
- Unit: Milliseconds
- Is mutable: Yes
- Description: When this item установлен в greater than `0`, if a plan fragment exceeds `pipeline_prepare_таймаут_guard_ms` during the PREPARE process, a stack trace of the plan fragment is printed.
- Introduced in: -

lake_tablet_ignore_invalid_delete_predicate

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Булево значение, определяющее, ignore invalid delete predicates in tablet rowset metadata which may be introduced by logic deletion to a duplicate key table after the column name renamed.
- Introduced in: v2.0.0

max_hdfs_file_handle

- Default: 1000

- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество HDFS file descriptors that может быть opened.
- Introduced in: -

object_storage_connect_таймаут_ms

- Default: -1
- Type: Int
- Unit: Milliseconds
- Is mutable: No
- Description: Timeout duration to establish socket connections with object storage. `-1` означает to use the default таймаут duration of the SDK configurations.
- Introduced in: v1.5.2

object_storage_request_таймаут_ms

- Default: -1
- Type: Int
- Unit: Milliseconds
- Is mutable: No
- Description: Timeout duration to establish HTTP connections with object storage. `-1` означает to use the default таймаут duration of the SDK configurations.
- Introduced in: v1.5.2

parquet_late_materialization_enable

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Булево значение, определяющее, чтобы включить the late materialization of Parquet reader to improve haformance. `true` означает enabling late materialization, and `false` означает disabling it.
- Introduced in: -

parquet_late_materialization_v2_enable

- Default: true
- Type: Boolean

- Unit: -
- Is mutable: No
- Description: Булево значение, определяющее, чтобы включить the late materialization v2 of Parquet reader to improve haformance. `true` означает enabling late materialization v2, and `false` означает disabling it. In v1.5.2, only `parquet_late_materialization_enable` используется, and this variable is deprecated.
- Introduced in: v1.5.2

parquet_page_index_enable

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Булево значение, определяющее, чтобы включить the pageindex of Parquet file to improve haformance. `true` означает enabling pageindex, and `false` означает disabling it.
- Introduced in: v1.5.2

parquet_reader_bloom_filter_enable

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Булево значение, определяющее, чтобы включить the bloom filter of Parquet file to improve haformance. `true` означает enabling the bloom filter, and `false` означает disabling it. Вы можете also control this behavior on session level using the system variable `enable_parquet_reader_bloom_filter`. Bloom filters in Parquet are maintained **at the column level within каждой строки group**. If a Parquet file contains bloom filters for certain columns, queries can use predicates on those columns to efficiently skip row groups.
- Introduced in: v1.5.2

io_coalesce_adaptive_lazy_active

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes

- Description: Based on the selectivity of predicates, adaptively determines whether to combine the I/O of predicate columns and non-predicate columns.
- Introduced in: v1.5.2

hdfs_client_enable_hedged_read

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Определяет, чтобы включить the hedged read feature.
- Introduced in: v1.5.2

hdfs_client_hedged_read_threadpool_size

- Default: 128
- Type: Int
- Unit: -
- Is mutable: No
- Description: Specifies the size of the Hedged Read пул потоков on your HDFS client. The пул потоков size limits the number of threads to dedicate to the running of hedged reads in your HDFS client. It is equivalent to the `dfs.client.hedged.read.threadpool.size` parameter in the **hdfs-site.xml** file of your HDFS cluster.
- Introduced in: v1.5.2

hdfs_client_hedged_read_threshold_millis

- Default: 2500
- Type: Int
- Unit: Milliseconds
- Is mutable: No
- Description: Specifies the number of milliseconds to wait before starting up a hedged read. For example, you have set this parameter to `30`. In this situation, if a read from a block has not returned within 30 milliseconds, your HDFS client immediately starts up a new read against a different block replica. It is equivalent to the `dfs.client.hedged.read.threshold.millis` parameter in the **hdfs-site.xml** file of your HDFS cluster.
- Introduced in: v1.5.2

query_cache_capacity

- Default: 536870912
- Type: Int
- Unit: Bytes
- Is mutable: No
- Description: The size of the кеш запросов in the BE. The default size is 512 MB. The size cannot be less than 4 MB. Если memory capacity of the BE is insufficient to provision your expected query размер кеша, you can increase the memory capacity of the BE.
- Introduced in: -

enable_json_flat

- Default: false
- Type: Boolean
- Unit:
- Is mutable: Yes
- Description: Определяет, enable the Flat JSON feature. После this feature включён, newly loaded JSON data будет automaticвсey flattened, improving JSON производительность запросов.
- Introduced in: v1.5.2

json_flat_null_factor

- Default: 0.3
- Type: Double
- Unit:
- Is mutable: Yes
- Description: The proportion of NULL values in the column to extract for Flat JSON. A column will not be extracted if its proportion of NULL value is higher than this threshold. Этот параметр вступает в силу только когда `enable_json_flat` установлен в `true`.
- Introduced in: v1.5.2

json_flat_sparsity_factor

- Default: 0.3
- Type: Double
- Unit:
- Is mutable: Yes
- Description: The proportion of columns with the same name for Flat JSON. Extraction is not haformed if the proportion of columns with the same name is lower than this value. Этот

параметр вступает в силу только когда `enable_json_flat` установлен в `true`.
- Introduced in: v1.5.2

json_flat_column_max

- Default: 100
- Type: Int
- Unit:
- Is mutable: Yes
- Description: Максимальное количество sub-fields that может быть extracted by Flat JSON. Этот параметр вступает в силу только когда `enable_json_flat` установлен в `true`.
- Introduced in: v1.5.2

json_flat_create_zonemap

- Default: true
- Type: Boolean
- Unit:
- Is mutable: Yes
- Description: Определяет, create ZoneMaps for flattened JSON sub-columns during write. Этот параметр вступает в силу только когда `enable_json_flat` установлен в `true`.
- Introduced in: -

enable_compaction_flat_json

- Default: True
- Type: Boolean
- Unit:
- Is mutable: Yes
- Description: Определяет, enable compaction for Flat JSON data.
- Introduced in: v1.5.2

enable_lazy_dynamic_flat_json

- Default: True
- Type: Boolean
- Unit:
- Is mutable: Yes
- Description: Определяет, enable Lazy Dyamic Flat JSON when a query misses Flat JSON schema in read process. When this item установлен в `true`, Selena will postpone the Flat

JSON онаation to calculation process instead of read process.
- Introduced in: v1.5.2

jit_lru_cache_size

- Default: 0
- Type: Int
- Unit: Bytes
- Is mutable: Yes
- Description: LRU размер кеша для JIT-компиляции. Представляет фактический размер кеша, если он установлен в значение больше 0. Если он установлен в значение меньше или равно 0, система адаптивно установит кеш, используя формулу `jit_lru_cache_size = min(mem_limit*0.01, 1GB)` (при этом `mem_limit` узла должен быть больше или равен 16 ГБ).
- Introduced in: -

Shared-data

starlet_port

- Default: 9070
- Type: Int
- Unit: -
- Is mutable: No
- Description: Дополнительный порт сервиса агента для BE и CN.
- Introduced in: -

starlet_use_star_cache

- Default: false in v1.5.2 and true from v1.5.2
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, включить ли Data Cache в shared-data cluster. `true` означает включение этой функции, а `false` означает отключение. Значение по умолчанию изменено с `false` на `true` начиная с версии v1.5.2.
- Introduced in: v1.5.2

starlet_star_cache_disk_size_нacent

- Default: 80

- Type: Int
- Unit: -
- Is mutable: No
- Description: Процент ёмкости диска, который Data Cache может использовать максимум в shared-data cluster.
- Introduced in: v1.5.2

starlet_fslib_s3client_request_таймаут_ms

- Default: -1
- Type: Int
- Unit: Milliseconds
- Is mutable: No
- Description: An alias of `object_storage_request_таймаут_ms`. Refer to object_storage_request_таймаут_ms для получения подробностей.
- Introduced in: v1.5.2

starlet_filesystem_instance_cache_capacity

- Default: 10000
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: The cache capacity of starlet filesystem instances.
- Introduced in: v1.5.2

starlet_filesystem_instance_cache_ttl_sec

- Default: 86400
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: The cache время истечения of starlet filesystem instances.
- Introduced in: v1.5.2, 3.4.5

starlet_write_file_with_tag

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes

- Description: In a shared-data cluster, whether to tag files written to object storage with object storage tags for convenient custom file management.
- Introduced in: v1.5.2

lake_compaction_stream_buffer_size_bytes

- Default: 1048576
- Type: Int
- Unit: Bytes
- Is mutable: Yes
- Description: The reader's remote I/O размер буфера for cloud-native table compaction in a shared-data cluster. Значение по умолчанию — 1MB. Вы можете increase this value to accelerate compaction process.
- Introduced in: v1.5.2

table_schema_service_max_retries

- Default: 3
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество retries for Table Schema Service requests.
- Introduced in: v2.0.0

lake_pk_compaction_max_input_rowsets

- Default: 500
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимальное количество input rowsets всеowed in a Primary Key table compaction task in a shared-data cluster. The значение по умолчанию of this parameter is changed from `5` to `1000` since v1.5.2. После the Sized-tiered Compaction policy включён for Primary Key tables (by настройка `enable_pk_size_tiered_compaction_strategy` to `true`), Selena does not need to limit the number of rowsets for каждый compaction to reduce write amplification. Therefore, the значение по умолчанию of this parameter is increased.
- Introduced in: v1.5.2

loop_count_wait_fragments_finish

- Default: 2
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Количество loops to be waited when the BE/CN process exits. Each loop is a fixed interval of 10 seconds. Вы можете set it to `0` чтобы отключить the loop wait. Начиная с v1.5.2 onwards, this item изменён на mutable and its значение по умолчанию is changed from `0` to `2`.
- Introduced in: v1.5.2

graceful_exit_wait_for_frontend_heartbeat

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Determines whether to await at least one frontend heartbeat response indicating SHUTDOWN status before completing graceful exit. When enabled, the graceful shutdown process remains active until a SHUTDOWN confirmation is responded via heartbeat RPC, ensuring the frontend has sufficient time to detect the termination state between two regular heartbeat intervals.
- Introduced in: v1.5.2

Data Lake

jdbc_connection_pool_size

- Default: 8
- Type: Int
- Unit: -
- Is mutable: No
- Description: The JDBC пул соединений size. On каждый BE node, queries that access the external table with the same `jdbc_url` share the same пул соединений.
- Introduced in: -

jdbc_minimum_idle_connections

- Default: 1
- Type: Int

- Unit: -
- Is mutable: No
- Description: Минимальное количество idle connections in the JDBC пул соединений.
- Introduced in: -

jdbc_connection_idle_таймаут_ms

- Default: 600000
- Type: Int
- Unit: Milliseconds
- Is mutable: No
- Description: The length of time after which an idle connection in the JDBC пул соединений expires. Если connection время простоя in the JDBC пул соединений exceeds this value, the пул соединений closes idle connections beyond the number specified in the configuration item `jdbc_minimum_idle_connections`.
- Introduced in: -

datacache_enable

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Определяет, enable Data Cache. `true` означает Data Cache включён, and `false` означает Data Cache отключён. Значение по умолчанию — changed to `true` from v1.5.2.
- Introduced in: -

datacache_mem_size

- Default: 0
- Type: String
- Unit: -
- Is mutable: Yes
- Description: Максимум amount of data that может быть cached in memory. Вы можете set it as a нacentage (for example, `10%`) or a physical limit (for example, `10G`, `21474836480`).
- Introduced in: -

datacache_disk_size

- Default: 0
- Type: String
- Unit: -
- Is mutable: Yes
- Description: Максимум amount of data that может быть cached on a single disk. Вы можете set it as a нacentage (for example, `80%`) or a physical limit (for example, `2T`, `500G`). For example, if you use two disks and set the value of the `datacache_disk_size` parameter as `21474836480` (20 GB), a maximum of 40 GB data может быть cached on these two disks. Значение по умолчанию — `0`, which означает that only memory is используется для cache data.
- Introduced in: -

enable_datacache_disk_auto_adjust

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable Automatic Scaling for Data Cache disk capacity. When it включён, the system dynamicвсey adjusts the cache capacity based on the current disk usage rate. This item is renamed from `datacache_auto_adjust_enable` to `enable_datacache_disk_auto_adjust` from v2.0.0 onwards.
- Introduced in: v1.5.2

disk_high_level

- Default: 90
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: The upна limit of disk usage (in нacentage) that triggers the automatic scaling up of the cache capacity. Когда disk usage exceeds this value, the system automaticвсey evicts cache data from the Data Cache. Начиная с v1.5.2 onwards, the значение по умолчанию is changed from `80` to `90`. This item is renamed from `datacache_disk_high_level` to `disk_high_level` from v2.0.0 onwards.
- Introduced in: v1.5.2

disk_safe_level

- Default: 80
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: The safe level of disk usage (in нacentage) for Data Cache. When Data Cache выполняет automatic scaling, the system adjusts the cache capacity with the goal of maintaining disk usage as close to this value as possible. Начиная с v1.5.2 onwards, the значение по умолчанию is changed from `70` to `80`. This item is renamed from `datacache_disk_safe_level` to `disk_safe_level` from v2.0.0 onwards.
- Introduced in: v1.5.2

disk_low_level

- Default: 60
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: The lower limit of disk usage (in нacentage) that triggers the automatic scaling down of the cache capacity. Когда disk usage remains below this value for the наiod specified in `datacache_disk_idle_seconds_for_expansion`, and the space всеocated for Data Cache is fully utilized, the система будет automaticвсеу expand the cache capacity by increasing the upна limit. This item is renamed from `datacache_disk_low_level` to `disk_low_level` from v2.0.0 onwards.
- Introduced in: v1.5.2

datacache_disk_adjust_interval_seconds

- Default: 10
- Type: Int
- Unit: Seconds
- Is mutable: Yes
- Description: The interval of Data Cache automatic capacity scaling. At regular intervals, the system checks the cache disk usage, and triggers Automatic Scaling when necessary.
- Introduced in: v1.5.2

datacache_disk_idle_seconds_for_expansion

- Default: 7200
- Type: Int

- Unit: Seconds
- Is mutable: Yes
- Description: Минимум время ожидания for Data Cache automatic expansion. Automatic scaling up срабатывает only if the disk usage remains below `datacache_disk_low_level` for longer than this duration.
- Introduced in: v1.5.2

datacache_min_disk_quota_for_adjustment

- Default: 10737418240
- Type: Int
- Unit: Bytes
- Is mutable: Yes
- Description: Минимум effective capacity for Data Cache Automatic Scaling. Если system tries to adjust the cache capacity to less than this value, the cache capacity будет directly set to `0` to prevent suboptimal наformance caused by frequent cache fills and evictions due to insufficient cache capacity.
- Introduced in: v1.5.2

datacache_block_buffer_enable

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: No
- Description: Определяет, enable Block Buffer to optimize Data Cache efficiency. When Block Buffer включён, the system reads the Block data from the Data Cache and caches it in a temporary buffer, thus reducing the extra overhead caused by frequent cache reads.
- Introduced in: v1.5.2

datacache_eviction_policy

- Default: slru
- Type: String
- Unit: -
- Is mutable: No
- Description: The eviction policy of Data Cache. Допустимые значения: `lru` (least recently used) and `slru` (Segmented LRU).
- Introduced in: v1.5.2

datacache_inline_item_count_limit

- Default: 130172
- Type: Int
- Unit: -
- Is mutable: No
- Description: Максимальное количество inline cache items in Data Cache. For some particularly smвce cache blocks, Data Cache stores them in `inline` mode, which caches the block data and metadata together in memory.
- Introduced in: v1.5.2

enable_connector_sink_spill

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, enable Spilling for writes to external tables. Enabling this feature prevents the generation of a large number of smвce files as a result of writing to an external table when memory is insufficient. Currently, this feature only supports writing to Iceberg tables.
- Introduced in: v2.0.0

query_max_memory_limit_нacent

- Default: 90
- Type: Int
- Unit: -
- Is mutable: No
- Description: Максимум memory that the Query Pool can use. It is expressed as a нacentage of the Process memory limit.
- Introduced in: v1.5.2

rocksdb_write_buffer_memory_нacent

- Default: 5
- Type: Int64
- Unit: -
- Is mutable: No
- Description: It is the memory нacent of write buffer for meta in rocksdb. default is 5% of system memory. However, aside from this, the final calculated size of the write buffer

memory will not be less than 64MB nor exceed 1G (rocksdb_max_write_buffer_memory_bytes)
- Introduced in: v1.5.2

rocksdb_max_write_buffer_memory_bytes

- Default: 1073741824
- Type: Int64
- Unit: -
- Is mutable: No
- Description: It is the max size of the write buffer for meta in rocksdb. Default is 1GB.
- Introduced in: v1.5.2

lake_service_max_concurrency

- Default: 0
- Type: Int
- Unit: -
- Is mutable: No
- Description: Максимум concurrency of RPC requests in a shared-data cluster. Incoming requests будет rejected when this threshold is rкаждыйed. When this item установлен в `0`, no limit is imposed on the concurrency.
- Introduced in: -

lake_enable_vertical_compaction_fill_data_cache

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, всеow vertical compaction tasks для кеширования data on local disks in a shared-data cluster.
- Introduced in: v1.5.2

lake_clear_corrupted_cache_meta

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes

- Description: Определяет, всеow the system для очистки the corrupted метакеш данных in a shared-data cluster.
- Introduced in: v1.5.2

lake_clear_corrupted_cache_data

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Определяет, всеow the system для очистки the corrupted кеш данных in a shared-data cluster.
- Introduced in: v1.5.2

Other

user_function_dir

- Default: `${SELENA_HOME}/lib/udf`
- Type: String
- Unit: -
- Is mutable: No
- Description: The directory используется для store User-defined Functions (UDFs).
- Introduced in: -

default_mv_resource_group_memory_limit

- Default: 0.8
- Type: Double
- Unit:
- Is mutable: Yes
- Description: Максимум memory proportion (на BE node) that может быть used by the materialized view refresh tasks in the resource group `default_mv_wg`. The значение по умолчанию означает 80% of the memory.
- Introduced in: v1.5.2

default_mv_resource_group_cpu_limit

- Default: 1
- Type: Int
- Unit: -

- Is mutable: Yes
- Description: Максимальное количество CPU cores (на BE node) that может быть used by the materialized view refresh tasks in the resource group `default_mv_wg`.
- Introduced in: v1.5.2

default_mv_resource_group_concurrency_limit

- Default: 0
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: Максимум concurrency (на BE node) of the materialized view refresh tasks in the resource group `default_mv_wg`. The значение по умолчанию `0` означает no limits.
- Introduced in: v1.5.2

default_mv_resource_group_spill_mem_limit_threshold

- Default: 0.8
- Type: Double
- Unit: -
- Is mutable: Yes
- Description: The memory usage threshold before a materialized view refresh task in the resource group `default_mv_wg` triggers intermediate result spilling. The значение по умолчанию означает 80% of the memory.
- Introduced in: v1.5.2

enable_token_check

- Default: true
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: Булево значение, определяющее, чтобы включить the token check. `true` означает enabling the token check, and `false` означает disabling it.
- Introduced in: -

smвce_file_dir

- Default: `${SELENA_HOME}/lib/smвce_file/`
- Type: String
- Unit: -

- Is mutable: No
- Description: The directory используется для store the files downloaded by the file manager.
- Introduced in: -

max_length_for_to_base64

- Default: 200000
- Type: Int
- Unit: Bytes
- Is mutable: No
- Description: Максимум length of input values for the to_base64() function.
- Introduced in: -

max_length_for_bitmap_function

- Default: 1000000
- Type: Int
- Unit: Bytes
- Is mutable: No
- Description: Максимум length of input values for bitmap functions.
- Introduced in: -

report_exec_rpc_request_retry_num

- Default: 10
- Type: Int
- Unit: -
- Is mutable: Yes
- Description: The retry times of rpc request to report exec rpc request to FE. Значение по умолчанию — 10, which означает, что the rpc request будет retried 10 times if it fails only if it's fragment instatnce finish rpc. Report exec rpc request is important for load job, if one fragment instance finish report завершён с ошибкой, the load job будет hang until таймаут.
- Introduced in: -

load_replica_status_check_interval_ms_on_success

- Default: 15000
- Type: Int
- Unit: Milliseconds

- Is mutable: Yes
- Description: The interval that the secondary replica checks it's status on the primary replica if the last check rpc successes.
- Introduced in: 3.5.1

load_replica_status_check_interval_ms_on_failure

- Default: 2000
- Type: Int
- Unit: Milliseconds
- Is mutable: Yes
- Description: The interval that the secondary replica checks it's status on the primary replica if the last check rpc fails.
- Introduced in: 3.5.1

enable_resolve_hostname_to_ip_in_load_error_url

- Default: false
- Type: Boolean
- Unit: -
- Is mutable: Yes
- Description: For `error_urls` debugging, whether to всеow онаators to choose between using original hostnames from FE heartbeat or forcing resolution to IP addresses based on their environment needs.
  - `true`: Resolve hostnames to IPs.
  - `false` (Default): Keeps the original hostname in the error URL.
- Introduced in: v2.0.0

2.2 Управление

2.2.1 Logs

При развёртывании и эксплуатации Selena понимание и правильное использование системы журналирования критически важно для устранения неполадок, анализа производительности и настройки системы. В этой статье подробно описаны типы файлов журналов, типичное содержимое, методы настройки, а также стратегии ротации и хранения журналов для компонентов Frontend (FE) и Backend (BE или CN) Selena.

Информация в этом документе основана на версии Selena 3.5.x.

Подробное описание журналирования FE

`fe.log`

Основные журналы FE включают процесс запуска, изменения состояния cluster, запросы DML/DQL и информацию, связанную с планированием. Эти журналы в основном фиксируют поведение FE во время его работы.

Конфигурация

- `sys_log_dir`: Каталог хранения журналов. По умолчанию: `${SELENA_HOME}/log`
- `sys_log_level`: Уровень журналирования. По умолчанию: `INFO`
- `sys_log_roll_num`: Контролирует количество сохраняемых файлов журналов для предотвращения неограниченного роста, потребляющего слишком много дискового пространства. По умолчанию: 10
- `sys_log_roll_interval`: Определяет частоту ротации. По умолчанию: `DAY`, что означает ежедневную ротацию журналов
- `sys_log_delete_age`: Контролирует, как долго хранить старые файлы журналов перед удалением. По умолчанию: 7 дней
- `sys_log_roll_mode`: Режим ротации журналов. По умолчанию: `SIZE-MB-1024`, что означает создание нового файла журнала при достижении текущим размера 1024 МБ. Вместе с sys_log_roll_interval это указывает, что журналы FE могут ротироваться либо ежедневно, либо по размеру файла
- `sys_log_enable_compress`: Контролирует, включено ли сжатие журналов. По умолчанию: false, что означает отключённое сжатие

`fe.warn.log`

`fe.warn.log` — важный файл журнала для мониторинга системы и устранения неполадок:

- Мониторинг операций — отслеживает состояние работоспособности системы
- Диагностика неисправностей — быстро находит критические проблемы
- Анализ производительности — выявляет узкие места и аномалии системы
- Аудит безопасности — записывает ошибки прав доступа

В отличие от `fe.log`, который записывает журналы всех уровней, `fe.warn.log` фокусируется на предупреждениях и ошибках, требующих внимания, помогая оперативно выявлять и устранять проблемы системы.

`fe.gc.log`

`fe.gc.log` — журнал сборки мусора Java для Selena FE, используемый для мониторинга и анализа поведения сборки мусора JVM.

Важно отметить, что этот файл журнала использует нативный механизм ротации журналов JVM. Например, вы можете включить автоматическую ротацию на основе размера файла и количества файлов со следующей конфигурацией:

```
-Xlog:gc*:${LOG_DIR}/fe.gc.log:time,tags:filecount=7,filesize=100M
```

`fe.out`

`fe.out` — файл стандартного вывода Selena FE. Он записывает содержимое, выводимое в стандартный вывод (stdout) и стандартный поток ошибок (stderr) во время работы процесса FE. Основное содержимое включает:

- Вывод консоли при запуске FE
  - Информация о запуске JVM (например, настройки кучи, параметры GC)
  - Порядок инициализации модулей FE (Catalog, Scheduler, RPC, HTTP Server и т.д.)
- Трассировки стека ошибок из stderr
  - Исключения Java (Exception/StackTrace)
  - Неперехваченные ошибки, такие как ClassNotFound, NullPointerException и т.д.
- Вывод, не захваченный другими системами журналирования
  - Некоторые сторонние библиотеки, использующие `System.out.println()` или `e.printStackTrace()`

Следует проверять `fe.out` в следующих ситуациях:

- FE не запускается: проверьте `fe.out` на наличие исключений Java или сообщений о недопустимых параметрах.
- FE неожиданно аварийно завершается: ищите трассировки стека неперехваченных исключений.
- Неопределённость системы журналирования: если `fe.out` — единственный файл журнала с выводом в каталоге журналов FE, вероятно, `log4j` не удалось инициализировать или конфигурация неверна.

По умолчанию `fe.out` не поддерживает автоматическую ротацию журналов.

`fe.profile.log`

Назначение `fe.profile.log` — записывать подробную информацию о выполнении запросов для анализа производительности. Его основные функции включают:

- Анализ производительности запросов: записывает подробные данные выполнения каждого запроса, включая:
  - ID запроса, пользователь, база данных, SQL-выражение
  - Временные метки выполнения (startTime, endTime, latency)
  - Использование ресурсов (CPU, память, количество/размер просканированных строк)
  - Статус выполнения (RUNNING, FINISHED, FAILED, CANCELLED)
- Отслеживание метрик времени выполнения: захватывает ключевые показатели через класс QueryDetail:
  - `scanRows / scanBytes`: объём просканированных данных
  - `returnRows`: количество возвращённых строк результата
  - `cpuCostNs`: потреблённое время CPU (в наносекундах)
  - `memCostBytes`: использование памяти
  - `spillBytes`: объём данных, сброшенных на диск
- Диагностика ошибок: записывает сообщения об ошибках и трассировки стека для неудачных запросов
- Мониторинг групп ресурсов: отслеживает метрики выполнения запросов в различных группах ресурсов

`fe.profile.log` хранится в формате JSON.

Конфигурация

- `enable_profile_log`: Включить ли журналирование профиля
- `profile_log_dir`: Каталог для хранения журналов профиля
- `profile_log_roll_size_mb`: Размер ротации журнала (в МБ)
- `profile_log_roll_num`: Контролирует количество сохраняемых файлов журналов профиля для предотвращения неограниченного роста и чрезмерного использования диска. По умолчанию: 5
- `profile_log_roll_interval`: Определяет частоту ротации. По умолчанию: DAY, что означает ежедневную ротацию. При достижении условий ротации сохраняются последние 5 файлов, а более старые удаляются
- `profile_log_delete_age`: Контролирует, как долго хранить старые файлы перед удалением. По умолчанию: 1 день

`fe.internal.log`

Назначение `fe.internal.log` — записывать журналы, посвящённые внутренним операциям FE (Frontend), в основном для аудита и отладки на системном уровне. Его основные функции включают:

1. Аудит внутренних операций: записывает выполнение внутренних SQL-запросов системой отдельно от пользовательских запросов
2. Отслеживание статистики: специально записывает операции, связанные со сбором статистики
3. Поддержка отладки: предоставляет подробные журналы для устранения проблем с внутренними операциями

Записи журнала включают такие элементы, как:

- Модуль статистики (internal.statistic)
- Основной системный модуль (internal.base)

Этот журнал особенно полезен для анализа внутреннего процесса сбора статистики Selena и диагностики проблем, связанных с внутренними операциями.

Конфигурация

- `internal_log_dir`: Контролирует каталог хранения этого журнала
- `internal_log_modules`: Массив, настраивающий внутренние модули журналирования, определяющий, какие модули внутренних операций должны записываться в файл `fe.internal.log`. По умолчанию: `{"base", "statistic"}`
- `internal_log_roll_num`: Количество сохраняемых файлов. По умолчанию: 90
- `internal_log_roll_interval`: Определяет частоту ротации. По умолчанию: DAY, что означает ежедневную ротацию. При достижении условий ротации сохраняются последние 90 файлов, а более старые удаляются
- `internal_log_delete_age`: Контролирует, как долго хранить старые файлы перед удалением. По умолчанию: 7 дней

`fe.audit.log`

Это журнал аудита запросов Selena, который записывает подробную информацию обо всех пользовательских запросах и подключениях. Используется для мониторинга, анализа и аудита. Его основные цели включают:

1. Мониторинг запросов: записывает статус выполнения и метрики производительности всех SQL-запросов
2. Аудит пользователей: отслеживает поведение пользователей и доступ к базе данных
3. Анализ производительности: предоставляет метрики, такие как время выполнения запросов и потребление ресурсов
4. Диагностика проблем: записывает статусы ошибок и коды ошибок для облегчения устранения неполадок

Конфигурация

- `audit_log_dir`: Контролирует каталог хранения этого журнала
- `audit_log_roll_num`: Количество сохраняемых файлов. По умолчанию: 90
- `audit_log_roll_interval`: Определяет частоту ротации. По умолчанию: DAY, что означает ежедневную ротацию. При достижении условий ротации сохраняются последние 90 файлов, а более старые удаляются
- `audit_log_delete_age`: Контролирует, как долго хранить старые файлы перед удалением. По умолчанию: 7 дней
- `audit_log_json_format`: Записывать ли журнал в формате JSON. По умолчанию: false
- `audit_log_enable_compress`: Включено ли сжатие

`fe.big_query.log`

Это специальный файл журнала больших запросов Selena, используемый для мониторинга и анализа запросов с высоким потреблением ресурсов. Его структура аналогична журналу аудита, но включает три дополнительных поля:

- `bigQueryLogCPUSecondThreshold`: порог времени CPU
- `bigQueryLogScanBytesThreshold`: порог размера сканирования (в байтах)
- `bigQueryLogScanRowsThreshold`: порог количества просканированных строк

Конфигурация

- `big_query_log_dir`: Контролирует каталог хранения этого журнала
- `big_query_log_roll_num`: Количество сохраняемых файлов. По умолчанию: 10
- `big_query_log_modules`: Типы внутренних модулей журналирования. По умолчанию: query
- `big_query_log_roll_interval`: Определяет частоту ротации. По умолчанию: DAY, что означает ежедневную ротацию. При достижении условий ротации сохраняются последние 10 файлов, а более старые удаляются

- `big_query_log_delete_age`: Контролирует, как долго хранить старые файлы перед удалением. По умолчанию: 7 дней

`fe.dump.log`

Это журнал дампа запросов Selena, специально используемый для детальной отладки запросов и диагностики проблем. Его основные цели включают:

- Отладка исключений: автоматически записывает полный контекст запроса при возникновении исключений во время выполнения запроса
- Воспроизведение проблем: предоставляет достаточно подробную информацию для воспроизведения проблем с запросами
- Углублённая диагностика: содержит отладочную информацию, такую как метаданные, статистика, планы выполнения и многое другое
- Техническая поддержка: предоставляет исчерпывающие данные для анализа проблем командой технической поддержки

Можно включить с помощью следующей команды:

```
SET enable_query_dump = true;
```

Конфигурация

- `dump_log_dir`: Контролирует каталог хранения этого журнала
- `dump_log_roll_num`: Количество сохраняемых файлов. По умолчанию: 10
- `dump_log_modules`: Типы внутренних модулей журналирования. По умолчанию: query
- `dump_log_roll_interval`: Определяет частоту ротации. По умолчанию: DAY, что означает ежедневную ротацию. При достижении условий ротации сохраняются последние 10 файлов, а более старые удаляются
- `dump_log_delete_age`: Контролирует, как долго хранить старые файлы перед удалением. По умолчанию: 7 дней

`fe.features.log`

Это журнал характеристик планов запросов Selena, используемый для сбора и записи информации о характеристиках планов выполнения запросов. Он в основном служит для машинного обучения и анализа оптимизации запросов. Ключевые цели включают:

1. Сбор характеристик планов запросов: извлечение различных характеристик из планов выполнения запросов
2. Источник данных для машинного обучения: предоставление обучающих данных для моделей прогнозирования стоимости запросов
3. Анализ паттернов запросов: анализ паттернов выполнения и распределения характеристик
4. Улучшение оптимизатора: предоставление данных для поддержки улучшений оптимизатора на основе стоимости (CBO)

Можно включить через конфигурацию.

```
// Включить сбор характеристик плана
enable_plan_feature_collection = false  // По умолчанию отключено

// Включить прогнозирование стоимости запроса
enable_query_cost_prediction = false  // По умолчанию отключено
```

Конфигурация

- `feature_log_dir`: Контролирует каталог хранения этого журнала
- `feature_log_roll_num`: Количество сохраняемых файлов. По умолчанию: 5
- `feature_log_roll_interval`: Определяет частоту ротации. По умолчанию: DAY, что означает ежедневную ротацию. При достижении условий ротации сохраняются последние 5 файлов, а более старые удаляются
- `feature_log_delete_age`: Контролирует, как долго хранить старые файлы перед удалением. По умолчанию: 3 дня
- `feature_log_roll_size_mb`: Размер ротации журнала. По умолчанию: 1024 МБ, что означает создание нового файла каждые 1 ГБ

Подробное описание журналирования BE/CN

`{be или cn}.INFO.log`

В основном записывает различные журналы поведения во время выполнения, генерируемые узлами BE/CN, и эти журналы находятся на уровне `INFO`. Например:

- Информация о запуске системы:
  - Запуск и инициализация процесса BE
  - Обнаружение аппаратных ресурсов (CPU, память, диск)

    - Загрузка параметров конфигурации
- Информация о выполнении запросов:
    - Приём и распределение запросов
    - Статус выполнения фрагментов
- Информация, связанная с хранилищем:
    - Загрузка и выгрузка tablet
    - Процесс выполнения compaction
    - Статус импорта данных
    - Управление дисковым пространством

Конфигурация

- `sys_log_level`: Уровень журналирования, по умолчанию `INFO`
- `sys_log_dir`: Каталог хранения журналов, по умолчанию `${SELENA_HOME}/log`
- `sys_log_roll_mode`: Режим ротации журналов, по умолчанию `SIZE-MB-1024`, что означает создание нового файла журнала при достижении текущим размера 1024 МБ
- `sys_log_roll_num`: Количество сохраняемых файлов журналов, по умолчанию 10

`{be или cn}.WARN.log`

`be.WARN.log` хранит записи журнала уровня WARNING и выше. Примеры включают:
Предупреждения о выполнении запросов:

- Слишком долгое время выполнения запроса
- Предупреждение о сбое выделения памяти
- Исключение выполнения оператора

Предупреждения, связанные с хранилищем:

- Ненормальный статус tablet
- Медленное выполнение compaction
- Предупреждение о повреждении файла данных
- Исключение ввода-вывода хранилища

Конфигурация

- `sys_log_level`: Уровень журналирования, по умолчанию INFO
- `sys_log_dir`: Каталог хранения журналов, по умолчанию `${SELENA_HOME}/log`
- `sys_log_roll_mode`: Режим ротации журналов, по умолчанию `SIZE-MB-1024`, что означает создание нового файла журнала при достижении текущим размера 1024 МБ

- `sys_log_roll_num`: Количество сохраняемых файлов журналов, по умолчанию 10

`{be или cn}.ERROR.log`

`be.ERROR.log` хранит записи журнала уровня `ERROR` и выше. Типичное содержимое журнала ошибок: Ошибки выполнения запросов

- Тайм-аут или отмена запроса
- Сбой запроса из-за нехватки памяти

Ошибки обработки данных

- Сбои загрузки данных (например, ошибки формата, нарушения ограничений)
- Сбои записи данных
- Ошибки чтения данных (например, повреждение файла, ошибки ввода-вывода)

Ошибки системы хранения

- Сбои загрузки tablet
- Сбои выполнения compaction
- Повреждённые файлы данных
- Ошибки ввода-вывода диска

Конфигурация

- `sys_log_level`: Уровень журналирования, по умолчанию `INFO`
- `sys_log_dir`: Каталог хранения журналов, по умолчанию `${SELENA_HOME}/log`
- `sys_log_roll_mode`: Режим ротации журналов, по умолчанию `SIZE-MB-1024`, что означает создание нового файла журнала при достижении текущим размера 1024 МБ
- `sys_log_roll_num`: Количество сохраняемых файлов журналов, по умолчанию 10

`{be или cn}.FATAL.log`

В основном записывает различные журналы поведения во время выполнения, генерируемые узлами BE/CN, и эти журналы находятся на уровне `FATAL`. При генерации такого журнала процесс узла BE/CN завершится.

Конфигурация

- `sys_log_level`: Уровень журналирования, по умолчанию `INFO`
- `sys_log_dir`: Каталог хранения журналов, по умолчанию `${SELENA_HOME}/log`

- `sys_log_roll_mode`: Режим ротации журналов, по умолчанию `SIZE-MB-1024`, что означает создание нового файла журнала при достижении текущим размера 1024 МБ
- `sys_log_roll_num`: Количество сохраняемых файлов журналов, по умолчанию 10

`error_log`

Этот журнал в основном записывает различные ошибки, отклонённые записи и проблемы ETL, возникающие на узлах BE/CN во время импорта данных. Пользователи могут получить основные причины ошибок импорта через `http://be_ip:be_port/api/get_log_file`. Файлы журналов хранятся в каталоге `${SELENA_HOME}/storage/error_log`.

Конфигурация

- `load_error_log_reserve_hours`: Как долго хранятся файлы журналов ошибок. По умолчанию: 48 часов, что означает удаление файлов журналов через 48 часов после их создания.

2.2.2 Масштабирование cluster

В этом разделе описывается, как масштабировать узлы Selena.

Масштабирование FE

Selena имеет два типа узлов FE: Follower и Observer. Follower-узлы участвуют в выборах и записи. Observer-узлы используются только для синхронизации журналов и расширения производительности чтения.

> - Количество FE-узлов типа Follower (включая leader) должно быть нечётным, и рекомендуется развёртывать 3 из них для формирования режима высокой доступности (HA).
> - Когда FE развёрнут в режиме высокой доступности (1 leader, 2 follower), рекомендуется добавить Observer FE для лучшей производительности чтения.

Добавление узлов FE

После развёртывания узла FE и запуска сервиса выполните следующую команду для добавления FE.

```
alter system add follower "fe_host:edit_log_port";
alter system add observer "fe_host:edit_log_port";
```

Удаление узлов FE

Удаление FE аналогично добавлению. Выполните следующую команду для удаления FE.

```
alter system drop follower "fe_host:edit_log_port";
alter system drop observer "fe_host:edit_log_port";
```

После расширения и сокращения вы можете просмотреть информацию об узлах, выполнив `show proc '/frontends';`.

Масштабирование BE

Selena автоматически выполняет балансировку нагрузки после масштабирования BE без влияния на общую производительность.

При добавлении нового узла BE планировщик tablet системы обнаружит новый узел и его низкую нагрузку. Затем он начнёт перемещать tablet с узлов BE с высокой нагрузкой на новый узел BE с низкой нагрузкой, чтобы обеспечить равномерное распределение данных и нагрузки по всему cluster.

Процесс балансировки основан на показателе loadScore, рассчитываемом для каждого BE, который учитывает как использование диска, так и количество replica. Система стремится перемещать tablet с узлов с более высоким loadScore на узлы с более низким loadScore.

Вы можете проверить параметр конфигурации FE `tablet_sched_disable_balance`, чтобы убедиться, что автоматическая балансировка не отключена (параметр по умолчанию false, что означает, что балансировка tablet включена по умолчанию). Подробнее см. в документации по управлению replica.

Добавление узлов BE

Выполните следующую команду для добавления BE.

```
alter system add backend 'be_host:be_heartbeat_service_port';
```

Выполните следующую команду для проверки статуса BE.

```
show proc '/backends';
```

Удаление узлов BE

Существует два способа удаления узла BE — `DROP` и `DECOMMISSION`.

`DROP` немедленно удаляет узел BE, а потерянные дубликаты будут восполнены посредством планирования FE. `DECOMMISSION` сначала убеждается, что дубликаты восполнены, и только затем удаляет узел BE. `DECOMMISSION` более безопасен и рекомендуется для масштабирования BE.

Команды обоих методов похожи:

- `alter system decommission backend "be_host:be_heartbeat_service_port";`
- `alter system drop backend "be_host:be_heartbeat_service_port";`

Удаление backend — это опасная операция, поэтому перед выполнением необходимо подтвердить её дважды.

- `alter system drop backend "be_host:be_heartbeat_service_port";`

Масштабирование CN

Добавление узлов CN

Выполните следующую команду для добавления CN.

```
ALTER SYSTEM ADD COMPUTE NODE "cn_host:cn_heartbeat_service_port";
```

Выполните следующую команду для проверки статуса CN.

```
SHOW PROC '/compute_nodes';
```

Удаление узлов CN

Удаление CN аналогично добавлению. Выполните следующую команду для удаления CN.

```
ALTER SYSTEM DROP COMPUTE NODE "cn_host:cn_heartbeat_service_port";
```

Вы можете просмотреть информацию об узлах, выполнив `SHOW PROC '/compute_nodes';`.

2.2.3 Управление чёрным списком BE и CN

Начиная с версии v1.5.2, Selena поддерживает функцию чёрного списка BE, которая позволяет запретить использование определённых узлов BE при выполнении запросов, тем самым избегая частых сбоев запросов или другого неожиданного поведения, вызванного неудачными подключениями к узлам BE. Примером использования чёрного списка может быть сетевая проблема, препятствующая подключению к одному или нескольким BE.

Начиная с версии v2.0.0, Selena поддерживает добавление узлов CN (Compute Nodes) в чёрный список.

По умолчанию Selena может автоматически управлять чёрным списком BE и CN, добавляя узлы BE или CN, потерявшие соединение, в чёрный список и удаляя их из чёрного списка при восстановлении соединения. Однако Selena не удалит узел из чёрного списка, если он был добавлен вручную.

> **ПРИМЕЧАНИЕ**
>
> - Только пользователи с привилегией BLACKLIST на уровне SYSTEM могут использовать эту функцию.
> - Каждый узел FE ведёт свой собственный чёрный список BE и CN и не делится им с другими узлами FE.

Добавление BE/CN в чёрный список

Вы можете вручную добавить узел BE/CN в чёрный список с помощью команды ADD BACKEND/COMPUTE NODE BLACKLIST. В этой команде необходимо указать ID узла BE/CN, который нужно добавить в чёрный список. Вы можете получить ID BE, выполнив команду SHOW BACKENDS, а ID CN — выполнив команду SHOW COMPUTE NODES.

Пример:

```
-- Получить ID BE.
SHOW BACKENDS\G
*************************** 1. row ***************************
            BackendId: 10001
                   IP: xxx.xx.xx.xxx
                   ...
-- Добавить BE в чёрный список.
ADD BACKEND BLACKLIST 10001;

-- Получить ID CN.
SHOW COMPUTE NODES\G
*************************** 1. row ***************************
        ComputeNodeId: 10005
                   IP: xxx.xx.xx.xxx
                   ...
-- Добавить CN в чёрный список.
ADD COMPUTE NODE BLACKLIST 10005;
```

Удаление BE/CN из чёрного списка

Вы можете вручную удалить узел BE/CN из чёрного списка с помощью команды DELETE BACKEND/COMPUTE NODE BLACKLIST. В этой команде также необходимо указать ID узла BE/CN.

Пример:

```
-- Удалить BE из чёрного списка.
DELETE BACKEND BLACKLIST 10001;

-- Удалить CN из чёрного списка.
DELETE COMPUTE NODE BLACKLIST 10005;
```

Просмотр чёрного списка BE/CN

Вы можете просмотреть узлы BE/CN в чёрном списке с помощью команды SHOW BACKEND/COMPUTE NODE BLACKLIST.

Пример:

```
-- Просмотр чёрного списка BE.
SHOW BACKEND BLACKLIST;
+-----------+------------------+---------------------+---------------------
---------+--------------------+
| BackendId | AddBlackListType | LostConnectionTime  |
LostConnectionNumberInPeriod | CheckTimePeriod(s) |
+-----------+------------------+---------------------+---------------------
---------+--------------------+
| 10001     | MANUAL           | 2024-04-28 11:52:09 | 0
| 5                  |
+-----------+------------------+---------------------+---------------------
---------+--------------------+

-- Просмотр чёрного списка CN.
SHOW COMPUTE NODE BLACKLIST;
+---------------+------------------+---------------------+-----------------
-------------+--------------------+
| ComputeNodeId | AddBlackListType | LostConnectionTime  |
LostConnectionNumberInPeriod | CheckTimePeriod(s) |
+---------------+------------------+---------------------+-----------------
-------------+--------------------+
| 10005         | MANUAL           | 2025-08-18 10:47:51 | 0
| 5                  |
+---------------+------------------+---------------------+-----------------
-------------+--------------------+
```

Возвращаемые поля:

- `AddBlackListType`: Способ добавления узла BE/CN в чёрный список. `MANUAL` означает, что он был добавлен вручную пользователем. `AUTO` означает, что он был автоматически добавлен Selena.
- `LostConnectionTime`:
  - Для типа `MANUAL` указывает время, когда узел BE/CN был вручную добавлен в чёрный список.
  - Для типа `AUTO` указывает время последнего успешного установления соединения.
- `LostConnectionNumberInPeriod`: Количество разрывов соединения, обнаруженных в течение `CheckTimePeriod(s)`, который является интервалом, с которым Selena проверяет состояние соединения узлов BE/CN в чёрном списке.
- `CheckTimePeriod(s)`: Интервал, с которым Selena проверяет состояние соединения узлов BE/CN в чёрном списке. Его значение равно значению, указанному для

параметра конфигурации FE `black_host_history_sec`. Единица измерения: секунды.

Настройка автоматического управления чёрным списком BE/CN

Каждый раз, когда узел BE/CN теряет соединение с узлом FE или запрос завершается с ошибкой по таймауту на узле BE/CN, узел FE добавляет узел BE/CN в свой чёрный список BE и CN. Узел FE будет постоянно оценивать связность узла BE/CN в чёрном списке, подсчитывая количество сбоев соединения в течение определённого периода времени. Selena удалит узел BE/CN из чёрного списка только в том случае, если количество его сбоев соединения ниже заданного порога.

Вы можете настроить автоматическое управление чёрным списком BE и CN с помощью следующих конфигураций FE:

- `black_host_history_sec`: Продолжительность времени для хранения истории сбоев соединения узлов BE/CN в чёрном списке.
- `black_host_connect_failures_within_time`: Порог сбоев соединения, допустимый для узла BE/CN в чёрном списке.

Если узел BE/CN добавлен в чёрный список автоматически, Selena будет оценивать его связность и решать, можно ли его удалить из чёрного списка. В течение `black_host_history_sec` узел BE/CN в чёрном списке может быть удалён из него только в том случае, если количество сбоев его соединения меньше порога, установленного в `black_host_connect_failures_within_time`.

2.2.4 Резервное копирование и восстановление данных

В этом разделе описывается, как выполнять резервное копирование и восстановление данных в Selena, а также как мигрировать данные в новый cluster Selena.

Selena поддерживает резервное копирование данных в виде снимков во внешнюю систему хранения и восстановление данных в любые cluster Selena.

Начиная с версии v1.5.2, Selena расширила функциональность BACKUP и RESTORE, добавив поддержку большего количества объектов и переработав синтаксис для большей гибкости.

Selena поддерживает следующие внешние системы хранения:

- Apache™ Hadoop® (HDFS) cluster

- AWS S3
- Google GCS
- MinIO

Selena поддерживает резервное копирование следующих объектов:

- Внутренние базы данных, таблицы (всех типов и стратегий партиционирования) и партиции
- Метаданные внешних каталогов (поддерживается с версии v1.5.2)
- Синхронные материализованные представления и асинхронные материализованные представления
- Логические представления (поддерживается с версии v1.5.2)
- Пользовательские функции (поддерживается с версии v1.5.2)

> **ПРИМЕЧАНИЕ**
>
> Cluster Selena с общими данными (shared-data) не поддерживают BACKUP и RESTORE данных.

Создание репозитория

Перед резервным копированием данных необходимо создать репозиторий, который используется для хранения снимков данных во внешней системе хранения. В cluster Selena можно создать несколько репозиториев. Подробные инструкции см. в разделе CREATE REPOSITORY.

- Создание репозитория в HDFS

Следующий пример создаёт репозиторий с именем `test_repo` в cluster HDFS.

```
CREATE REPOSITORY test_repo
WITH BROKER
ON LOCATION "hdfs://<hdfs_host>:<hdfs_port>/repo_dir/backup"
PROPERTIES(
    "username" = "<hdfs_username>",
    "password" = "<hdfs_password>"
);
```

- Создание репозитория в AWS S3

Вы можете выбрать учётные данные на основе IAM-пользователя (Access Key и Secret Key), Instance Profile или Assumed Role в качестве метода аутентификации для доступа к AWS S3.

- Следующий пример создаёт репозиторий с именем `test_repo` в бакете AWS S3 `bucket_s3`, используя учётные данные на основе IAM-пользователя.

```
CREATE REPOSITORY test_repo
WITH BROKER
ON LOCATION "s3a://bucket_s3/backup"
PROPERTIES(
    "aws.s3.access_key" = "XXXXXXXXXXXXXXXXX",
    "aws.s3.secret_key" = "yyyyyyyyyyyyyyyyyyyyyyyy",
    "aws.s3.region" = "us-east-1"
);
```

- Следующий пример создаёт репозиторий с именем `test_repo` в бакете AWS S3 `bucket_s3`, используя Instance Profile в качестве метода аутентификации.

```
CREATE REPOSITORY test_repo
WITH BROKER
ON LOCATION "s3a://bucket_s3/backup"
PROPERTIES(
    "aws.s3.use_instance_profile" = "true",
    "aws.s3.region" = "us-east-1"
);
```

- Следующий пример создаёт репозиторий с именем `test_repo` в бакете AWS S3 `bucket_s3`, используя Assumed Role в качестве метода аутентификации.

```
CREATE REPOSITORY test_repo
WITH BROKER
ON LOCATION "s3a://bucket_s3/backup"
PROPERTIES(
    "aws.s3.use_instance_profile" = "true",
    "aws.s3.iam_role_arn" = "arn:aws:iam::xxxxxxxxxx:role/yyyyyyyy",
    "aws.s3.region" = "us-east-1"
);
```

> **ПРИМЕЧАНИЕ**

> Selena поддерживает создание репозиториев в AWS S3 только по протоколу S3A. Поэтому при создании репозиториев в AWS S3 необходимо заменить `s3://` в URI S3, который передаётся в качестве местоположения репозитория в `ON LOCATION`, на `s3a://`.

- Создание репозитория в Google GCS

Следующий пример создаёт репозиторий с именем `test_repo` в бакете Google GCS `bucket_gcs`.

```
CREATE REPOSITORY test_repo
WITH BROKER
ON LOCATION "s3a://bucket_gcs/backup"
PROPERTIES(
    "fs.s3a.access.key" = "xxxxxxxxxxxxxxxxxxxx",
    "fs.s3a.secret.key" = "yyyyyyyyyyyyyyyyyyyy",
    "fs.s3a.endpoint" = "storage.googleapis.com"
);
```

> **ПРИМЕЧАНИЕ**
>
> - Selena поддерживает создание репозиториев в Google GCS только по протоколу S3A. Поэтому при создании репозиториев в Google GCS необходимо заменить префикс в URI GCS, который передаётся в качестве местоположения репозитория в `ON LOCATION`, на `s3a://`.
> - Не указывайте `https` в адресе конечной точки.

- Создание репозитория в MinIO

Следующий пример создаёт репозиторий с именем `test_repo` в бакете MinIO `bucket_minio`.

```
CREATE REPOSITORY test_repo
WITH BROKER
ON LOCATION "s3://bucket_minio/backup"
PROPERTIES(
    "aws.s3.access_key" = "XXXXXXXXXXXXXXXXXX",
    "aws.s3.secret_key" = "yyyyyyyyyyyyyyyyyy",
    "aws.s3.endpoint" = "http://minio:9000"
);
```

После создания репозитория вы можете проверить его с помощью SHOW REPOSITORIES. После восстановления данных вы можете удалить репозиторий в Selena с помощью DROP REPOSITORY. Однако снимки данных, сохранённые во внешней системе хранения, не могут быть удалены через Selena. Их необходимо удалить вручную во внешней системе хранения.

Резервное копирование данных

После создания репозитория необходимо создать снимок данных и сохранить его во внешнем репозитории. Подробные инструкции см. в разделе BACKUP. BACKUP — это асинхронная операция. Вы можете проверить статус задания BACKUP с помощью SHOW BACKUP или отменить задание BACKUP с помощью CANCEL BACKUP.

Selena поддерживает ПОЛНОЕ резервное копирование на уровне детализации базы данных, таблицы или партиции.

Если в таблице хранится большой объём данных, рекомендуется выполнять резервное копирование и восстановление по партициям. Это позволяет снизить затраты на повторные попытки в случае сбоя задания. Если требуется регулярное резервное копирование инкрементных данных, вы можете настроить план партиционирования для таблицы и каждый раз создавать резервные копии только новых партиций.

Резервное копирование базы данных

Выполнение полного BACKUP базы данных создаст резервную копию всех таблиц, синхронных и асинхронных материализованных представлений, логических представлений и UDF в базе данных.

Следующие примеры создают резервную копию базы данных `sr_hub` в снимке `sr_hub_backup` и загружают снимок в репозиторий `test_repo`.

```
-- Поддерживается с версии v1.5.2.
BACKUP DATABASE sr_hub SNAPSHOT sr_hub_backup
TO test_repo;

-- Совместимо с синтаксисом более ранних версий.
BACKUP SNAPSHOT sr_hub.sr_hub_backup
TO test_repo;
```

Резервное копирование таблицы

Selena поддерживает резервное копирование и восстановление таблиц всех типов и стратегий партиционирования. Выполнение полного BACKUP таблицы создаст резервную копию таблицы и синхронных материализованных представлений, построенных на её основе.

Следующие примеры создают резервную копию таблицы `sr_member` из базы данных `sr_hub` в снимке `sr_member_backup` и загружают снимок в репозиторий `test_repo`.

```
-- Поддерживается с версии v1.5.2.
BACKUP DATABASE sr_hub SNAPSHOT sr_member_backup
TO test_repo
ON (TABLE sr_member);

-- Совместимо с синтаксисом более ранних версий.
BACKUP SNAPSHOT sr_hub.sr_member_backup
TO test_repo
ON (sr_member);
```

Следующий пример создаёт резервную копию двух таблиц, `sr_member` и `sr_pmc`, из базы данных `sr_hub` в снимке `sr_core_backup` и загружает снимок в репозиторий `test_repo`.

```
BACKUP DATABASE sr_hub SNAPSHOT sr_core_backup
TO test_repo
ON (TABLE sr_member, TABLE sr_pmc);
```

Следующий пример создаёт резервную копию всех таблиц из базы данных `sr_hub` в снимке `sr_all_backup` и загружает снимок в репозиторий `test_repo`.

```
BACKUP DATABASE sr_hub SNAPSHOT sr_all_backup
TO test_repo
ON (ALL TABLES);
```

Резервное копирование партиции

Следующие примеры создают резервную копию партиции `p1` таблицы `sr_member` из базы данных `sr_hub` в снимке `sr_par_backup` и загружают снимок в репозиторий `test_repo`.

```
-- Поддерживается с версии v1.5.2.
BACKUP DATABASE sr_hub SNAPSHOT sr_par_backup
TO test_repo
ON (TABLE sr_member PARTITION (p1));

-- Совместимо с синтаксисом более ранних версий.
BACKUP SNAPSHOT sr_hub.sr_par_backup
TO test_repo
ON (sr_member PARTITION (p1));
```

Вы можете указать несколько имён партиций, разделённых запятыми (`,`), для пакетного резервного копирования партиций.

Резервное копирование материализованного представления

Вам не нужно вручную создавать резервные копии синхронных материализованных представлений, поскольку они будут включены в резервную копию при выполнении операции BACKUP базовой таблицы.

Асинхронные материализованные представления могут быть включены в резервную копию при выполнении операции BACKUP базы данных, к которой они принадлежат. Вы также можете создать их резервную копию вручную.

Следующий пример создаёт резервную копию материализованного представления `sr_mv1` из базы данных `sr_hub` в снимке `sr_mv1_backup` и загружает снимок в репозиторий `test_repo`.

```
BACKUP DATABASE sr_hub SNAPSHOT sr_mv1_backup
TO test_repo
ON (MATERIALIZED VIEW sr_mv1);
```

Следующий пример создаёт резервную копию двух материализованных представлений, `sr_mv1` и `sr_mv2`, из базы данных `sr_hub` в снимке `sr_mv2_backup` и загружает снимок в репозиторий `test_repo`.

```
BACKUP DATABASE sr_hub SNAPSHOT sr_mv2_backup
TO test_repo
ON (MATERIALIZED VIEW sr_mv1, MATERIALIZED VIEW sr_mv2);
```

Следующий пример создаёт резервную копию всех материализованных представлений из базы данных `sr_hub` в снимке `sr_mv3_backup` и загружает снимок в репозиторий `test_repo`.

```
BACKUP DATABASE sr_hub SNAPSHOT sr_mv3_backup
TO test_repo
ON (ALL MATERIALIZED VIEWS);
```

Резервное копирование логического представления

Следующий пример создаёт резервную копию логического представления `sr_view1` из базы данных `sr_hub` в снимке `sr_view1_backup` и загружает снимок в репозиторий `test_repo`.

```
BACKUP DATABASE sr_hub SNAPSHOT sr_view1_backup
TO test_repo
ON (VIEW sr_view1);
```

Следующий пример создаёт резервную копию двух логических представлений, `sr_view1` и `sr_view2`, из базы данных `sr_hub` в снимке `sr_view2_backup` и загружает снимок в репозиторий `test_repo`.

```
BACKUP DATABASE sr_hub SNAPSHOT sr_view2_backup
TO test_repo
ON (VIEW sr_view1, VIEW sr_view2);
```

Следующий пример создаёт резервную копию всех логических представлений из базы данных `sr_hub` в снимке `sr_view3_backup` и загружает снимок в репозиторий `test_repo`.

```
BACKUP DATABASE sr_hub SNAPSHOT sr_view3_backup
TO test_repo
ON (ALL VIEWS);
```

Резервное копирование UDF

Следующий пример создаёт резервную копию UDF `sr_udf1` из базы данных `sr_hub` в снимке `sr_udf1_backup` и загружает снимок в репозиторий `test_repo`.

```
BACKUP DATABASE sr_hub SNAPSHOT sr_udf1_backup
TO test_repo
ON (FUNCTION sr_udf1);
```

Следующий пример создаёт резервную копию двух UDF, `sr_udf1` и `sr_udf2`, из базы данных `sr_hub` в снимке `sr_udf2_backup` и загружает снимок в репозиторий `test_repo`.

```
BACKUP DATABASE sr_hub SNAPSHOT sr_udf2_backup
TO test_repo
ON (FUNCTION sr_udf1, FUNCTION sr_udf2);
```

Следующий пример создаёт резервную копию всех UDF из базы данных `sr_hub` в снимке `sr_udf3_backup` и загружает снимок в репозиторий `test_repo`.

```
BACKUP DATABASE sr_hub SNAPSHOT sr_udf3_backup
TO test_repo
ON (ALL FUNCTIONS);
```

Резервное копирование метаданных внешнего каталога

Следующий пример создаёт резервную копию метаданных внешнего каталога `iceberg` в снимке `iceberg_backup` и загружает снимок в репозиторий `test_repo`.

```
BACKUP EXTERNAL CATALOG (iceberg) SNAPSHOT iceberg_backup
TO test_repo;
```

Следующий пример создаёт резервную копию метаданных двух внешних каталогов, `iceberg` и `hive`, в снимке `iceberg_hive_backup` и загружает снимок в репозиторий `test_repo`.

```
BACKUP EXTERNAL CATALOGS (iceberg, hive) SNAPSHOT iceberg_hive_backup
TO test_repo;
```

Следующий пример создаёт резервную копию метаданных всех внешних каталогов в снимке `all_catalog_backup` и загружает снимок в репозиторий `test_repo`.

```
BACKUP ALL EXTERNAL CATALOGS SNAPSHOT all_catalog_backup
TO test_repo;
```

Чтобы отменить операцию BACKUP для внешних каталогов, выполните следующее выражение:

```
CANCEL BACKUP FOR EXTERNAL CATALOG;
```

Восстановление данных

Вы можете восстановить снимок данных, сохранённый во внешней системе хранения, в текущий или другие cluster Selena для восстановления или миграции данных.

При восстановлении объекта из снимка необходимо указать временную метку снимка.

Используйте выражение RESTORE для восстановления снимков данных из внешней системы хранения.

RESTORE — это асинхронная операция. Вы можете проверить статус задания RESTORE с помощью SHOW RESTORE или отменить задание RESTORE с помощью CANCEL RESTORE.

(Опционально) Создание репозитория в новом cluster

Чтобы мигрировать данные в другой cluster Selena, необходимо создать репозиторий с тем же **именем репозитория** и **местоположением** в целевом cluster, иначе вы не сможете просмотреть ранее сохранённые снимки данных. Подробности см. в разделе Создание репозитория.

Получение временной метки снимка

Перед восстановлением данных вы можете проверить снимки в репозитории, чтобы получить временные метки, с помощью SHOW SNAPSHOT.

Следующий пример проверяет информацию о снимках в `test_repo`.

```
mysql> SHOW SNAPSHOT ON test_repo;
+------------------+-------------------------+--------+
| Snapshot         | Timestamp               | Status |
+------------------+-------------------------+--------+
| sr_member_backup | 2023-02-07-14-45-53-143 | OK     |
+------------------+-------------------------+--------+
1 row in set (1.16 sec)
```

Восстановление базы данных

Следующие примеры восстанавливают базу данных `sr_hub` из снимка `sr_hub_backup` в базу данных `sr_hub` в целевом cluster. Если база данных не существует в снимке, система вернёт ошибку. Если база данных не существует в целевом cluster, система создаст её автоматически.

```
-- Поддерживается с версии v1.5.2.
RESTORE SNAPSHOT sr_hub_backup
FROM test_repo
DATABASE sr_hub
PROPERTIES("backup_timestamp" = "2024-12-09-10-25-58-842");

-- Совместимо с синтаксисом более ранних версий.
RESTORE SNAPSHOT sr_hub.sr_hub_backup
FROM `test_repo`
PROPERTIES("backup_timestamp" = "2024-12-09-10-25-58-842");
```

Следующий пример восстанавливает базу данных `sr_hub` из снимка `sr_hub_backup` в базу данных `sr_hub_new` в целевом cluster. Если база данных `sr_hub` не существует в снимке, система вернёт ошибку. Если база данных `sr_hub_new` не существует в целевом cluster, система создаст её автоматически.

```
-- Поддерживается с версии v1.5.2.
RESTORE SNAPSHOT sr_hub_backup
FROM test_repo
DATABASE sr_hub AS sr_hub_new
PROPERTIES("backup_timestamp" = "2024-12-09-10-25-58-842");
```

Восстановление таблицы

Следующие примеры восстанавливают таблицу `sr_member` из базы данных `sr_hub` в снимке `sr_member_backup` в таблицу `sr_member` базы данных `sr_hub` в целевом cluster.

```
-- Поддерживается с версии v1.5.2.
RESTORE SNAPSHOT sr_member_backup
FROM test_repo
DATABASE sr_hub
ON (TABLE sr_member)
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");

-- Совместимо с синтаксисом более ранних версий.
RESTORE SNAPSHOT sr_hub.sr_member_backup
FROM test_repo
ON (sr_member)
PROPERTIES ("backup_timestamp"="2024-12-09-10-52-10-940");
```

Следующие примеры восстанавливают таблицу `sr_member` из базы данных `sr_hub` в снимке `sr_member_backup` в таблицу `sr_member_new` базы данных `sr_hub_new` в целевом cluster.

```
RESTORE SNAPSHOT sr_member_backup
FROM test_repo
DATABASE sr_hub  AS sr_hub_new
ON (TABLE sr_member AS sr_member_new)
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");
```

Следующий пример восстанавливает две таблицы, `sr_member` и `sr_pmc`, из базы данных `sr_hub` в снимке `sr_core_backup` в две таблицы, `sr_member` и `sr_pmc`, базы данных `sr_hub` в целевом cluster.

```
RESTORE SNAPSHOT sr_core_backup
FROM test_repo
DATABASE sr_hub
ON (TABLE sr_member, TABLE sr_pmc)
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");
```

Следующий пример восстанавливает все таблицы из базы данных `sr_hub` в снимке `sr_all_backup`.

```
RESTORE SNAPSHOT sr_all_backup
FROM test_repo
DATABASE sr_hub
ON (ALL TABLES);
```

Следующий пример восстанавливает одну из всех таблиц из базы данных `sr_hub` в снимке `sr_all_backup`.

```
RESTORE SNAPSHOT sr_all_backup
FROM test_repo
DATABASE sr_hub
ON (TABLE sr_member)
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");
```

Восстановление партиции

Следующие примеры восстанавливают партицию `p1` таблицы `sr_member` в снимке `sr_par_backup` в партицию `p1` таблицы `sr_member` в целевом cluster.

```
-- Поддерживается с версии v1.5.2.
RESTORE SNAPSHOT sr_par_backup
FROM test_repo
DATABASE sr_hub
ON (TABLE sr_member PARTITION (p1))
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");

-- Совместимо с синтаксисом более ранних версий.
RESTORE SNAPSHOT sr_hub.sr_par_backup
FROM test_repo
ON (sr_member PARTITION (p1))
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");
```

Вы можете указать несколько имён партиций, разделённых запятыми (`,`), для пакетного восстановления партиций.

Восстановление материализованного представления

Следующий пример восстанавливает материализованное представление `sr_mv1` из базы данных `sr_hub` в снимке `sr_mv1_backup` в целевой cluster.

```
RESTORE SNAPSHOT sr_mv1_backup
FROM test_repo
DATABASE sr_hub
ON (MATERIALIZED VIEW sr_mv1)
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");
```

Следующий пример восстанавливает два материализованных представления, `sr_mv1` и `sr_mv2`, из базы данных `sr_hub` в снимке `sr_mv2_backup` в целевой cluster.

```
RESTORE SNAPSHOT sr_mv2_backup
FROM test_repo
DATABASE sr_hub
ON (MATERIALIZED VIEW sr_mv1, MATERIALIZED VIEW sr_mv2)
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");
```

Следующий пример восстанавливает все материализованные представления из базы данных `sr_hub` в снимке `sr_mv3_backup` в целевой cluster.

```
RESTORE SNAPSHOT sr_mv3_backup
FROM test_repo
DATABASE sr_hub
ON (ALL MATERIALIZED VIEWS)
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");
```

Следующий пример восстанавливает одно из материализованных представлений из базы данных `sr_hub` в снимке `sr_mv3_backup` в целевой cluster.

```
RESTORE SNAPSHOT sr_mv3_backup
FROM test_repo
DATABASE sr_hub
ON (MATERIALIZED VIEW sr_mv1)
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");
```

К СВЕДЕНИЮ

После RESTORE вы можете проверить статус материализованного представления с помощью SHOW MATERIALIZED VIEWS.

- Если материализованное представление активно, его можно использовать сразу.
- Если материализованное представление неактивно, это может быть связано с тем, что его базовые таблицы не были восстановлены. После восстановления всех базовых таблиц вы можете использовать ALTER MATERIALIZED VIEW для повторной активации материализованного представления.

Восстановление логического представления

Следующий пример восстанавливает логическое представление `sr_view1` из базы данных `sr_hub` в снимке `sr_view1_backup` в целевой cluster.

```
RESTORE SNAPSHOT sr_view1_backup
FROM test_repo
DATABASE sr_hub
ON (VIEW sr_view1)
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");
```

Следующий пример восстанавливает два логических представления, `sr_view1` и `sr_view2`, из базы данных `sr_hub` в снимке `sr_view2_backup` в целевой cluster.

```
RESTORE SNAPSHOT sr_view2_backup
FROM test_repo
DATABASE sr_hub
ON (VIEW sr_view1, VIEW sr_view2)
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");
```

Следующий пример восстанавливает все логические представления из базы данных `sr_hub` в снимке `sr_view3_backup` в целевой cluster.

```
RESTORE SNAPSHOT sr_view3_backup
FROM test_repo
DATABASE sr_hub
ON (ALL VIEWS)
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");
```

Следующий пример восстанавливает одно из всех логических представлений из базы данных `sr_hub` в снимке `sr_view3_backup` в целевой cluster.

```
RESTORE SNAPSHOT sr_view3_backup
FROM test_repo
DATABASE sr_hub
ON (VIEW sr_view1)
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");
```

Восстановление UDF

Следующий пример восстанавливает UDF `sr_udf1` из базы данных `sr_hub` в снимке `sr_udf1_backup` в целевой cluster.

```
RESTORE SNAPSHOT sr_udf1_backup
FROM test_repo
DATABASE sr_hub
ON (FUNCTION sr_udf1)
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");
```

Следующий пример восстанавливает два UDF, `sr_udf1` и `sr_udf2`, из базы данных `sr_hub` в снимке `sr_udf2_backup` в целевой cluster.

```
RESTORE SNAPSHOT sr_udf2_backup
FROM test_repo
DATABASE sr_hub
ON (FUNCTION sr_udf1, FUNCTION sr_udf2)
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");
```

Следующий пример восстанавливает все UDF из базы данных `sr_hub` в снимке `sr_udf3_backup` в целевой cluster.

```
RESTORE SNAPSHOT sr_udf3_backup
FROM test_repo
DATABASE sr_hub
ON (ALL FUNCTIONS)
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");
```

Следующий пример восстанавливает один из всех UDF из базы данных `sr_hub` в снимке `sr_udf3_backup` в целевой cluster.

```
RESTORE SNAPSHOT sr_udf3_backup
FROM test_repo
DATABASE sr_hub
ON (FUNCTION sr_udf1)
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");
```

Восстановление метаданных внешнего каталога

Следующий пример восстанавливает метаданные внешнего каталога `iceberg` из снимка `iceberg_backup` в целевой cluster и переименовывает его в `iceberg_new`.

```
RESTORE SNAPSHOT iceberg_backup
FROM test_repo
EXTERNAL CATALOG (iceberg AS iceberg_new)
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");
```

Следующий пример восстанавливает метаданные двух внешних каталогов, `iceberg` и `hive`, из снимка `iceberg_hive_backup` в целевой cluster.

```
RESTORE SNAPSHOT iceberg_hive_backup
FROM test_repo
EXTERNAL CATALOGS (iceberg, hive)
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");
```

Следующий пример восстанавливает метаданные всех внешних каталогов из снимка `all_catalog_backup` в целевой cluster.

```
RESTORE SNAPSHOT all_catalog_backup
FROM test_repo
ALL EXTERNAL CATALOGS
PROPERTIES ("backup_timestamp" = "2024-12-09-10-52-10-940");
```

Чтобы отменить операцию RESTORE для внешних каталогов, выполните следующее выражение:

```
CANCEL RESTORE FOR EXTERNAL CATALOG;
```

Настройка заданий BACKUP или RESTORE

Вы можете оптимизировать производительность заданий BACKUP или RESTORE, изменив следующие параметры конфигурации в файле конфигурации BE **be.conf**:

Параметр конфигурации	Описание
make_snapshot_worker_count	Максимальное количество потоков для задач создания снимков заданий BACKUP на узле BE. По умолчанию: `5`. Увеличьте значение этого параметра для увеличения параллелизма задачи создания снимка.
release_snapshot_worker_count	Максимальное количество потоков для задач освобождения снимков неудавшихся заданий BACKUP на узле BE. По умолчанию: `5`. Увеличьте значение этого параметра для увеличения параллелизма задачи освобождения снимка.
upload_worker_count	Максимальное количество потоков для задач загрузки заданий BACKUP на узле BE. По умолчанию: `0`. `0` означает, что значение устанавливается равным количеству ядер CPU на машине, где находится BE. Увеличьте значение этого параметра для увеличения параллелизма задачи загрузки.
download_worker_count	Максимальное количество потоков для задач скачивания заданий RESTORE на узле BE. По умолчанию: `0`. `0` означает, что значение устанавливается равным количеству ядер CPU на машине, где находится BE. Увеличьте значение этого параметра для увеличения параллелизма задачи скачивания.

Примечания по использованию

- Выполнение операций резервного копирования и восстановления на глобальном уровне, уровне базы данных, таблицы и партиции требует различных привилегий. Подробную информацию см. в разделе Настройка ролей на основе сценариев.
- В каждой базе данных одновременно разрешено выполнение только одного задания BACKUP или RESTORE. В противном случае Selena вернёт ошибку.

- Поскольку задания BACKUP и RESTORE занимают много ресурсов вашего cluster Selena, вы можете выполнять резервное копирование и восстановление данных, когда ваш cluster Selena не сильно загружен.
- Selena не поддерживает указание алгоритмов сжатия данных для резервного копирования.
- Поскольку данные резервируются в виде снимков, данные, загруженные после создания снимка, не включаются в снимок. Поэтому, если вы загружаете данные в старый cluster после создания снимка и до завершения задания RESTORE, вам также необходимо загрузить данные в cluster, в который восстанавливаются данные. Рекомендуется параллельно загружать данные в оба cluster в течение некоторого времени после завершения миграции данных, а затем перевести приложение на новый cluster после проверки корректности данных и сервисов.
- До завершения задания RESTORE вы не можете оперировать таблицей, которая восстанавливается.
- Таблицы с первичным ключом не могут быть восстановлены в cluster Selena версии ранее v1.5.2.
- Вам не нужно создавать таблицу, которая будет восстановлена, в новом cluster перед её восстановлением. Задание RESTORE создаст её автоматически.
- Если существует таблица с дублирующимся именем с таблицей, которая будет восстановлена, Selena сначала проверяет, соответствует ли схема существующей таблицы схеме таблицы, которая будет восстановлена. Если схемы совпадают, Selena перезаписывает существующую таблицу данными из снимка. Если схема не совпадает, задание RESTORE завершится неудачей. Вы можете либо переименовать таблицу, которая будет восстановлена, используя ключевое слово `AS`, либо удалить существующую таблицу перед восстановлением данных.
- Если задание RESTORE перезаписывает существующую базу данных, таблицу или партицию, перезаписанные данные не могут быть восстановлены после того, как задание войдёт в фазу COMMIT. Если задание RESTORE завершится неудачей или будет отменено на этом этапе, данные могут быть повреждены и недоступны. В этом случае вы можете только выполнить операцию RESTORE снова и дождаться завершения задания. Поэтому мы рекомендуем не восстанавливать данные путём перезаписи, если вы не уверены, что текущие данные больше не используются. Операция перезаписи сначала проверяет согласованность метаданных между снимком и существующей базой данных, таблицей или партицией. Если обнаружено несоответствие, операция RESTORE не может быть выполнена.

- В настоящее время Selena не поддерживает резервное копирование и восстановление конфигурационных данных, связанных с учётными записями пользователей, привилегиями и группами ресурсов.
- В настоящее время Selena не поддерживает резервное копирование и восстановление отношений Colocate JOIN между таблицами.

2.2.5 Cluster Snapshot

> *Beta* **BETA FEATURE**
>
> Рекомендации по использованию бета-функций

В этом разделе описывается использование Cluster Snapshot для аварийного восстановления на cluster с разделяемыми данными (shared-data).

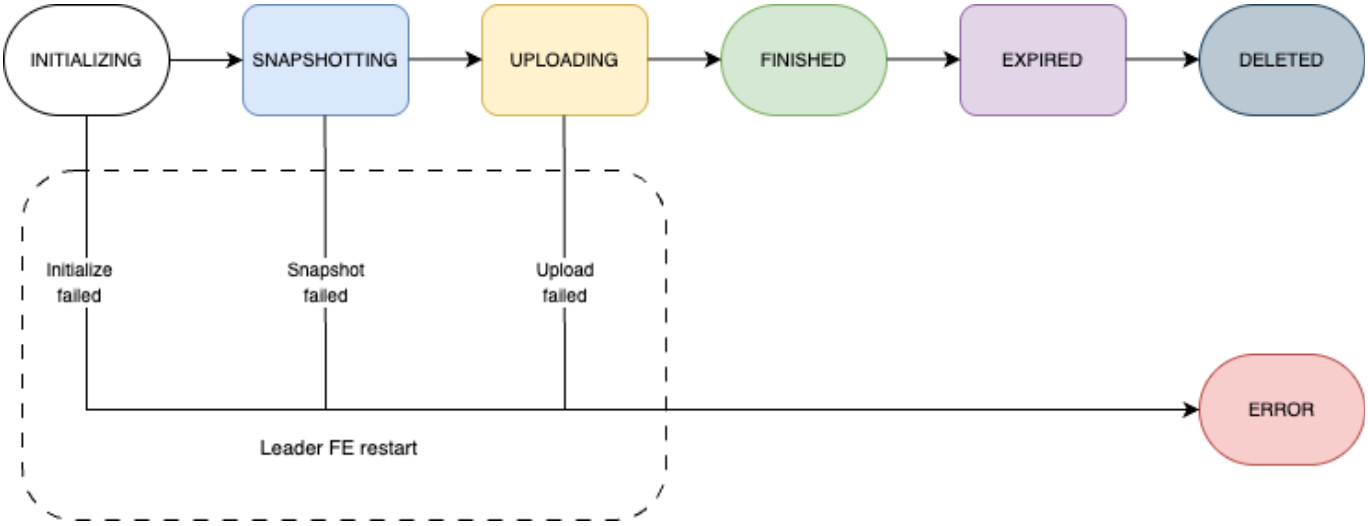
Эта функция поддерживается начиная с версии v1.5.2 и доступна только на cluster с разделяемыми данными.

Обзор

Основная идея аварийного восстановления для cluster с разделяемыми данными заключается в том, чтобы обеспечить хранение полного состояния cluster (включая данные и метаданные) в объектном хранилище. Таким образом, если cluster столкнётся со сбоем, его можно восстановить из объектного хранилища, если данные и метаданные останутся нетронутыми. Кроме того, такие функции, как резервное копирование и кросс-региональная репликация, предлагаемые облачными провайдерами, могут использоваться для удалённого восстановления и кросс-регионал ьного аварийного восстановления.

В cluster с разделяемыми данными состояние CN (данные) хранится в объектном хранилище, но состояние FE (метаданные) остаётся локальным. Чтобы гарантировать, что объектное хранилище содержит всё состояние cluster для восстановления, Selena теперь поддерживает Cluster Snapshot как для данных, так и для метаданных в объектном хранилище.

Рабочий процесс



Термины

- **Cluster snapshot**

  Cluster snapshot означает snapshot состояния cluster в определённый момент времени. Он содержит все объекты в cluster, такие как каталоги, базы данных, таблицы, пользователи и привилегии, задачи загрузки и многое другое. Он не включает все внешние зависимые объекты, такие как конфигурационные файлы внешних каталогов и локальные JAR-пакеты UDF.

- **Создание cluster snapshot**

  Система автоматически поддерживает snapshot, близко следующий за последним состоянием cluster. Исторические snapshot удаляются сразу после создания последнего, сохраняя только один snapshot доступным всё время.

- **Восстановление cluster**

  Восстановление cluster из snapshot.

Автоматический cluster snapshot

Автоматический Cluster Snapshot отключён по умолчанию.

Используйте следующую команду для включения этой функции:

Синтаксис:

```
ADMIN SET AUTOMATED CLUSTER SNAPSHOT ON
[STORAGE VOLUME <storage_volume_name>]
```

Параметр:

`storage_volume_name`: Указывает storage volume, используемый для хранения snapshot. Если этот параметр не указан, будет использоваться storage volume по умолчанию. Подробнее о создании storage volume см. CREATE STORAGE VOLUME.

Каждый раз, когда FE создаёт новый образ метаданных после завершения checkpoint метаданных, автоматически создаётся snapshot. Имя snapshot генерируется системой по формату `automated_cluster_snapshot_{timestamp}`.

Snapshot метаданных хранятся в `/{storage_volume_locations}/{service_id}/meta/image/automated_cluster_snapshot_timestamp`. Snapshot данных хранятся в том же месте, что и исходные данные.

Параметр конфигурации FE `automated_cluster_snapshot_interval_seconds` управляет циклом автоматизации snapshot. Значение по умолчанию — 600 секунд (10 минут).

Отключение автоматического cluster snapshot

Используйте следующую команду для отключения автоматического cluster snapshot:

```
ADMIN SET AUTOMATED CLUSTER SNAPSHOT OFF
```

После отключения автоматического Cluster Snapshot система автоматически удалит исторический snapshot.

Просмотр cluster snapshot

Вы можете запросить представление `information_schema.cluster_snapshots` для просмотра последнего cluster snapshot и snapshot, которые ещё не были удалены.

```
SELECT * FROM information_schema.cluster_snapshots;
```

Возвращаемые поля:

Поле	Описание
snapshot_name	Имя snapshot.
snapshot_type	Тип snapshot. Допустимые значения: `automated` и `manual`.
created_time	Время создания snapshot.
fe_journal_id	ID журнала FE.
starmgr_journal_id	ID журнала StarManager.
properties	Относится к функции, которая ещё не доступна.
storage_volume	Storage volume, где хранится snapshot.
storage_path	Путь хранения, в котором находится snapshot.

Просмотр задания cluster snapshot

Вы можете запросить представление `information_schema.cluster_snapshot_jobs` для просмотра информации о заданиях cluster snapshot.

```
SELECT * FROM information_schema.cluster_snapshot_jobs;
```

Возвращаемые поля:

Поле	Описание
snapshot_name	Имя snapshot.
job_id	ID задания.
created_time	Время создания задания.
finished_time	Время завершения задания.
state	Состояние задания. Допустимые значения: `INITIALIZING`, `SNAPSHOTING`, `FINISHED`, `EXPIRED`, `DELETED` и `ERROR`.
detail_info	Информация о прогрессе текущего этапа выполнения.
error_message	Сообщение об ошибке задания (если есть).

Восстановление cluster

Выполните следующие шаги для восстановления cluster с помощью cluster snapshot.

1. **(Опционально)** Если местоположение хранилища (storage volume), в котором хранится cluster snapshot, изменилось, все файлы из исходного пути хранения должны быть скопированы в новый путь. Для этого вам необходимо изменить

конфигурационный файл **cluster_snapshot.yaml** в директории `fe/conf` узлов Leader FE. Шаблон **cluster_snapshot.yaml** см. в Приложении.

2. Запустите узел Leader FE.

```
./fe/bin/start_fe.sh --cluster_snapshot --daemon
```

3. Запустите другие узлы FE **после очистки директорий** `meta`.

```
./fe/bin/start_fe.sh --helper <leader_ip>:<leader_edit_log_port> --
daemon
```

4. Запустите узлы CN **после очистки директорий** `storage_root_path`.

```
./be/bin/start_cn.sh --daemon
```

Если вы изменили **cluster_snapshot.yaml** на шаге 1, узел и storage volume будут переконфигурированы в новом cluster в соответствии с информацией в файле.

Приложение

Шаблон **cluster_snapshot.yaml**:

```yaml
# Информация о cluster snapshot, который будет загружен для восстановления.
cluster_snapshot:
    # URI snapshot.
    # Пример 1: s3://defaultbucket/test/f7265e80-631c-44d3-a8ac-cf7cdc7adec811019/meta/image/automated_cluster_snapshot_1704038400000
    # Пример 2: s3://defaultbucket/test/f7265e80-631c-44d3-a8ac-cf7cdc7adec811019/meta
    cluster_snapshot_path: <cluster_snapshot_uri>
    # Имя storage volume для хранения snapshot. Вы должны определить его в разделе `storage_volumes`.
    # ПРИМЕЧАНИЕ: Оно должно быть идентичным тому, что было в исходном cluster.
    storage_volume_name: my_s3_volume

# [Опционально] Информация об узлах нового cluster, в котором будет восстановлен snapshot.
# Если этот раздел не указан, новый cluster после восстановления будет содержать только узел Leader FE.
# Узлы CN сохраняют информацию исходного cluster.
# ПРИМЕЧАНИЕ: НЕ включайте узел Leader FE в этот раздел.

frontends:
    # Хост FE.
  - host: xxx.xx.xx.x1
    # edit_log_port FE.
    edit_log_port: 9010
    # Тип узла FE. Допустимые значения: `follower` (по умолчанию) и `observer`.
    type: follower
  - host: xxx.xx.xx.x2
    edit_log_port: 9010
    type: observer

compute_nodes:
    # Хост CN.
  - host: xxx.xx.xx.x3
    # heartbeat_service_port CN.
    heartbeat_service_port: 9050
  - host: xxx.xx.xx.x4
    heartbeat_service_port: 9050

# Информация о storage volume в новом cluster. Используется для восстановления клонированного snapshot.
# ПРИМЕЧАНИЕ: Имя storage volume должно быть идентичным тому, что было в исходном cluster.
```

```yaml
storage_volumes:
  # Пример для S3-совместимого storage volume.
  - name: my_s3_volume
    type: S3
    location: s3://defaultbucket/test/
    comment: my s3 volume
    properties:
      - key: aws.s3.region
        value: us-west-2
      - key: aws.s3.endpoint
        value: https://s3.us-west-2.amazonaws.com
      - key: aws.s3.access_key
        value: xxxxxxxxxx
      - key: aws.s3.secret_key
        value: yyyyyyyyyy
  # Пример для HDFS storage volume.
  - name: my_hdfs_volume
    type: HDFS
    location: hdfs://127.0.0.1:9000/sr/test/
    comment: my hdfs volume
    properties:
      - key: hadoop.security.authentication
        value: simple
      - key: username
        value: selena
```

ПРИМЕЧАНИЕ

Подробнее об учётных данных для AWS см. Аутентификация в AWS S3.

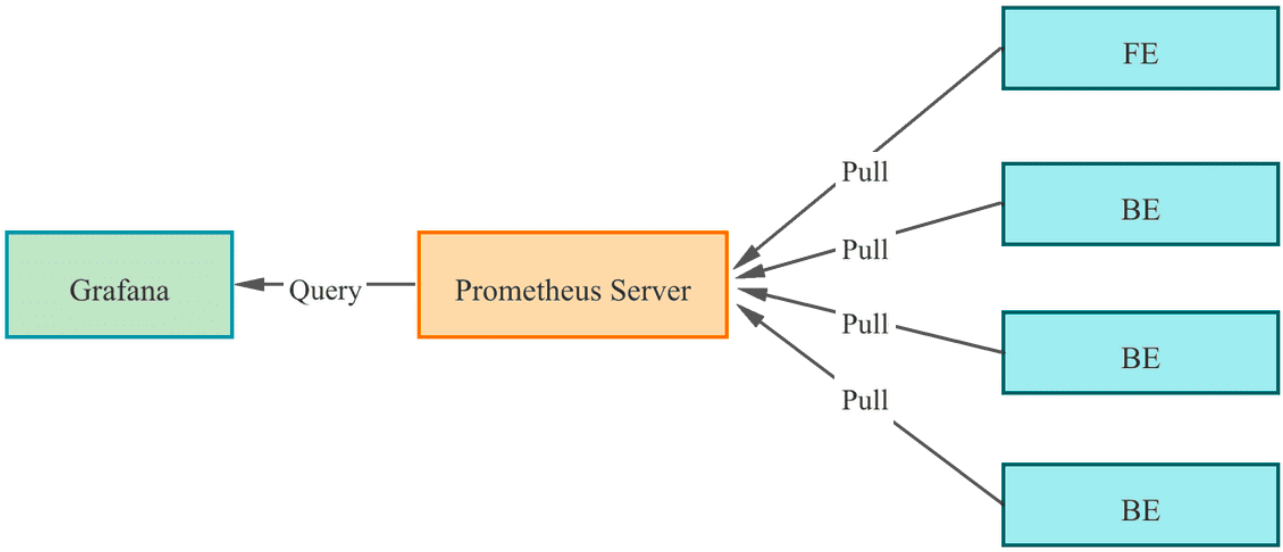
2.2.6 Мониторинг и алертинг с Prometheus и Grafana

Selena предоставляет решение для мониторинга и алертинга с использованием Prometheus и Grafana. Это позволяет визуализировать работу вашего cluster, облегчая мониторинг и устранение неполадок.

Обзор

Selena предоставляет интерфейс сбора информации, совместимый с Prometheus. Prometheus может получать метрики Selena, подключаясь к HTTP-портам узлов BE и FE и сохраняя информацию в своей собственной базе данных временных рядов. Grafana затем может использовать Prometheus в качестве источника данных для визуализации метрик.

Используя шаблоны dashboard, предоставленные Selena, вы можете легко мониторить свой cluster Selena и настроить алерты для него с помощью Grafana.



Выполните следующие шаги для интеграции вашего cluster Selena с Prometheus и Grafana:

1. Установите необходимые компоненты - Prometheus и Grafana.
2. Изучите основные метрики мониторинга Selena.
3. Настройте канал алертов и правила алертов.

Шаг 1: Установка компонентов мониторинга

Порты по умолчанию Prometheus и Grafana не конфликтуют с портами Selena. Однако для production рекомендуется разворачивать их на сервере, отличном от серверов ваших cluster Selena. Это уменьшает риск конфликтов ресурсов и избегает потенциального сбоя алертов из-за аварийного завершения работы сервера.

Кроме того, обратите внимание, что Prometheus и Grafana не могут мониторить доступность своих собственных сервисов. Поэтому в production-окружении рекомендуется использовать Supervisor для настройки heartbeat-сервиса для них.

Следующий tutorial разворачивает компоненты мониторинга на узле мониторинга (IP: 192.168.110.23) с использованием пользователя ОС root. Они мониторят следующий cluster Selena (который использует порты по умолчанию). При настройке сервиса

мониторинга для вашего собственного cluster Selena на основе этого tutorial вам нужно только заменить IP-адреса.

Host	IP	OS user	Services
node01	192.168.110.101	root	1 FE + 1 BE
node02	192.168.110.102	root	1 FE + 1 BE
node03	192.168.110.103	root	1 FE + 1 BE

> **ПРИМЕЧАНИЕ**
>
> Prometheus и Grafana могут мониторить только узлы FE, BE и CN, но не узлы Broker.

1.1 Развертывание Prometheus

1.1.1 Загрузка Prometheus

Для Selena вам нужно только загрузить установочный пакет сервера Prometheus. Загрузите пакет на узел мониторинга.

Нажмите здесь, чтобы загрузить Prometheus.

Возьмем в качестве примера LTS-версию v1.5.2, нажмите на пакет, чтобы загрузить его.

prometheus

The Prometheus monitoring system and time series database. prometheus/prometheus

2.46.0 / 2023-07-25 Release notes				
File name	**OS**	**Arch**	**Size**	**SHA256 Checksum**
prometheus-2.46.0.darwin-amd64.tar.gz	darwin	amd64	90.71 MiB	c4741e960e27410a64ffe278e199e3cbe9df74bfea3ba56bf5ebc161eddb4b56
prometheus-2.46.0.linux-amd64.tar.gz	linux	amd64	90.48 MiB	d2177ea21a6f60046f9510c828d4f8969628cfd35686780b3898917ef9c268b9
prometheus-2.46.0.windows-amd64.zip	windows	amd64	92.48 MiB	b4ae13f26cea5d9e53c32490d61fef159f8f3094239a365cfb146698eceaa021
2.45.0 / 2023-06-23 LTS Release notes				
File name	**OS**	**Arch**	**Size**	**SHA256 Checksum**
prometheus-2.45.0.darwin-amd64.tar.gz	darwin	amd64	87.24 MiB	e65d4ca0676bec55d21947290d51a62126cba829f7a8cec93808a35f17e63174
prometheus-2.45.0.linux-amd64.tar.gz	linux	amd64	86.97 MiB	1c7f489a3cc919c1ed0df2ae673a280309dc4a3eaa6ee3411e7d1f4bdec4d4c5
prometheus-2.45.0.windows-amd64.zip	windows	amd64	88.94 MiB	c50feaf6b9f7bcf3d0caa3575122afcd470f1313824c480e715c4c80eb34f436

Alternatively, вы можете загрузить его с помощью команды `wget`:

```
# Следующий пример загружает LTS-версию v1.5.2.
# Вы можете загрузить другие версии, заменив номер версии в команде.
wget
https://github.com/prometheus/prometheus/releases/download/v1.5.2/prometheus-
2.45.0.linux-amd64.tar.gz
```

После завершения загрузки загрузите или скопируйте установочный пакет в каталог **/opt** на узле мониторинга.

1.1.2 Установка Prometheus

1. Перейдите в **/opt** и распакуйте установочный пакет Prometheus.

   ```
   cd /opt
   tar xvf prometheus-2.45.0.linux-amd64.tar.gz
   ```

2. Для удобства управления переименуйте распакованный каталог в **prometheus**.

   ```
   mv prometheus-2.45.0.linux-amd64 prometheus
   ```

3. Создайте путь для хранения данных для Prometheus.

   ```
   mkdir prometheus/data
   ```

4. Для удобства управления вы можете создать файл запуска системного сервиса для Prometheus.

   ```
   vim /etc/systemd/system/prometheus.service
   ```

   Добавьте следующее содержимое в файл:

```
[Unit]
Description=Prometheus service
After=network.target

[Service]
User=root
Type=simple
ExecReload=/bin/sh -c "/bin/kill -1 `/usr/bin/pgrep prometheus`"
ExecStop=/bin/sh -c "/bin/kill -9 `/usr/bin/pgrep prometheus`"
ExecStart=/opt/prometheus/prometheus --
config.file=/opt/prometheus/prometheus.yml --
storage.tsdb.path=/opt/prometheus/data --
storage.tsdb.retention.time=30d --storage.tsdb.retention.size=30GB

[Install]
WantedBy=multi-user.target
```

Затем сохраните и выйдите из редактора.

> **ПРИМЕЧАНИЕ**
>
> Если вы разворачиваете Prometheus по другому пути, убедитесь, что синхронизировали путь в команде ExecStart в файле выше. Кроме того, файл настраивает условия истечения срока для хранения данных Prometheus как "30 дней или более" или "более 30 ГБ". Вы можете изменить это в соответствии с вашими потребностями.

5. Измените конфигурационный файл Prometheus **prometheus/prometheus.yml**. Этот файл имеет строгие требования к формату содержимого. Пожалуйста, обращайте особое внимание на пробелы и отступы при внесении изменений.

```
vim prometheus/prometheus.yml
```

Добавьте следующее содержимое в файл:

```
global:
  scrape_interval: 15s # Установите глобальный интервал scrape равным 15s. По умолчанию 1 мин.
  evaluation_interval: 15s # Установите глобальный интервал evaluation правил равным 15s. По умолчанию 1 мин.
scrape_configs:
  - job_name: 'Selena_Cluster01' # Мониторимый cluster соответствует job. Вы можете настроить имя cluster Selena здесь.
    metrics_path: '/metrics'    # Укажите Restful API для получения метрик мониторинга.
    static_configs:
    # Следующая конфигурация указывает группу FE, которая включает 3 узла FE.
    # Здесь вам нужно заполнить IP и HTTP-порты, соответствующие каждому FE.
    # Если вы изменили HTTP-порты во время развертывания cluster, обязательно настройте их соответственно.
      - targets: ['192.168.110.101:8030','192.168.110.102:8030','192.168.110.103:8030']
        labels:
          group: fe
    # Следующая конфигурация указывает группу BE, которая включает 3 узла BE.
    # Здесь вам нужно заполнить IP и HTTP-порты, соответствующие каждому BE.
    # Если вы изменили HTTP-порты во время развертывания cluster, обязательно настройте их соответственно.
      - targets: ['192.168.110.101:8040','192.168.110.102:8040','192.168.110.103:8040']
        labels:
          group: be
```

ПРИМЕЧАНИЕ

Обратите внимание, что Prometheus не может обнаружить изменения сервиса (`targets`) после масштабирования cluster. Например, для cluster, развернутых на AWS, вы можете предоставить инстанцу EC2, который размещает сервис Prometheus, разрешения `ec2:DescribeInstances` и `ec2:DescribeTags`, и добавить свойства `ec2_sd_configs` и `relabel_configs` в **prometheus/prometheus.yml**. Для подробных инструкций см. Приложение - Включение обнаружения сервисов для Prometheus.

После того, как вы изменили конфигурационный файл, вы можете использовать `promtool` для проверки правильности изменений.

```
./prometheus/promtool check config prometheus/prometheus.yml
```

Следующее сообщение указывает, что проверка пройдена. Вы можете продолжить.

```
SUCCESS: prometheus/prometheus.yml is valid prometheus config file
syntax
```

6. Запустите Prometheus.

```
systemctl daemon-reload
systemctl start prometheus.service
```

7. Проверьте статус Prometheus.

```
systemctl status prometheus.service
```

Если возвращается `Active: active (running)`, это указывает, что Prometheus успешно запущен.

Вы также можете использовать `netstat` для проверки статуса порта Prometheus по умолчанию (9090).

```
netstat -nltp | grep 9090
```

8. Настройте Prometheus на запуск при загрузке.

```
systemctl enable prometheus.service
```

Другие команды:

- Остановить Prometheus.

```
systemctl stop prometheus.service
```

- Перезапустить Prometheus.

```
systemctl restart prometheus.service
```

- Перезагрузить конфигурацию во время работы.

```
systemctl reload prometheus.service
```

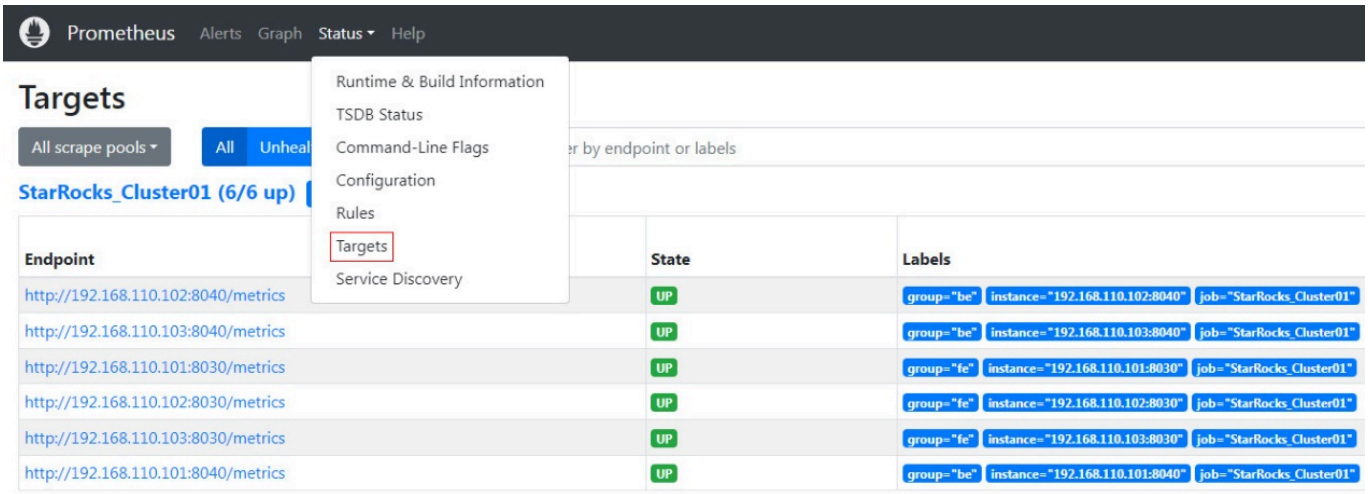
- Отключить запуск при загрузке.

```
systemctl disable prometheus.service
```

1.1.3 Доступ к Prometheus

Вы можете получить доступ к Web UI Prometheus через браузер, порт по умолчанию - 9090. Для узла мониторинга в этом tutorial вам нужно посетить `192.168.110.23:9090`.

На домашней странице Prometheus перейдите в **Status** --> **Targets** в верхнем меню. Здесь вы можете увидеть все мониторимые узлы для каждой группы job, настроенной в файле **prometheus.yml**. Обычно статус всех узлов должен быть UP, указывая, что коммуникация сервиса нормальная.



На этом этапе Prometheus настроен и установлен. Для более подробной информации вы можете обратиться к документации Prometheus.

1.2 Развертывание Grafana

1.2.1 Загрузка Grafana

Нажмите здесь, чтобы загрузить Grafana.

Alternatively, вы можете использовать команду `wget` для загрузки установочного пакета Grafana RPM.

```
# Следующий пример загружает LTS-версию v10.0.3.
# Вы можете загрузить другие версии, заменив номер версии в команде.
wget https://dl.grafana.com/enterprise/release/grafana-enterprise-10.0.3-
1.x86_64.rpm
```

1.2.2 Установка Grafana

1. Используйте команду `yum` для установки Grafana. Эта команда автоматически установит зависимости, необходимые для Grafana.

   ```
   yum -y install grafana-enterprise-10.0.3-1.x86_64.rpm
   ```

2. Запустите Grafana.

   ```
   systemctl start grafana-server.service
   ```

3. Проверьте статус Grafana.

   ```
   systemctl status grafana-server.service
   ```

   Если возвращается `Active: active (running)`, это указывает, что Grafana успешно запущена.

   Вы также можете использовать `netstat` для проверки статуса порта Grafana по умолчанию (3000).

```
netstat -nltp | grep 3000
```

4. Настройте Grafana на запуск при загрузке.

```
systemctl enable grafana-server.service
```

Другие команды:

- Остановить Grafana.

```
systemctl stop grafana-server.service
```

- Перезапустить Grafana.

```
systemctl restart grafana-server.service
```

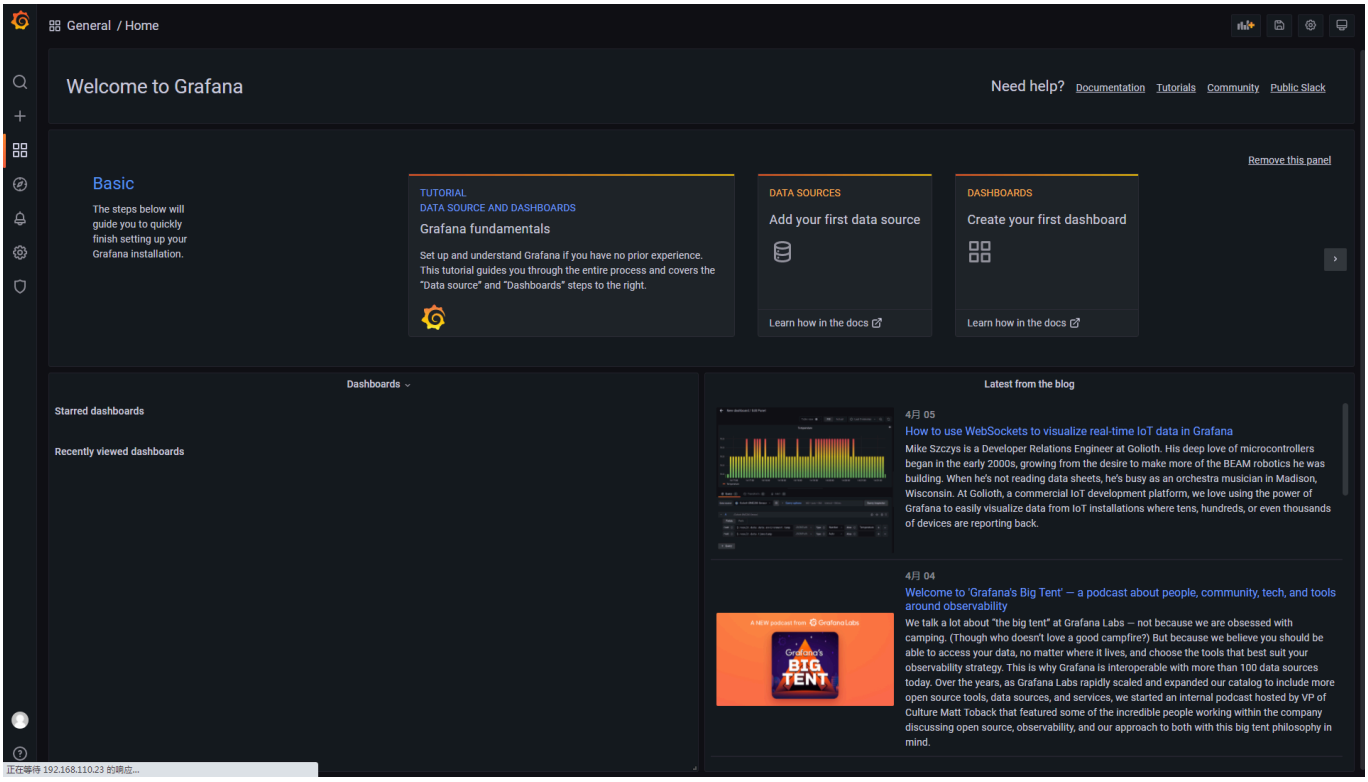
- Отключить запуск при загрузке.

```
systemctl disable grafana-server.service
```

Для дополнительной информации обратитесь к документации Grafana.

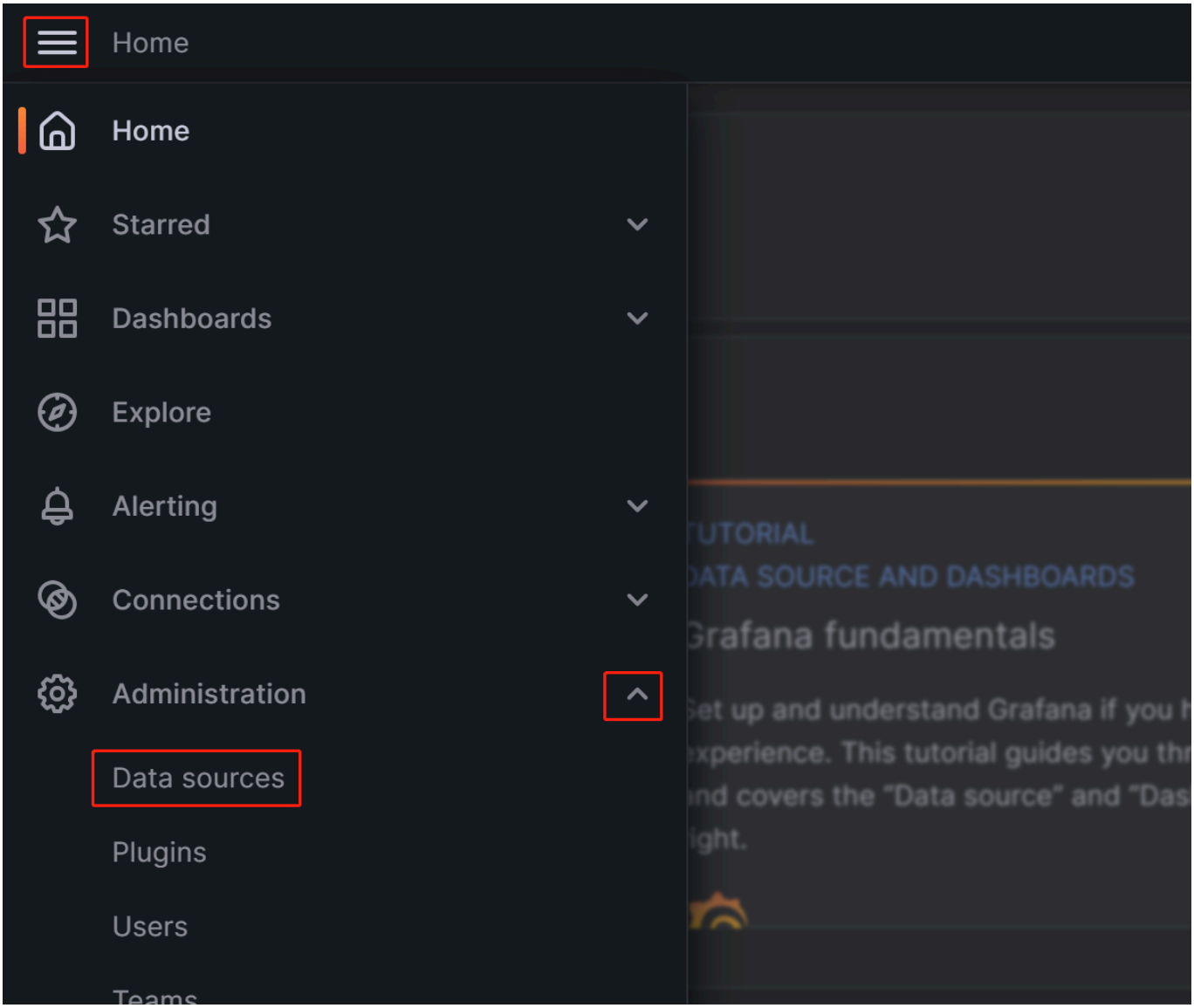
1.2.3 Доступ к Grafana

Вы можете получить доступ к Web UI Grafana через браузер, порт по умолчанию - 3000. Для узла мониторинга в этом tutorial вам нужно посетить `192.168.110.23:3000`. Имя пользователя и пароль по умолчанию, необходимые для входа, оба установлены на `admin`. При первом входе Grafana предложит вам изменить пароль по умолчанию. Если вы хотите пропустить это сейчас, вы можете нажать `Skip`. Затем вы будете перенаправлены на домашнюю страницу Web UI Grafana.

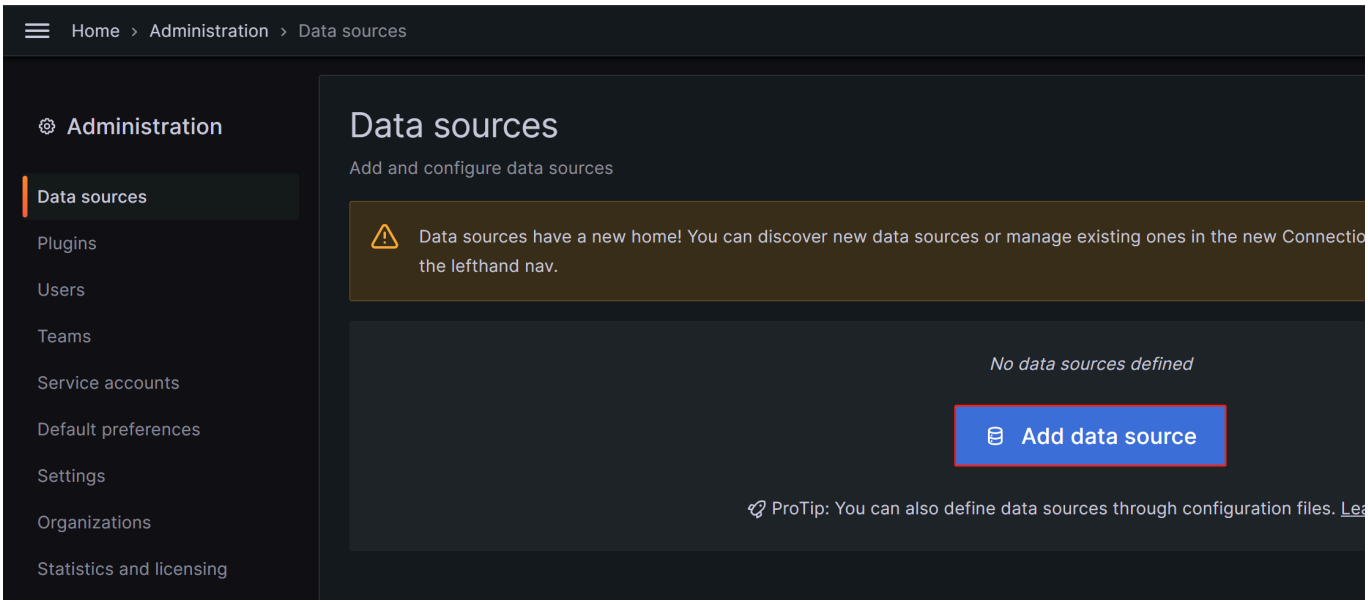


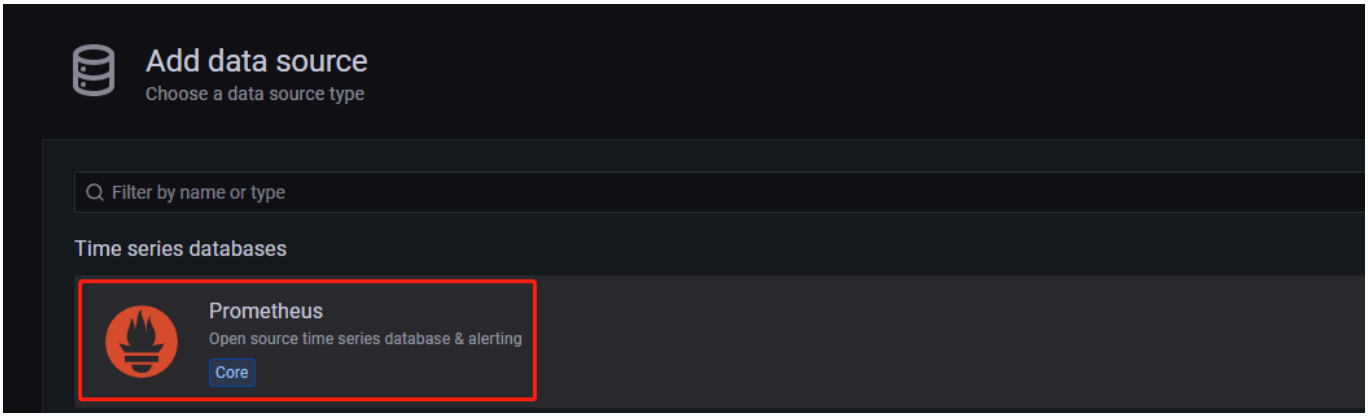
1.2.4 Настройка источников данных

Нажмите на кнопку меню в верхнем левом углу, разверните **Administration**, а затем нажмите **Data sources**.



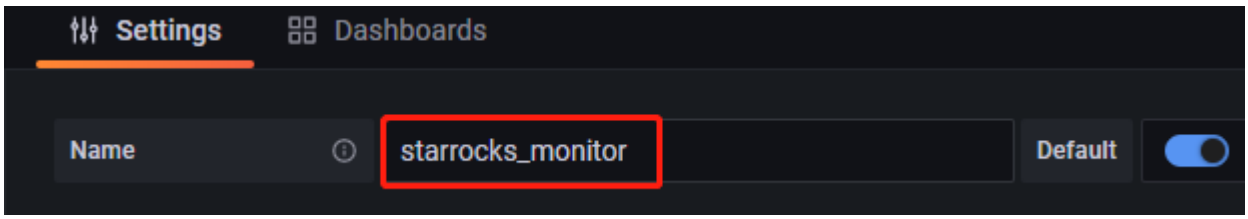
На появившейся странице нажмите **Add data source**, а затем выберите **Prometheus**.



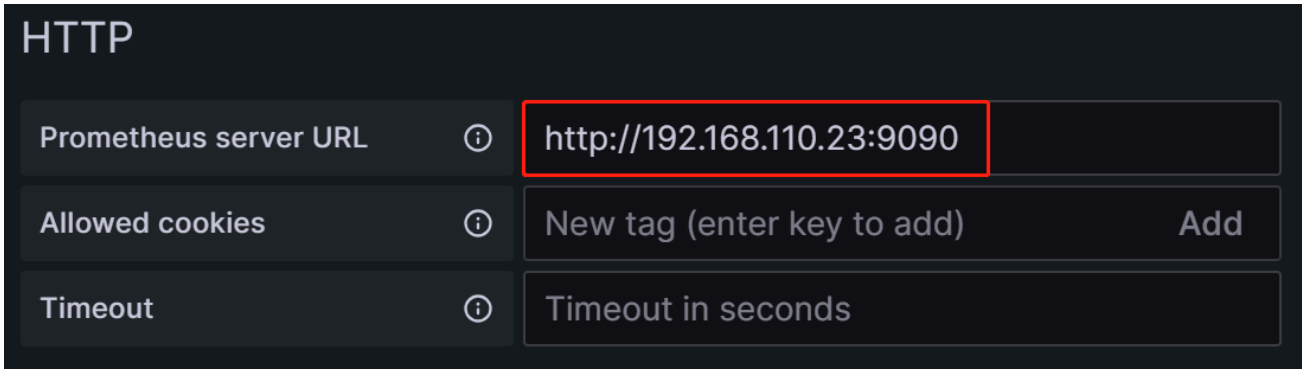


Чтобы интегрировать Grafana с вашим сервисом Prometheus, вам необходимо изменить следующую конфигурацию:

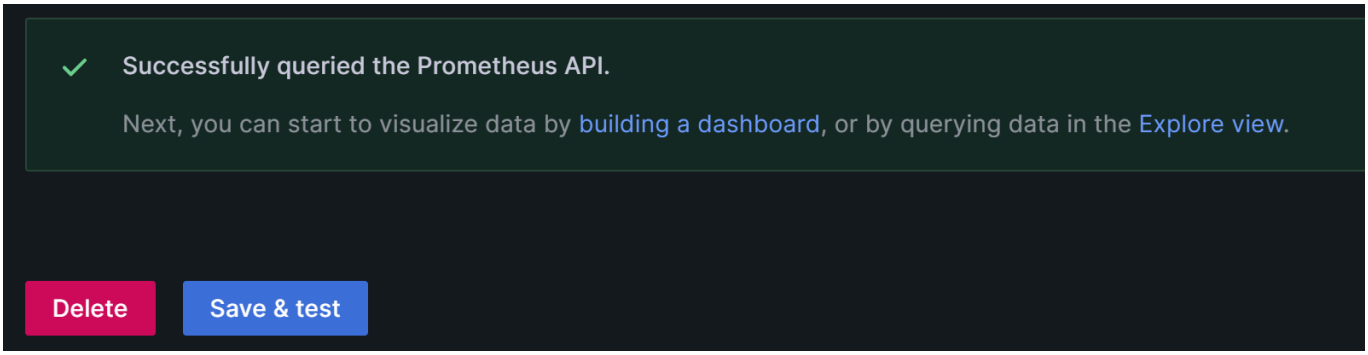
- **Name**: Имя источника данных. Вы можете настроить имя для источника данных.



- **Prometheus Server URL**: URL сервера Prometheus, который в этом tutorial является `http://192.168.110.23:9090`.



После завершения конфигурации нажмите **Save & Test**, чтобы сохранить и проверить конфигурацию. Если отображается **Successfully queried the Prometheus API**, это означает, что источник данных доступен.



1.2.5 Настройка Dashboard

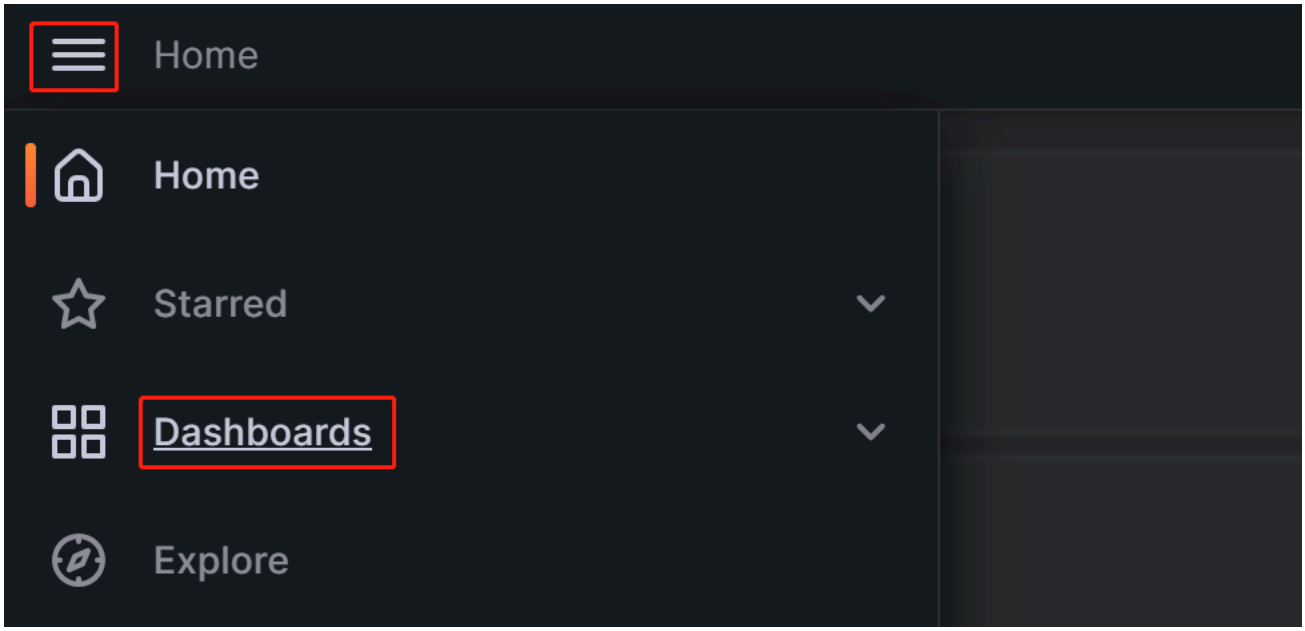
1. Загрузите соответствующий шаблон Dashboard на основе вашей версии Selena.

    - Шаблон Dashboard для всех архитектур
    - Шаблон Dashboard для Shared-data Cluster - General
    - Шаблон Dashboard для Shared-data Cluster - Starlet

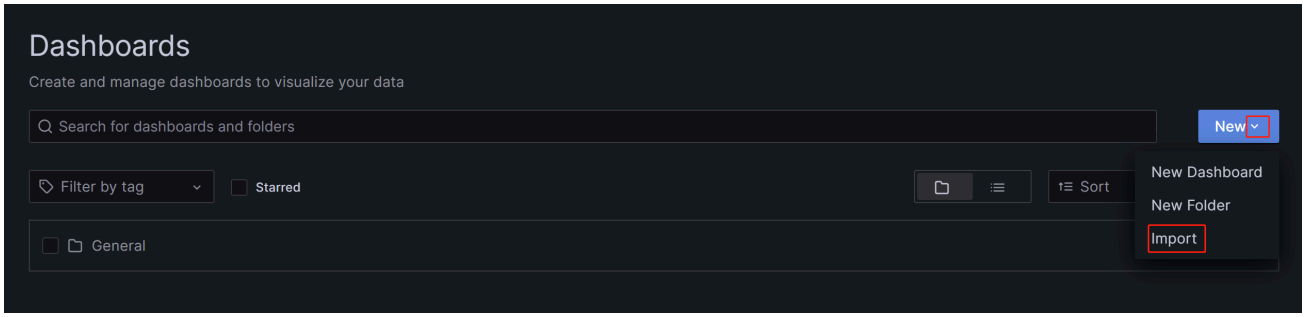
    > **ПРИМЕЧАНИЕ**
    >
    > Файл шаблона необходимо загрузить через Web UI Grafana. Поэтому вам нужно загрузить файл шаблона на машину, которую вы используете для доступа к Grafana, а не на сам узел мониторинга.

2. Настройте шаблон Dashboard.

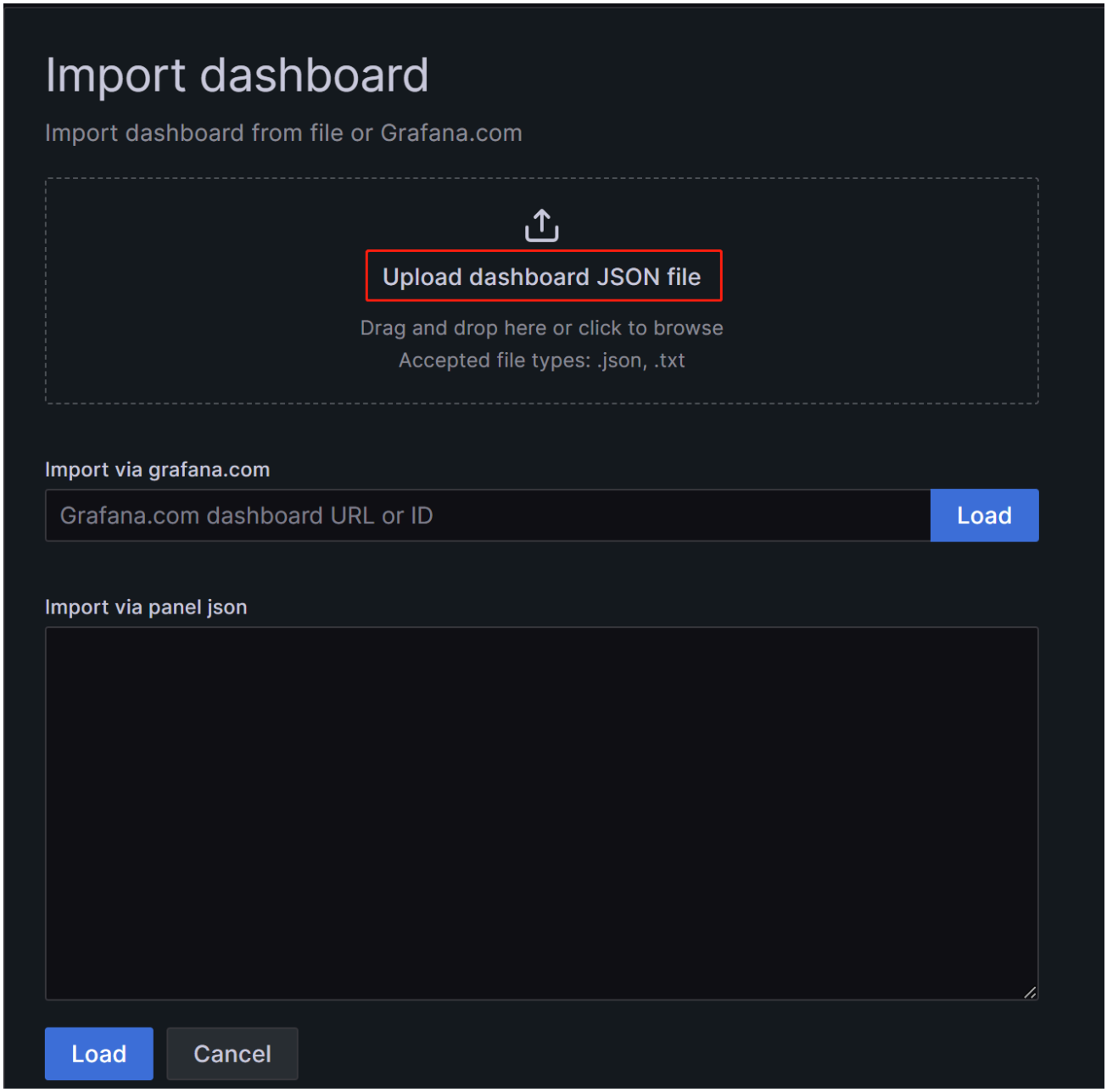
    Нажмите на кнопку меню в верхнем левом углу и нажмите **Dashboards**.

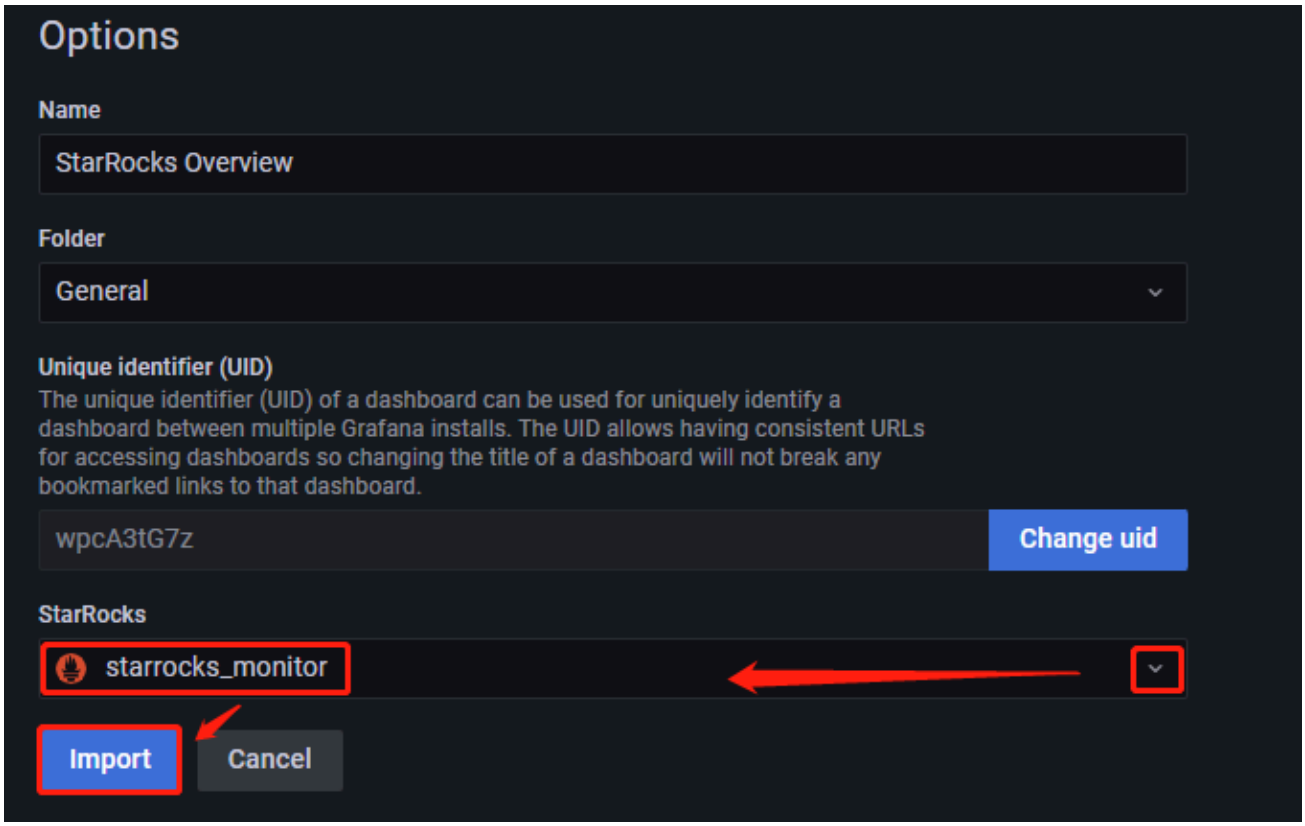
    На появившейся странице разверните кнопку **New** и нажмите **Import**.



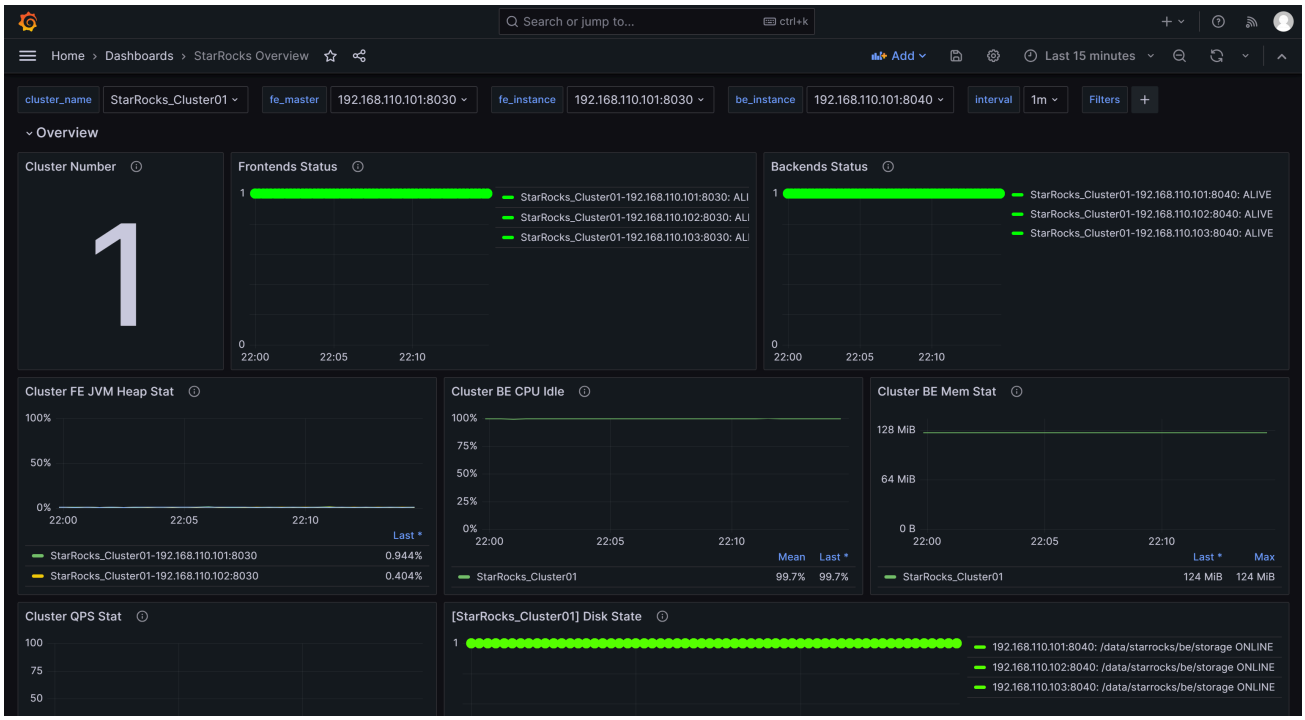
На новой странице нажмите на **Upload Dashboard JSON file** и загрузите файл шаблона, который вы загрузили ранее.



После загрузки файла вы можете переименовать Dashboard. По умолчанию он назван `Selena Overview`. Затем выберите источник данных, который вы создали ранее (`selena_monitor`). Затем нажмите **Import**.

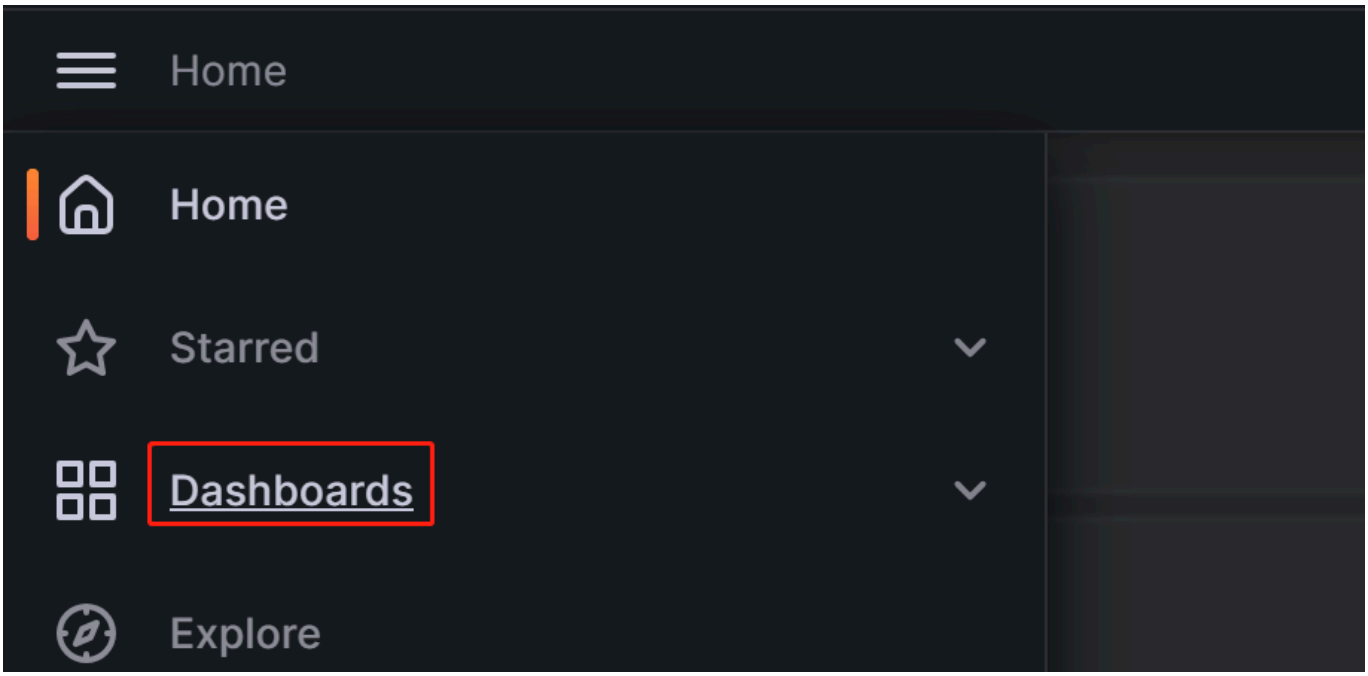


После завершения импорта вы должны увидеть отображенный Dashboard Selena.

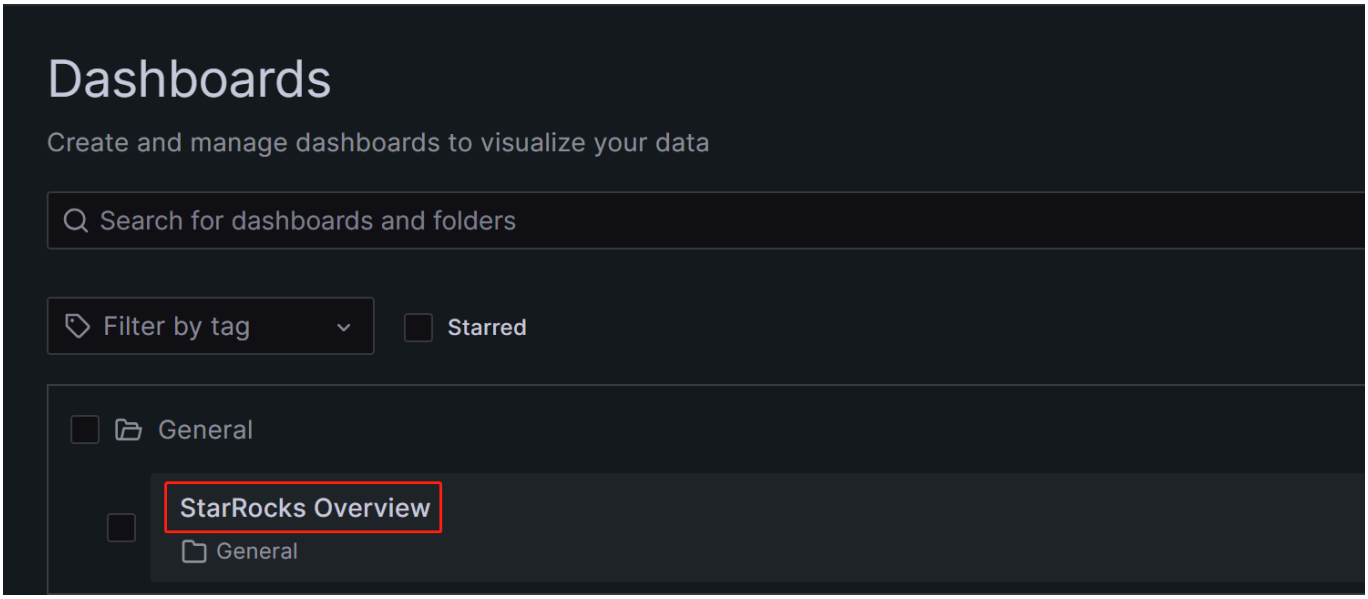


1.2.6 Мониторинг Selena через Grafana

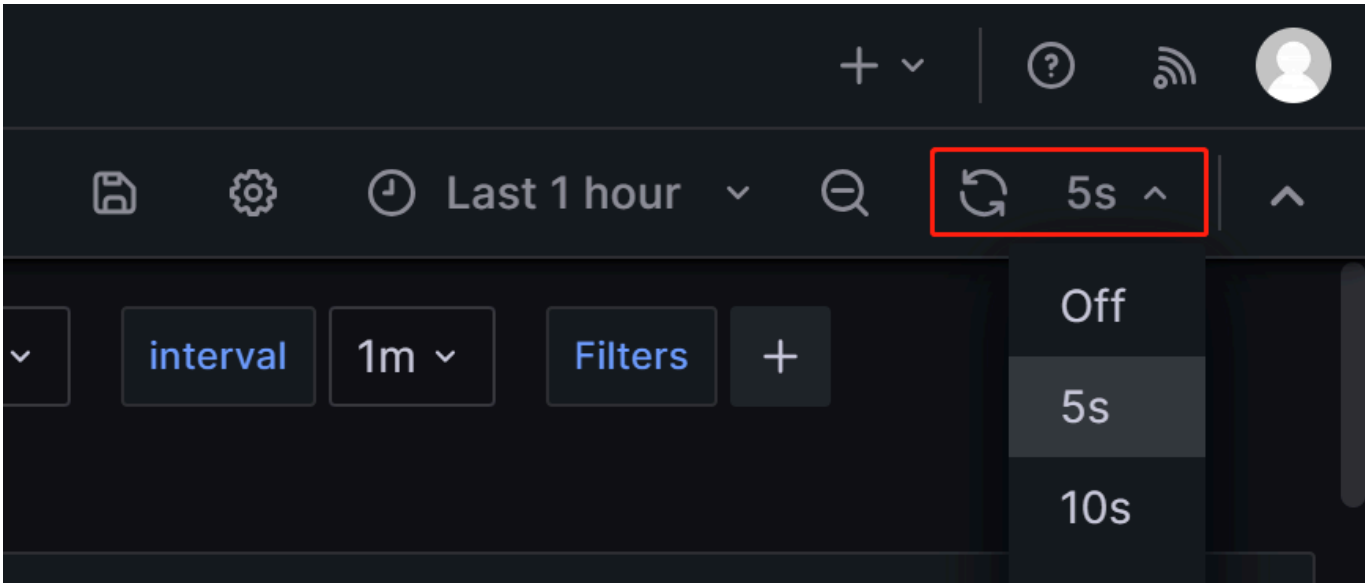
Войдите в Web UI Grafana, нажмите на кнопку меню в верхнем левом углу и нажмите **Dashboards**.



На появившейся странице выберите **Selena Overview** из каталога **General**.



После входа в Dashboard мониторинга Selena вы можете вручную обновить страницу в верхнем правом углу или установить интервал автоматического обновления для мониторинга статуса cluster Selena.



Шаг 2: Понимание основных метрик мониторинга

Чтобы удовлетворить потребности разработки, operations, DBA и других, Selena предоставляет широкий спектр метрик мониторинга. В этом разделе представлены только некоторые важные метрики, обычно используемые в бизнесе, и их правила алертов. Для других деталей метрик, пожалуйста, обратитесь к Метрики мониторинга.

2.1 Метрики для статуса FE и BE

Metric	Description	Alert rule	Note
Frontends Status	Статус узла FE. Статус активного узла представлен как `1`, в то время как узел, который выключен (DEAD), будет отображаться как `0`.	Статус всех узлов FE должен быть активным, и любой узел FE со статусом DEAD должен вызывать алерт.	Сбой любых узлов FE или BE считается критическим и требует немедленного устранения неполадок для выявления причины сбоя.
Backends Status	Статус узла BE. Статус активного узла представлен как `1`, в то время как узел, который выключен (DEAD), будет отображаться как `0`.	Статус всех узлов BE должен быть активным, и любой узел BE со статусом DEAD должен вызывать алерт.	

2.2 Метрики для сбоев запросов

Metric	Description	Alert rule	Note
Query	Частота сбоев запросов (включая timeout) в	Вы можете настроить это на основе фактического QPS	Обычно частота сбоев запросов должна быть низкой. Установка этого порога на 0.05

Metric	Description	Alert rule	Note
Error	течение одной минуты. Его значение рассчитывается как количество неудачных запросов за одну минуту, деленное на 60 секунд.	вашего бизнеса. Например, 0.05 можно использовать в качестве предварительной настройки. Вы можете настроить это позже по мере необходимости.	означает допущение максимум 3 неудачных запросов в минуту. Если вы получите алерт от этого элемента, вы можете проверить использование ресурсов или соответствующим образом настроить timeout запроса.

2.3 Метрики для сбоев внешних операций

Metric	Description	Alert rule	Note
Schema Change	Частота сбоев операции Schema Change.	Schema Change - это низкочастотная операция. Вы можете настроить этот элемент для отправки алерта немедленно при сбое.	Обычно операции Schema Change не должны завершаться сбоем. Если алерт запускается для этого элемента, вы можете рассмотреть возможность увеличения лимита памяти операций Schema Change, который по умолчанию установлен на 2 ГБ.

2.4 Метрики для сбоев внутренних операций

Metric	Description	Alert rule	Note
BE Compaction Score	Наивысший Compaction Score среди всех узлов BE, указывающий на текущее давление compaction.	В типичных offline-сценариях это значение обычно ниже 100. Однако при большом количестве задач загрузки Compaction Score может значительно увеличиться. В большинстве случаев требуется вмешательство, когда это значение превышает 800.	Обычно, если Compaction Score больше 1000, Selena вернет ошибку "Too many versions". В таких случаях вы можете рассмотреть возможность уменьшения concurrency и частоты загрузки.
Clone	Частота сбоев операции клонирования tablet.	Вы можете настроить этот элемент для отправки алерта немедленно при сбое.	Если алерт запускается для этого элемента, вы можете проверить статус узлов BE, статус диска и статус сети.

2.5 Метрики для доступности сервиса

Metric	Description	Alert rule	Note
Meta Log	Количество записей лога метаданных	Рекомендуется настроить этот элемент для немедленного	По умолчанию узел leader FE запускает checkpoint для сброса лога на диск, когда количество логов превышает 50 000. Если это значение превышает 50 000 на большой запас, это обычно

Metric	Description	Alert rule	Note
Count	BDB на узле FE.	алерта, если он превышает 100 000.	указывает на сбой checkpoint. Вы можете проверить, является ли конфигурация heap-памяти Xmx разумной в **fe.conf**.

2.6 Метрики для нагрузки системы

Metric	Description	Alert rule	Note
BE CPU Idle	Процент простоя CPU узла BE.	Рекомендуется настроить этот элемент для алерта, если процент простоя ниже 10% в течение 30 последовательных секунд.	Этот элемент используется для мониторинга узких мест ресурсов CPU. Использование CPU может значительно колебаться, и установка небольшого интервала опроса может привести к ложным алертам. Поэтому вам нужно настроить этот элемент на основе фактических условий бизнеса. Если у вас есть несколько задач пакетной обработки или большое количество запросов, вы можете рассмотреть возможность установки более низкого порога.
BE Mem	Использование памяти для узла BE.	Рекомендуется настроить этот элемент на 90% от доступного размера памяти для каждого BE.	Это значение эквивалентно значению Process Mem, и лимит памяти BE по умолчанию составляет 90% от размера памяти сервера (контролируется конфигурацией `mem_limit` в **be.conf**). Если вы развернули другие сервисы на том же сервере, обязательно настройте это значение, чтобы избежать OOM. Порог алерта для этого элемента должен быть установлен на 90% от фактического лимита памяти BE, чтобы вы могли подтвердить, достигли ли ресурсы памяти BE узкого места.
Disks Avail Capacity	Соотношение доступного дискового пространства (процент) локальных дисков на каждом узле BE.	Рекомендуется настроить этот элемент для алерта, если значение меньше 20%.	Рекомендуется зарезервировать достаточное доступное пространство для Selena на основе ваших бизнес-требований.
FE JVM Heap Stat	Процент использования heap-памяти JVM для каждого узла FE в cluster.	Рекомендуется настроить этот элемент для алерта, если значение больше или равно 80%.	Если алерт запускается для этого элемента, рекомендуется увеличить конфигурацию heap-памяти Xmx в **fe.conf**; в противном случае это может повлиять на эффективность запросов или привести к проблемам FE OOM.

Шаг 3: Настройка алертов через Email

3.1 Настройка SMTP-сервиса

Grafana поддерживает различные решения для алертов, такие как email и webhooks. Этот tutorial использует email в качестве примера.

Чтобы включить алерты по email, сначала вам нужно настроить информацию SMTP в Grafana, позволяя Grafana отправлять email на ваш почтовый ящик. Большинство часто используемых почтовых провайдеров поддерживают SMTP-сервисы, и вам нужно включить SMTP-сервис для вашей учетной записи email и получить код авторизации.

После выполнения этих шагов измените конфигурационный файл Grafana на узле, где развернута Grafana.

```
vim /usr/share/grafana/conf/defaults.ini
```

Пример:

```
##################### SMTP / Emailing #####################
[smtp]
enabled = true
host = <smtp_server_address_and_port>
user = johndoe@gmail.com
# If the password contains # or ; you have to wrap it with triple quotes.Ex
"""#password;"""
password = ABCDEFGHIJKLMNOP  # Пароль авторизации, полученный после
включения SMTP.
cert_file =
key_file =
skip_verify = true  ## Проверить SSL для SMTP-сервера
from_address = johndoe@gmail.com  ## Адрес, используемый при отправке
email.
from_name = Grafana
ehlo_identity =
startTLS_policy =

[emails]
welcome_email_on_sign_up = false
templates_pattern = emails/*.html, emails/*.txt
content_types = text/html
```

Вам необходимо изменить следующие элементы конфигурации:

- `enabled`: Разрешить ли Grafana отправлять email-алерты. Установите этот элемент на `true`.
- `host`: Адрес и порт SMTP-сервера для вашего email, разделенные двоеточием (`:`). Пример: `smtp.gmail.com:465`.
- `user`: Имя пользователя SMTP.
- `password`: Пароль авторизации, полученный после включения SMTP.
- `skip_verify`: Пропускать ли проверку SSL для SMTP-сервера. Установите этот элемент на `true`.
- `from_address`: Адрес email, используемый для отправки email-алертов.

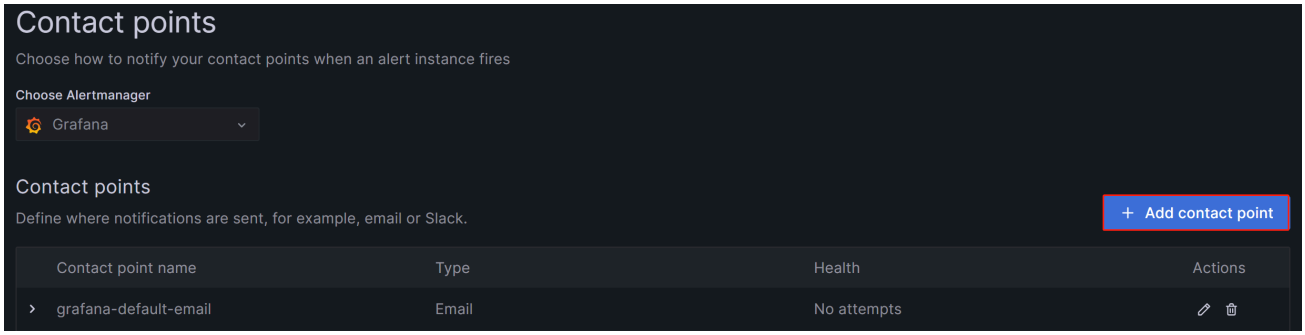
После завершения конфигурации перезапустите Grafana.

```
systemctl daemon-reload
systemctl restart grafana-server.service
```

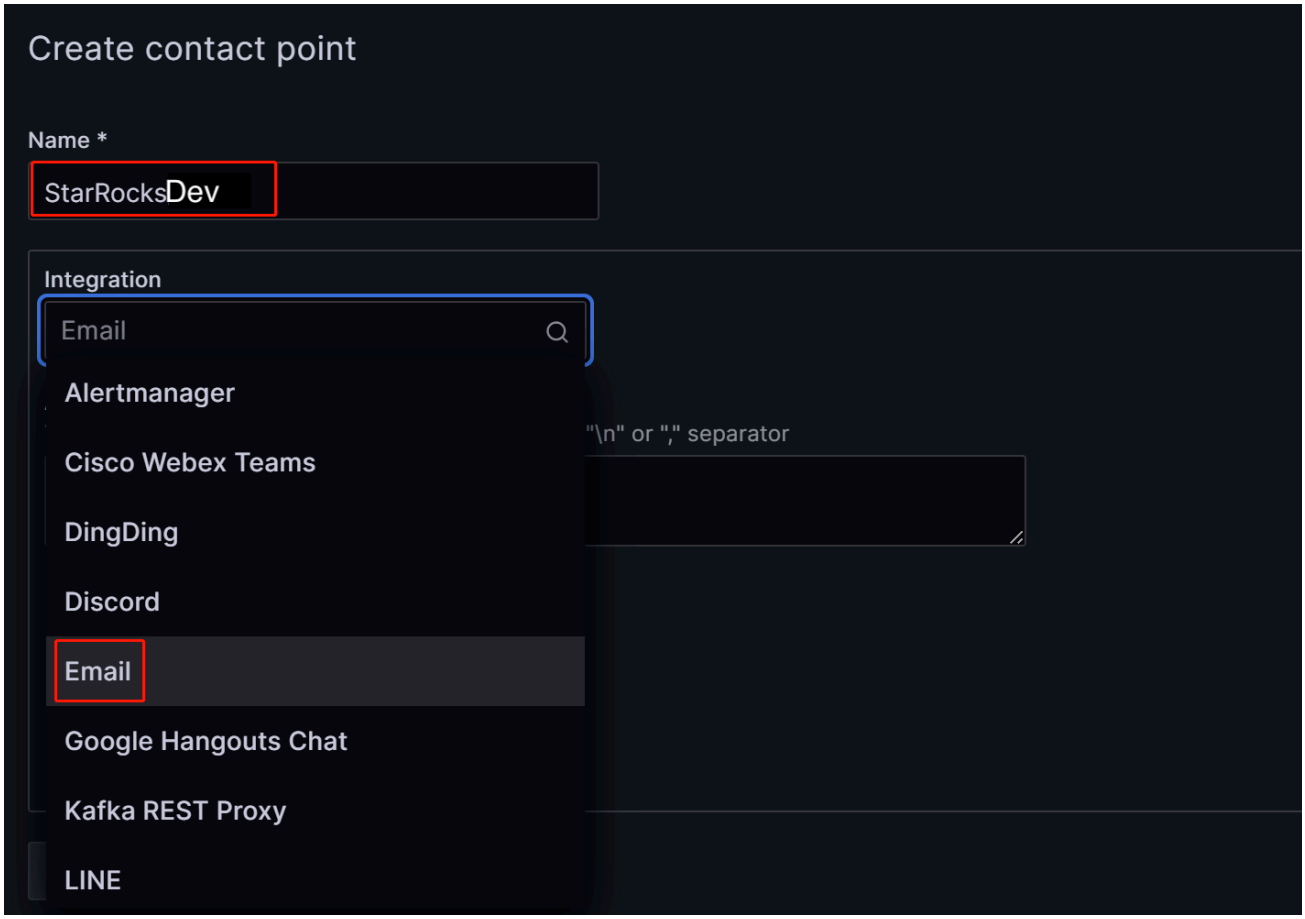
3.2 Создание канала алертов

Вам нужно создать канал алертов (Contact Point) в Grafana, чтобы указать, как уведомлять контакты, когда запускается алерт.

1. Войдите в Web UI Grafana, нажмите на кнопку меню в верхнем левом углу, разверните **Alerting** и выберите **Contact Points**. На странице **Contact points** нажмите **Add contact point**, чтобы создать новый канал алертов.



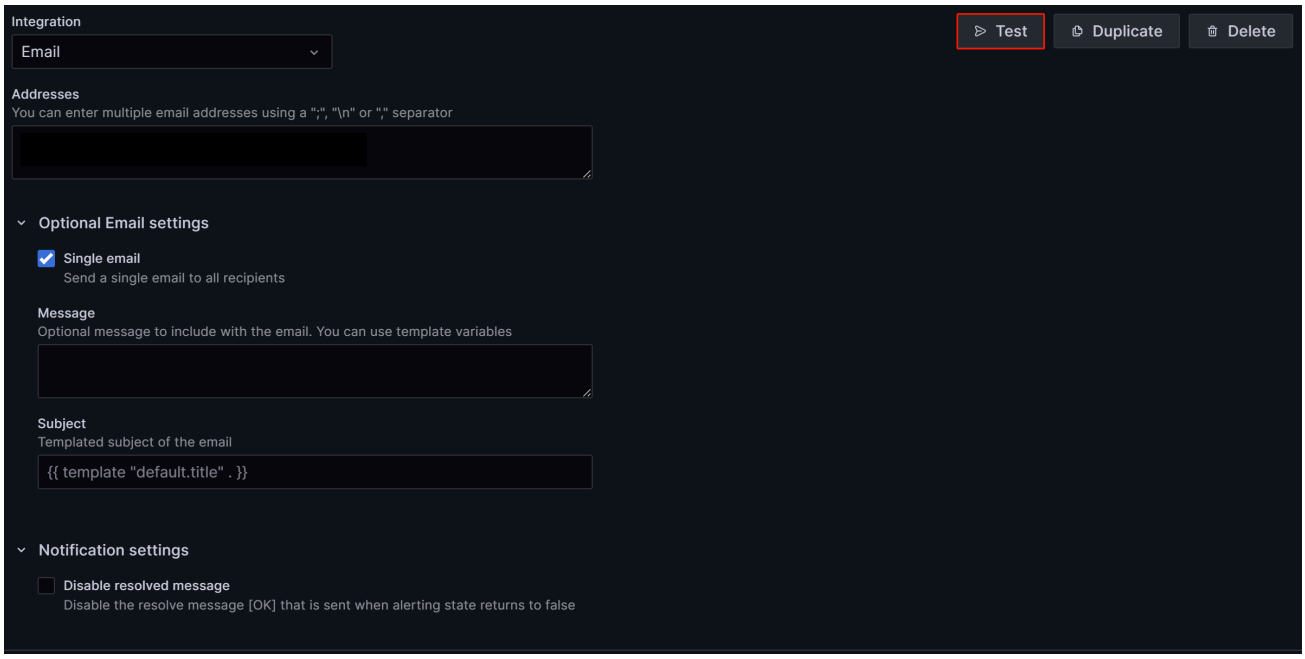
2. В поле **Name** настройте имя точки контакта. Затем в выпадающем списке **Integration** выберите **Email**.

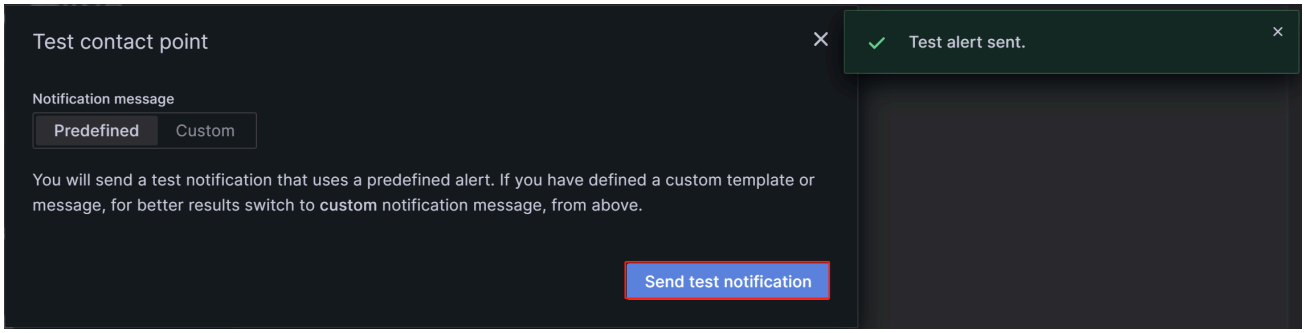


3. В поле **Addresses** введите адреса email контактов для получения алерта. Если есть несколько адресов email, разделите адреса точками с запятой (`;`), запятыми (`,`) или переносами строк.

   Конфигурации на странице можно оставить со значениями по умолчанию, за исключением следующих двух элементов:

   - **Single email**: При включении, если есть несколько контактов, алерт будет отправлен им через один email. Рекомендуется включить этот элемент.
   - **Disable resolved message**: По умолчанию, когда проблема, вызвавшая алерт, решена, Grafana отправляет другое уведомление о восстановлении сервиса. Если вам не нужно это уведомление о восстановлении, вы можете отключить этот элемент. Не рекомендуется отключать эту опцию.
4. После завершения конфигурации нажмите кнопку **Test** в верхнем правом углу страницы. В появившемся сообщении нажмите **Sent test notification**. Если ваш SMTP-сервис и конфигурация адреса правильные, целевая учетная запись email должна получить тестовый email с темой "TestAlert Grafana". После подтверждения, что вы можете успешно получить тестовый email алерта, нажмите кнопку **Save contact point** внизу страницы, чтобы завершить конфигурацию.





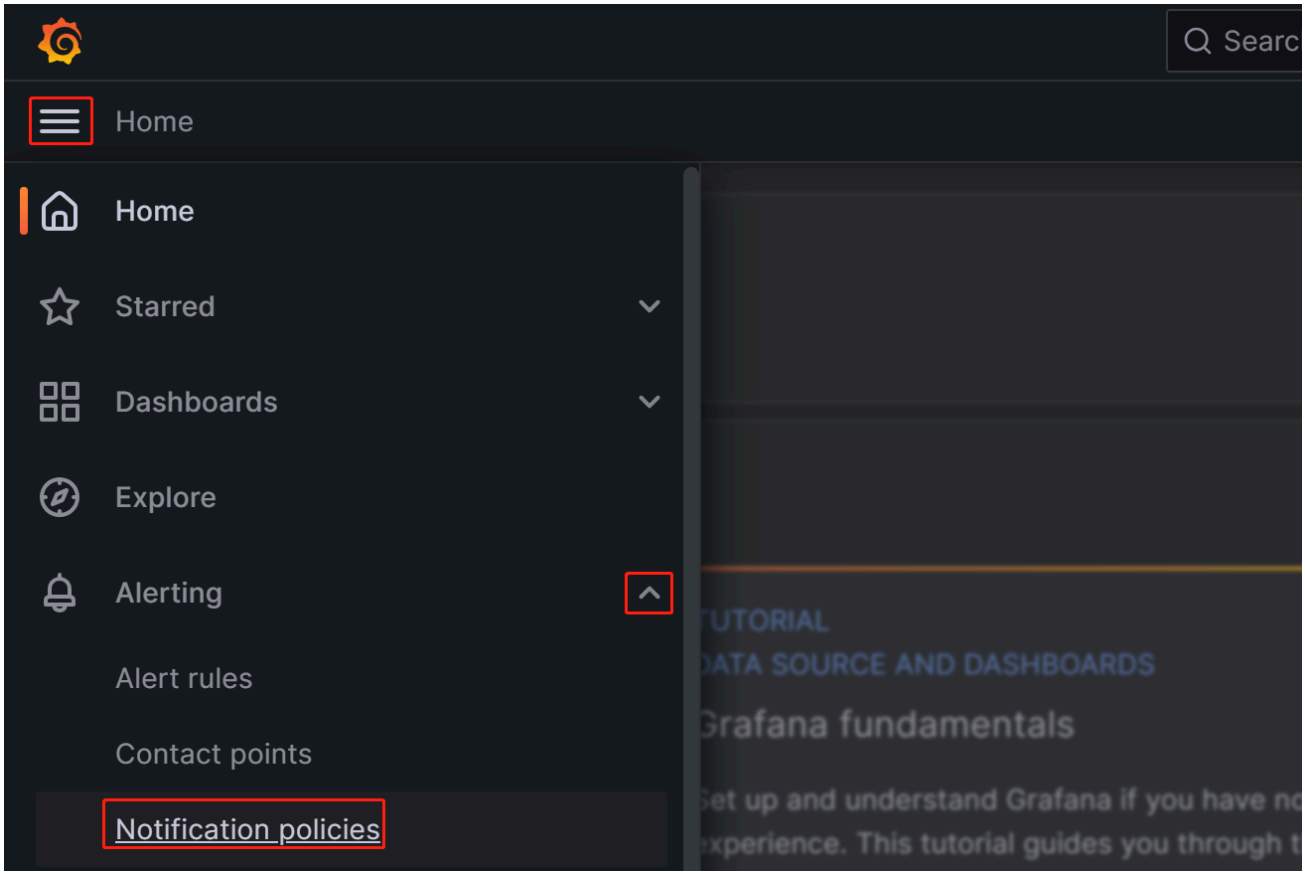
Вы можете настроить несколько методов уведомлений для каждой точки контакта через "Add contact point integration", что не будет подробно рассмотрено здесь. Для более подробной информации о Contact Points вы можете обратиться к документации Grafana

Для последующей демонстрации предположим, что на этом шаге вы создали две точки контакта, "SelenaDev" и "SelenaOp", используя разные адреса email.

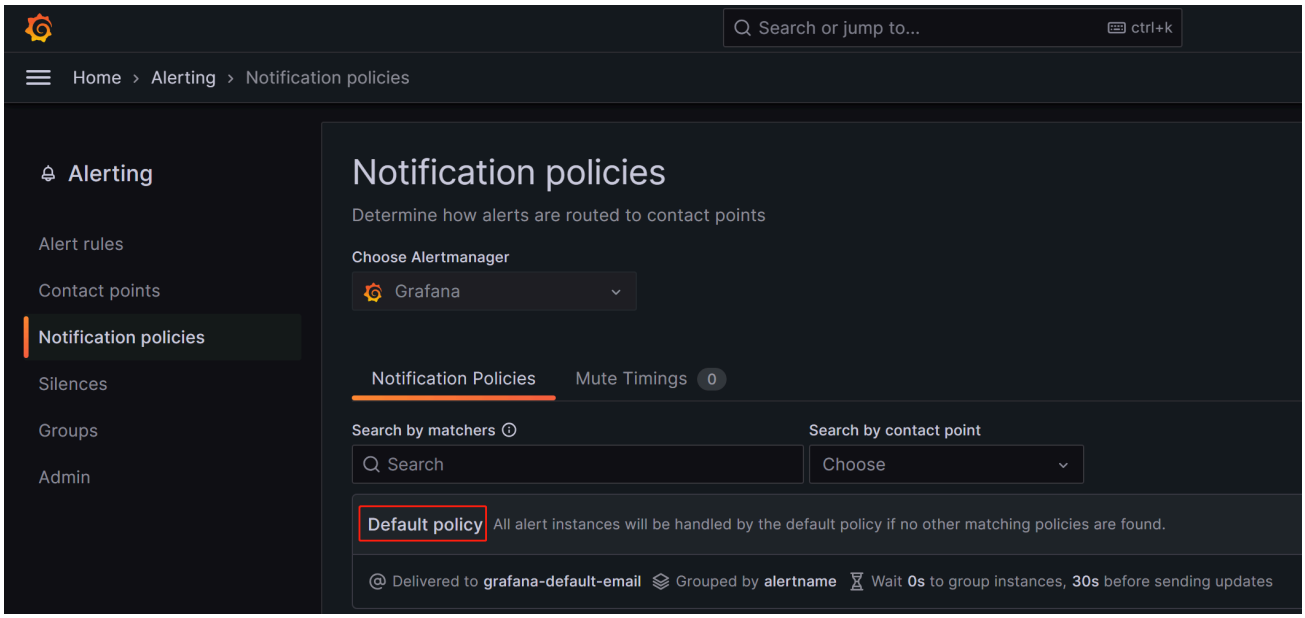
3.3 Установка политик уведомлений

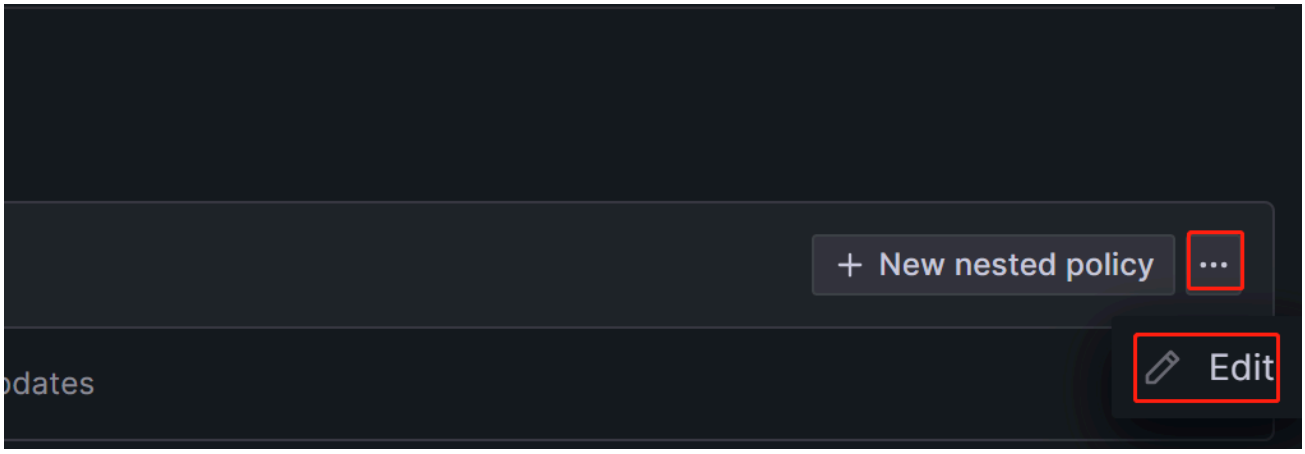
Grafana использует политики уведомлений для связывания точек контакта с правилами алертов. Политики уведомлений используют соответствующие labels для обеспечения гибкого способа маршрутизации различных алертов разным контактам, позволяя группировку алертов во время O&M.

1. Войдите в Web UI Grafana, нажмите на кнопку меню в верхнем левом углу, разверните **Alerting** и выберите **Notification policies**.



2. На странице **Notification policies** нажмите значок more (**...**) справа от **Default policy** и нажмите **Edit**, чтобы изменить политику по умолчанию.

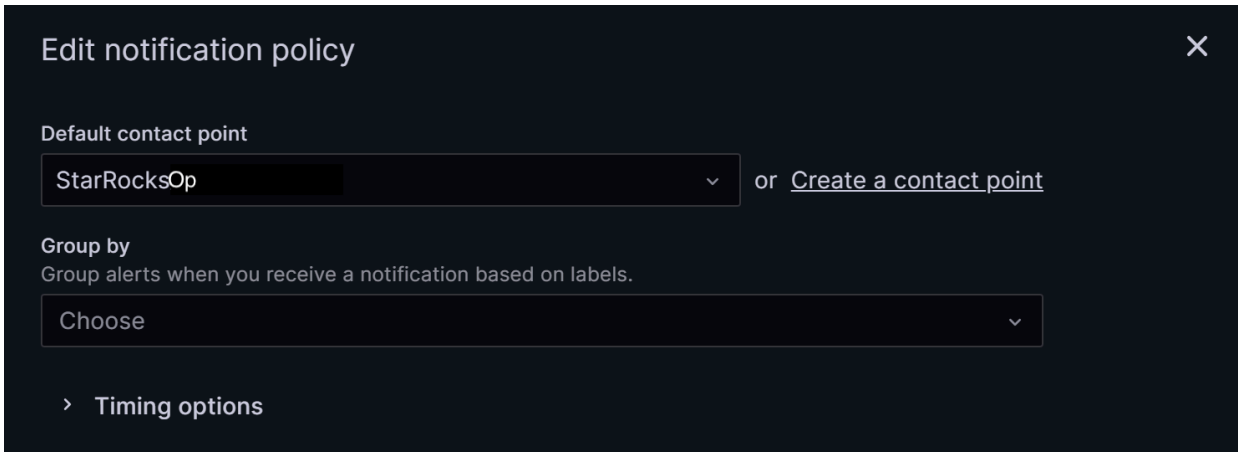




Политики уведомлений используют древовидную структуру, и политика по умолчанию представляет собой корневую политику по умолчанию для уведомления. Когда другие политики не установлены, все правила алертов по умолчанию будут соответствовать этой политике. Затем она будет использовать точку контакта по умолчанию, настроенную в ней, для уведомлений.

i. В поле **Default contact point** выберите точку контакта, которую вы создали ранее, например, "SelenaOp".

ii. **Group by** - это ключевая концепция в Grafana Alerting, группирующая инстансы алертов с похожими характеристиками в одну воронку. Этот tutorial не включает группировку, и вы можете использовать настройку по умолчанию.

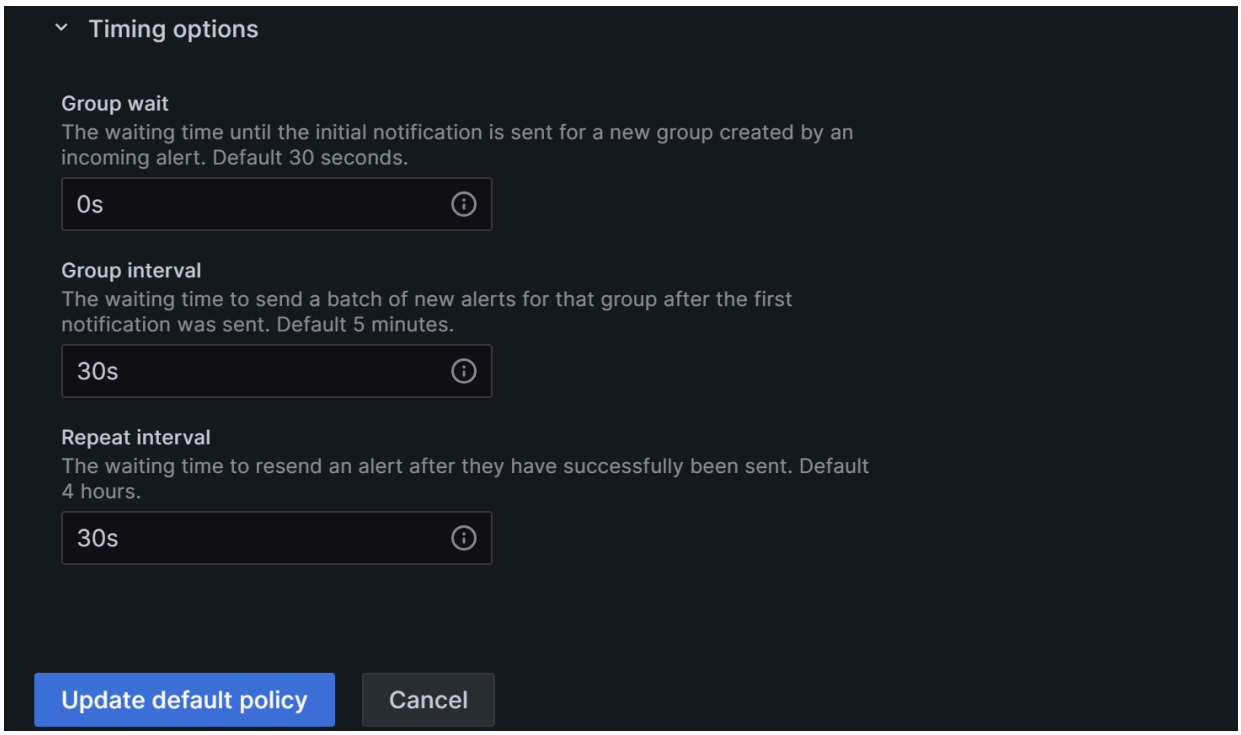


iii. Разверните поле **Timing options** и настройте **Group wait**, **Group interval** и **Repeat interval**.

- **Group wait**: Время ожидания для начальной отправки уведомления после того, как новый алерт создает новую группу. По умолчанию 30 секунд.

- **Group interval**: Интервал, с которым отправляются алерты для существующей группы. По умолчанию 5 минут, что означает, что уведомления не будут отправляться этой группе раньше, чем через 5 минут с момента отправки предыдущего алерта. Это означает, что уведомления не будут отправляться раньше, чем через 5 минут (по умолчанию) с момента последней пакетной доставки обновлений, независимо от того, был ли интервал правила алерта для этих инстансов алертов ниже. По умолчанию 5 минут.
- **Repeat interval**: Время ожидания для повторной отправки алерта после того, как они были успешно отправлены. Интервал, с которым отправляются алерты для существующей группы. По умолчанию 5 минут, что означает, что уведомления не будут отправляться этой группе раньше, чем через 5 минут с момента отправки предыдущего алерта.

Вы можете настроить параметры, как показано ниже, чтобы Grafana отправляла алерт по этим правилам: 0 секунд (Group wait) после **выполнения условий алерта**, Grafana отправит первый email алерта. После этого Grafana будет повторно отправлять алерт каждую 1 минуту (Group interval + Repeat interval).



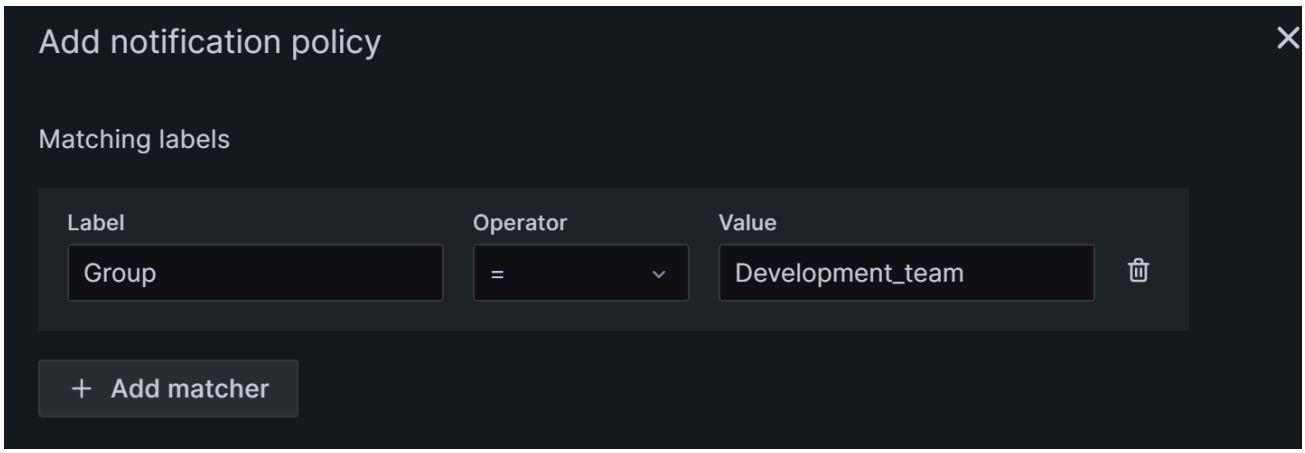
> **ПРИМЕЧАНИЕ**
>
> В предыдущем абзаце используется "выполнение условий алерта", а не "достижение порога алерта", чтобы избежать ложных алертов.

> Рекомендуется установить алерт для запуска через определенную продолжительность времени после достижения порога.

3. После завершения конфигурации нажмите **Update default policy**.

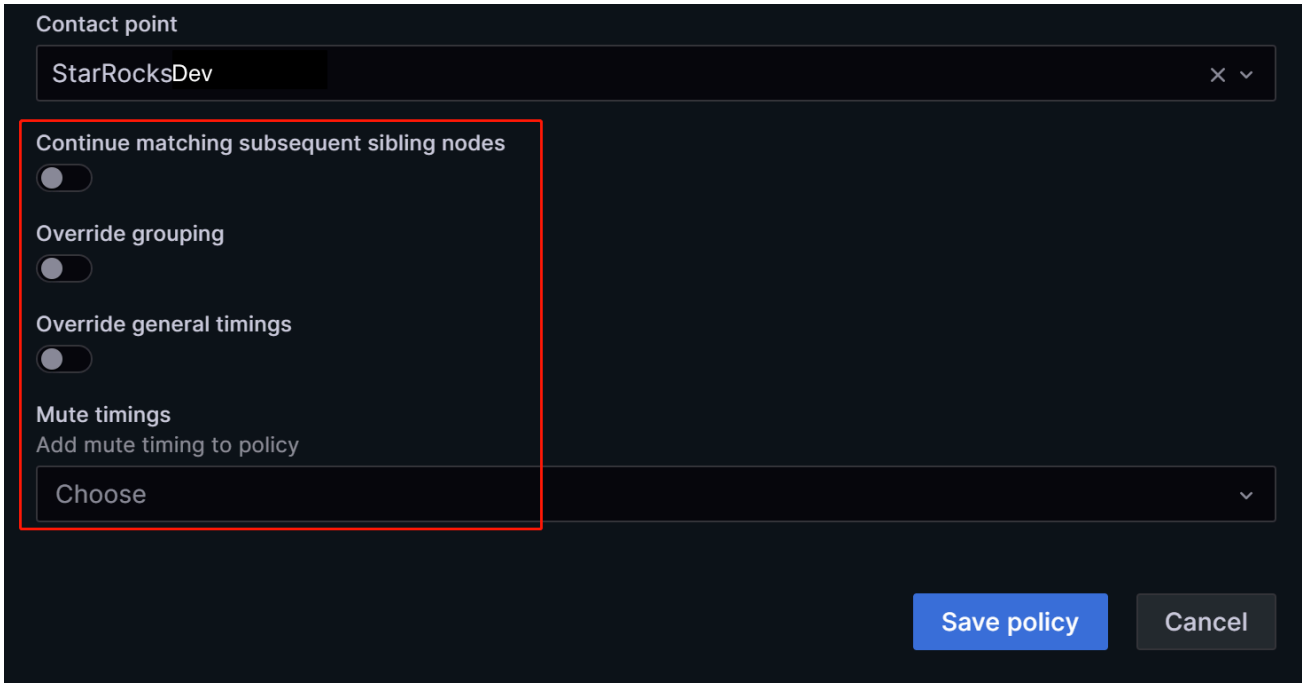
4. Если вам нужно создать вложенную политику, нажмите на **New nested policy** на странице **Notification policies**.

   Вложенные политики используют labels для определения правил соответствия. Labels, определенные во вложенной политике, могут использоваться в качестве условий для соответствия при настройке правил алертов позже. Следующий пример настраивает label как `Group=Development_team`.



   В поле **Contact point** выберите "SelenaDev". Таким образом, при настройке правил алертов с label `Group=Development_team`, "SelenaDev" будет настроен для получения алертов.

   Вы можете сделать так, чтобы вложенная политика наследовала опции времени от родительской политики. После завершения конфигурации нажмите **Save policy**, чтобы сохранить политику.

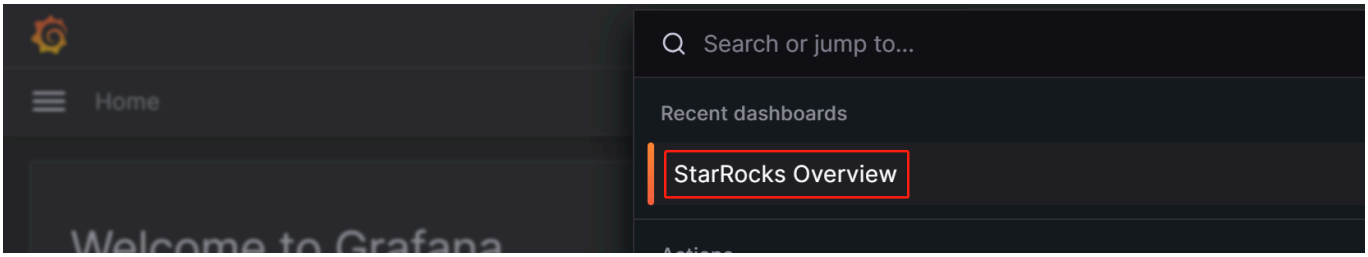


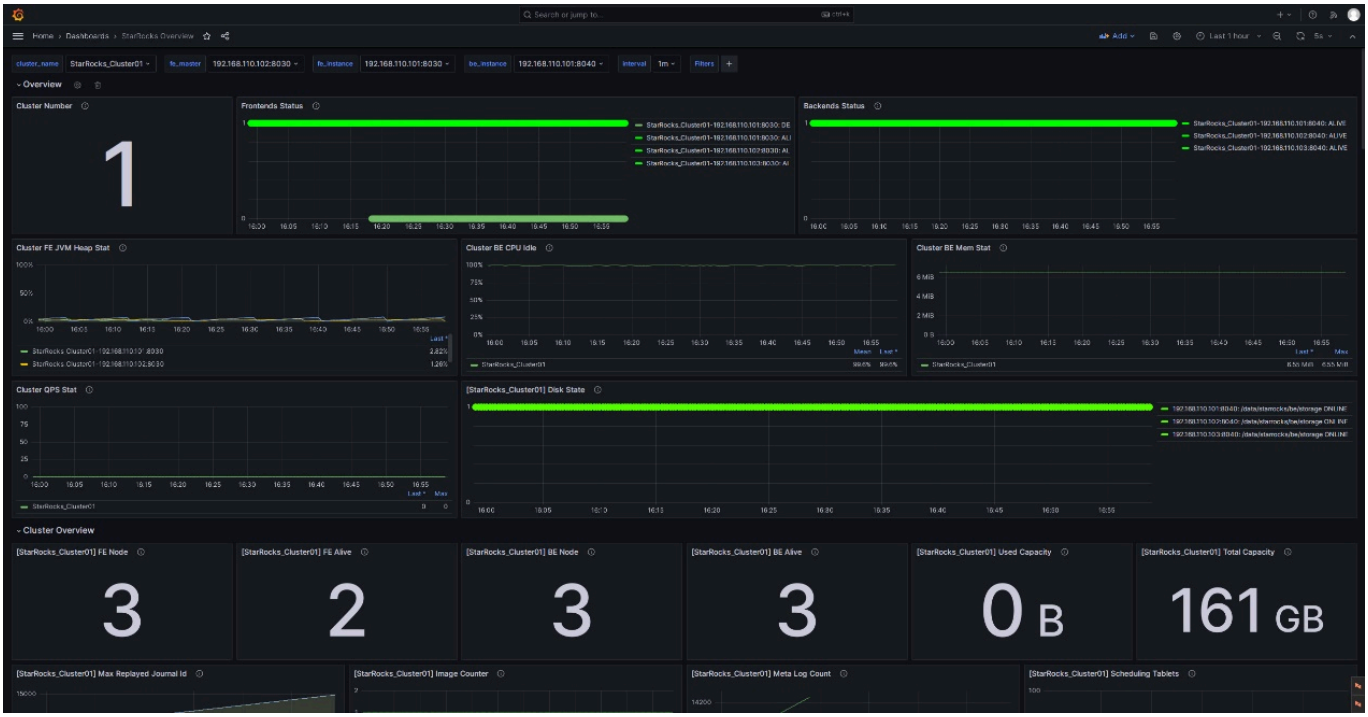
Если вас интересуют детали политик уведомлений или если ваш бизнес имеет более сложные сценарии алертов, вы можете обратиться к документации Grafana для дополнительной информации.

3.4 Определение правил алертов

После настройки политик уведомлений вам также нужно определить правила алертов для Selena.

Войдите в Web UI Grafana, найдите и перейдите к ранее настроенному Selena Overview Dashboard.





3.4.1 Правило алерта для статуса FE и BE

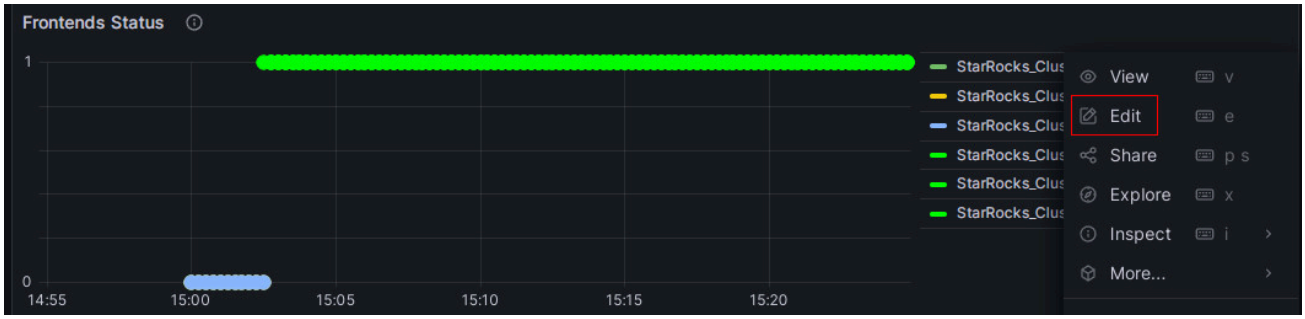
Для cluster Selena статус всех узлов FE и BE должен быть активным. Любой узел со статусом DEAD должен вызывать алерт.

Следующий пример использует метрики Frontends Status и Backends Status в Selena Overview для мониторинга статуса FE и BE. Поскольку вы можете настроить несколько cluster Selena в Prometheus, обратите внимание, что метрики Frontends Status и Backends Status предназначены для всех cluster, которые вы зарегистрировали.

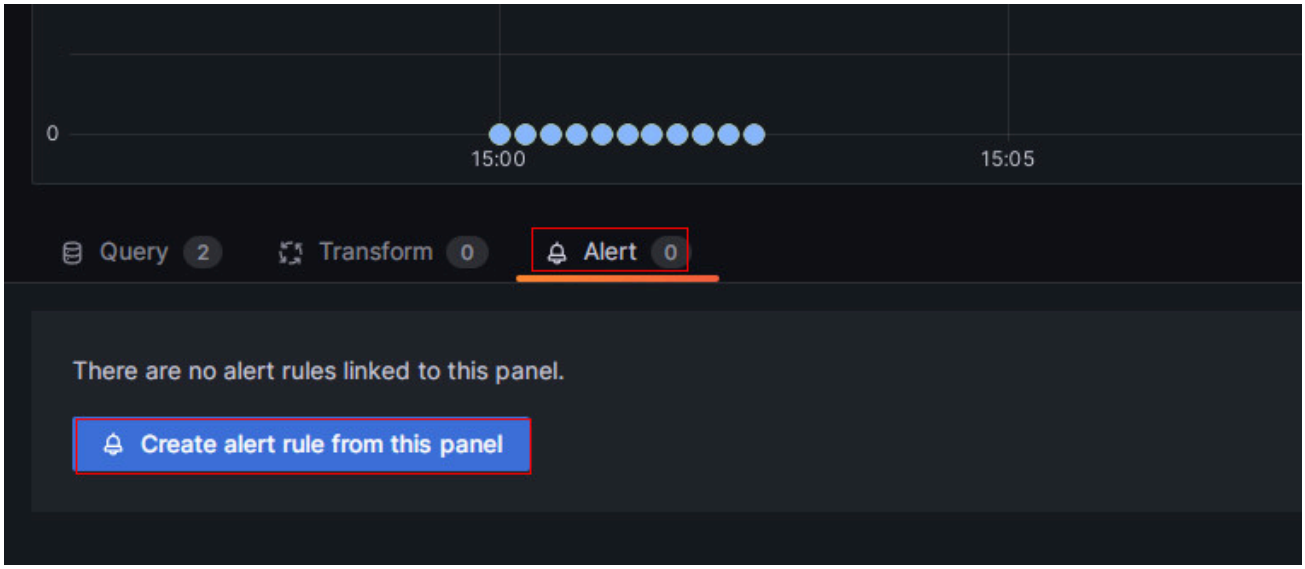
Настройка правила алерта для FE

Выполните следующие процедуры для настройки алертов для **Frontends Status**:

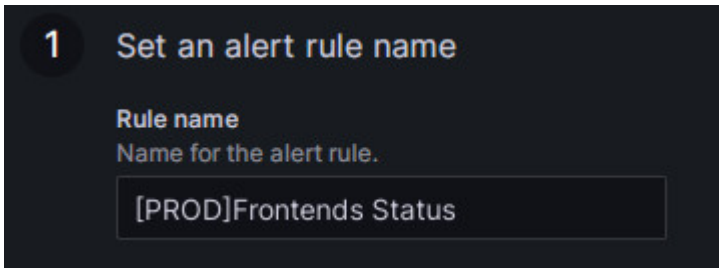
1. Нажмите на значок More (...) справа от элемента мониторинга **Frontends Status** и нажмите **Edit**.



2. На новой странице выберите **Alert**, затем нажмите **Create alert rule** from this panel, чтобы войти на страницу создания правила.



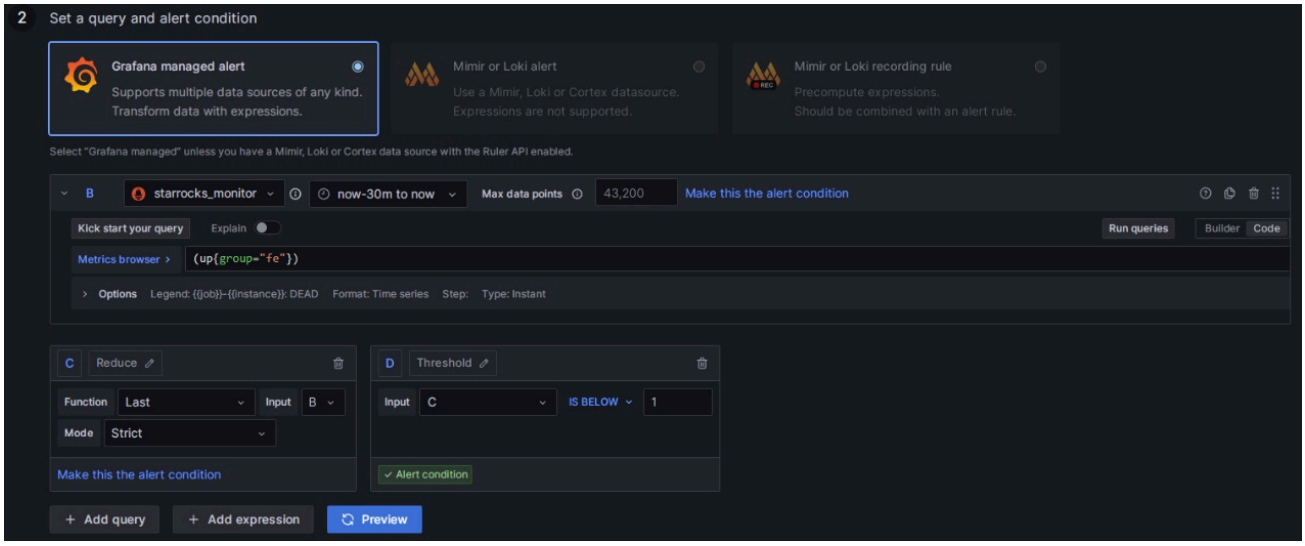
3. Установите имя правила в поле **Rule name**. Значение по умолчанию - заголовок метрики мониторинга. Если у вас несколько cluster, вы можете добавить имя cluster в качестве префикса для различения, например, "[PROD]Frontends Status".



4. Настройте правило алерта следующим образом.

   i. Выберите **Grafana managed alert**.
   ii. Для секции **B** измените правило как `(up{group="fe"})`.
   iii. Нажмите на значок удаления справа от секции **A**, чтобы удалить секцию **A**.
   iv. Для секции **C** измените поле **Input** на **B**.
   v. Для секции **D** измените условие на `IS BELOW 1`.

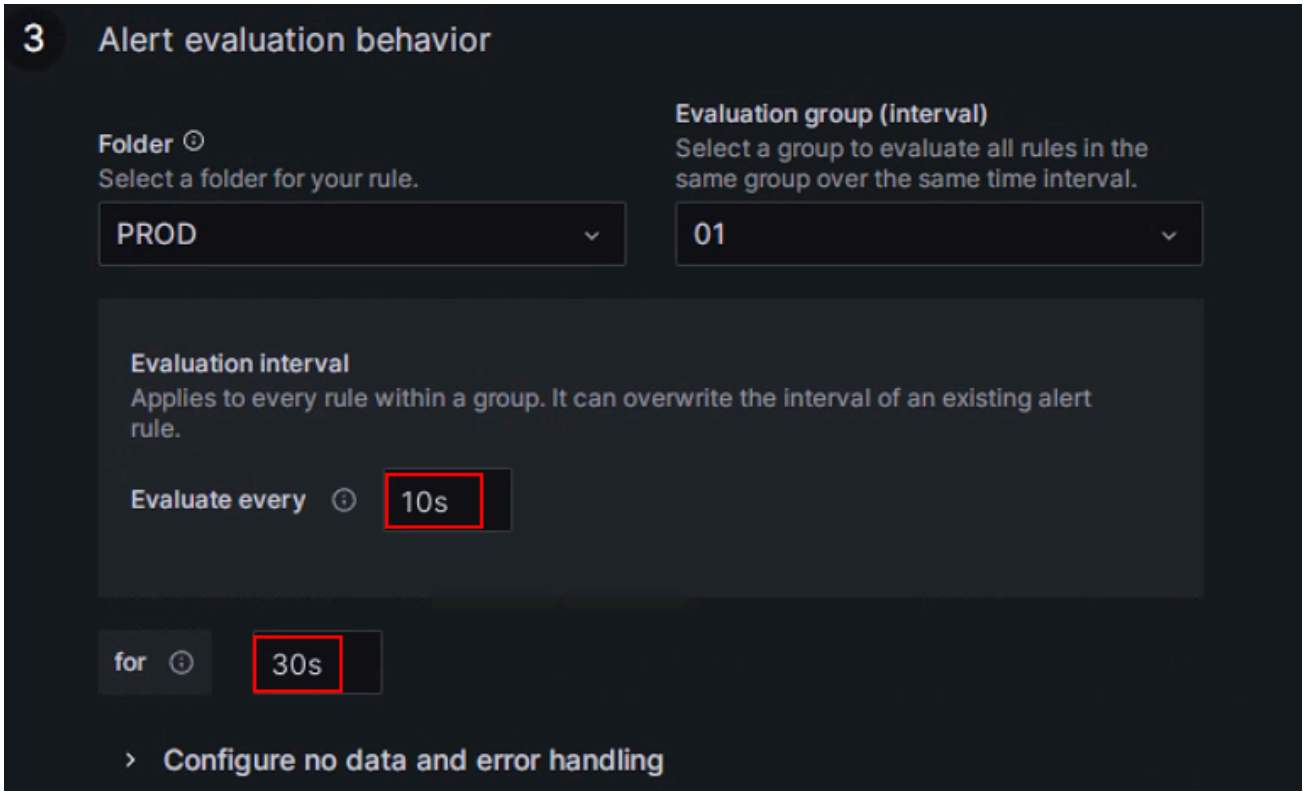
   После завершения этих настроек страница будет выглядеть, как показано ниже:



Нажмите, чтобы просмотреть подробные инструкции

5. Установите правила оценки алертов.

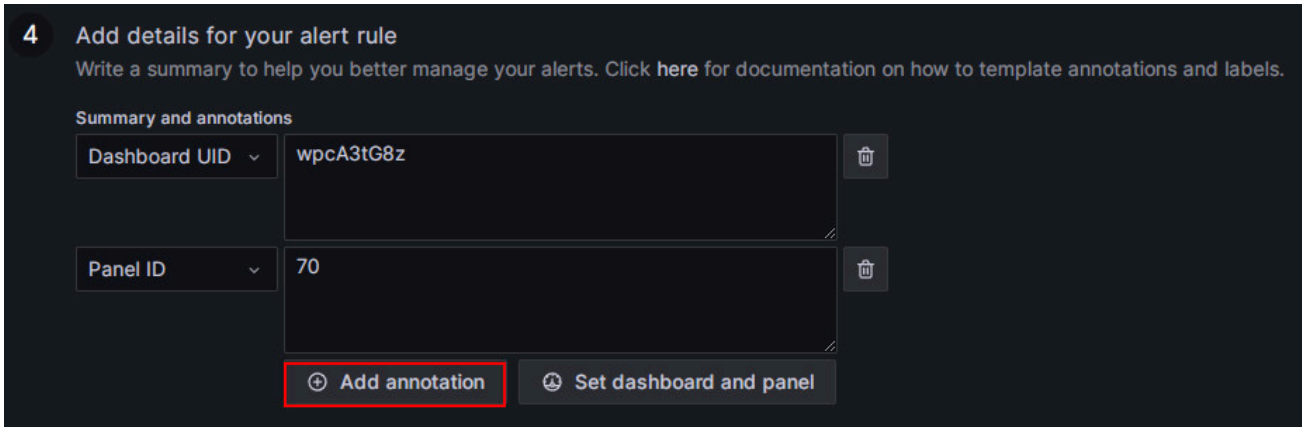
   Согласно документации Grafana, вам нужно настроить частоту оценки правил алертов и частоту, с которой меняется их статус. Простыми словами, это включает настройку "как часто проверять с правилами алертов" и "как долго должно сохраняться аномальное состояние после обнаружения перед запуском алерта (чтобы избежать ложных алертов, вызванных временными всплесками)". Каждая группа Evaluation содержит независимый интервал оценки для определения частоты проверки правил алертов. Вы можете создать новую папку с именем **PROD** специально для production cluster Selena и создать новую группу Evaluation `01` в ней. Затем настройте эту группу для проверки каждые `10` секунд и запустите алерт, если аномалия сохраняется в течение `30` секунд.



> **ПРИМЕЧАНИЕ**
>
> Упомянутая ранее опция "Disable resolved message" в разделе конфигурации канала алертов, которая контролирует время отправки email для восстановления сервиса cluster, также зависит от параметра "Evaluate every" выше. Другими словами, когда Grafana выполняет новую проверку и обнаруживает, что сервис восстановлен, он отправляет email для уведомления контактов.

6. Добавьте аннотации алертов.

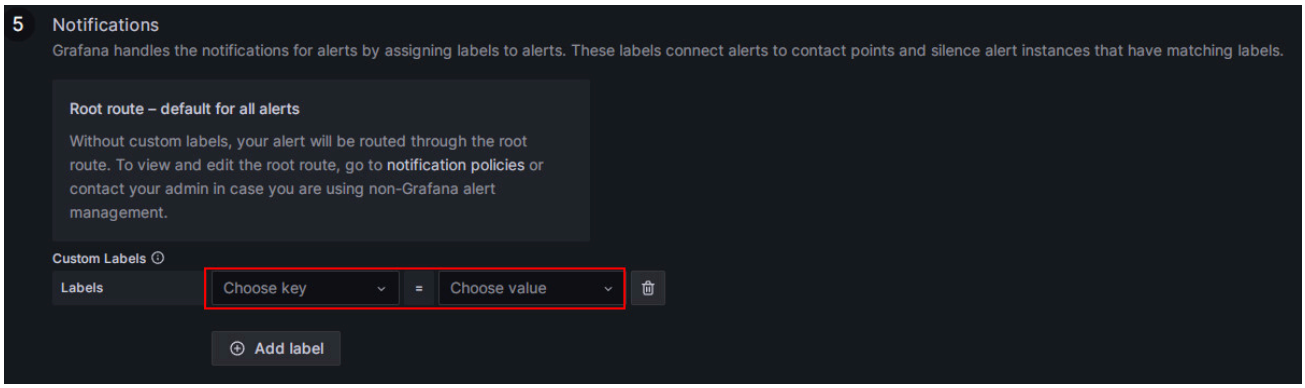
   В секции **Add details for your alert rule** нажмите **Add annotation**, чтобы настроить содержимое email алерта. Обратите внимание, не изменяйте поля **Dashboard UID** и **Panel ID**.



В выпадающем списке **Choose** выберите **Description** и добавьте описательное содержимое для email алерта, например, "FE node in your Selena production cluster failed, please check!"

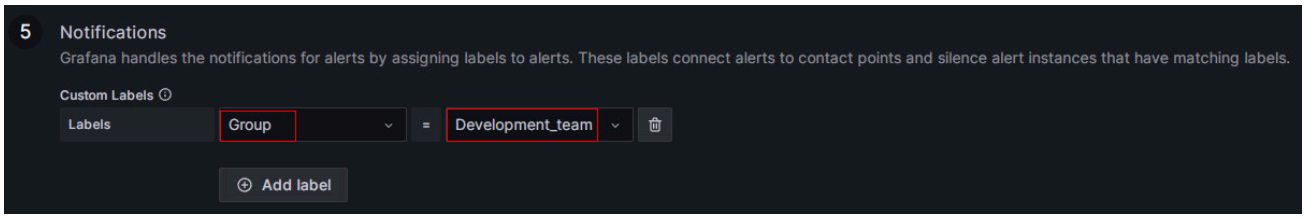
7. Соответствие политикам уведомлений.

Укажите политику уведомлений для правила алерта. По умолчанию все правила алертов соответствуют политике по умолчанию. Когда условие алерта выполнено, Grafana будет использовать точку контакта "SelenaOp" в политике по умолчанию для отправки сообщений алертов в настроенную группу email.



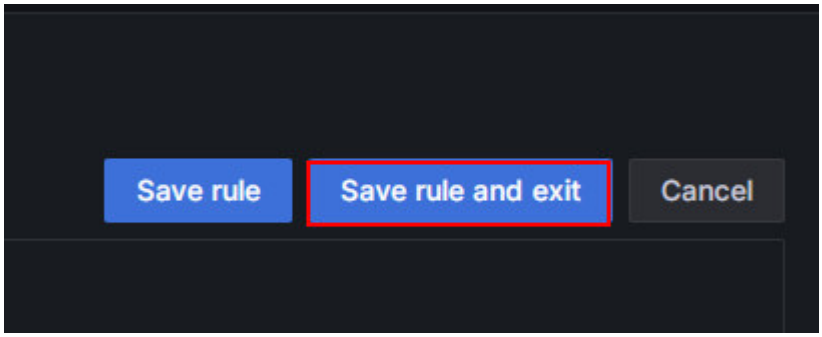
Если вы хотите использовать вложенную политику, установите поле **Label** на соответствующую вложенную политику, например, `Group=Development_team`.

Пример:



Когда условие алерта выполнено, email будут отправлены на "SelenaDev" вместо "SelenaOp" в политике по умолчанию.

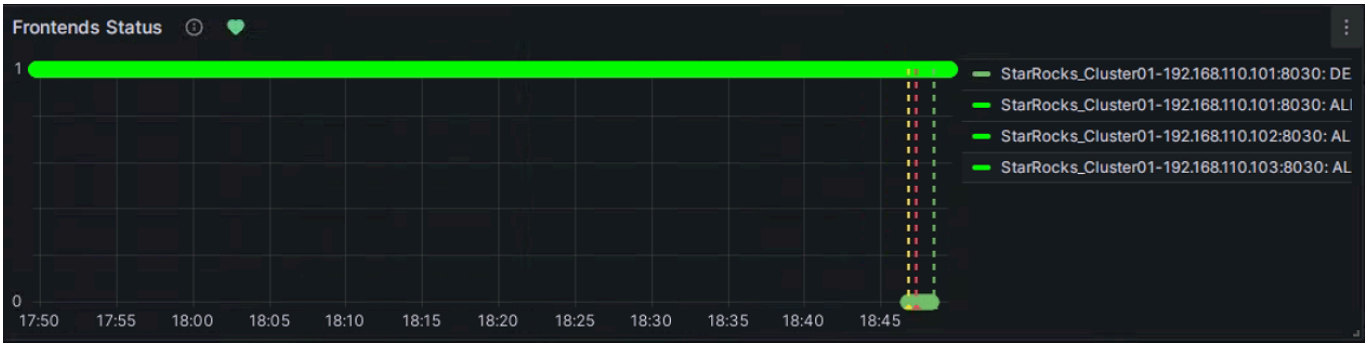
8. После завершения всех конфигураций нажмите **Save rule and exit**.



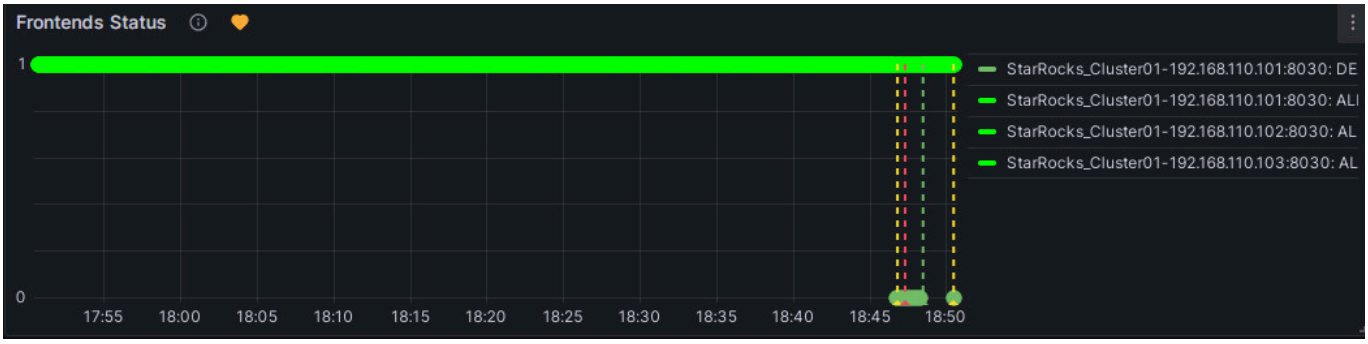
Тестирование запуска алерта

Вы можете вручную остановить узел FE для тестирования алерта. В этот момент символ в форме сердца справа от Frontends Status изменится с зеленого на желтый, а затем на красный.

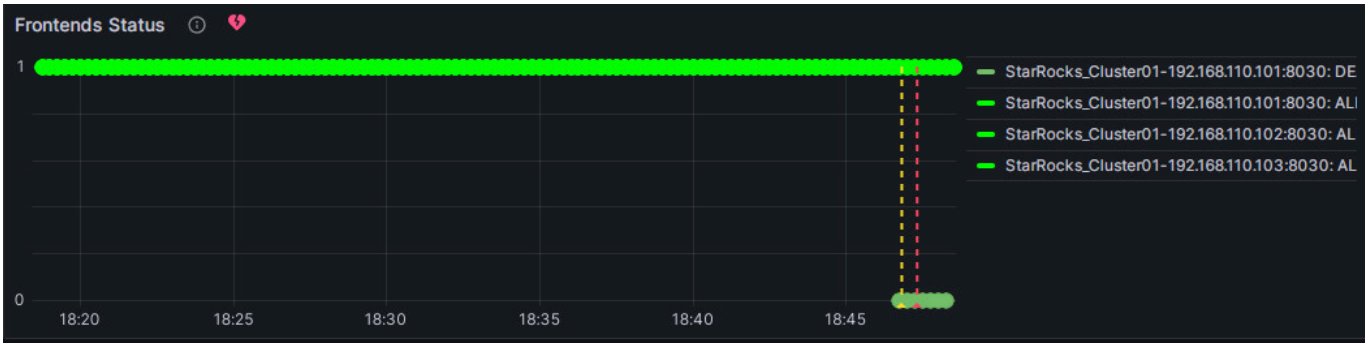
Зеленый: Указывает, что во время последней периодической проверки статус каждого инстанса элемента метрики был нормальным, и алерт не был запущен. Зеленый статус не гарантирует, что текущий узел находится в нормальном статусе. Может быть задержка в изменении статуса после аномалии сервиса узла, но обычно задержка не в порядке минут.



Желтый: Указывает, что во время последней периодической проверки инстанс элемента метрики был обнаружен аномальным, но продолжительность аномального состояния еще не достигла "Duration", настроенного выше. В этот момент Grafana не отправит алерт и продолжит периодические проверки до тех пор, пока продолжительность аномального состояния не достигнет настроенного "Duration". В течение этого периода, если статус восстановлен, символ изменится обратно на зеленый.



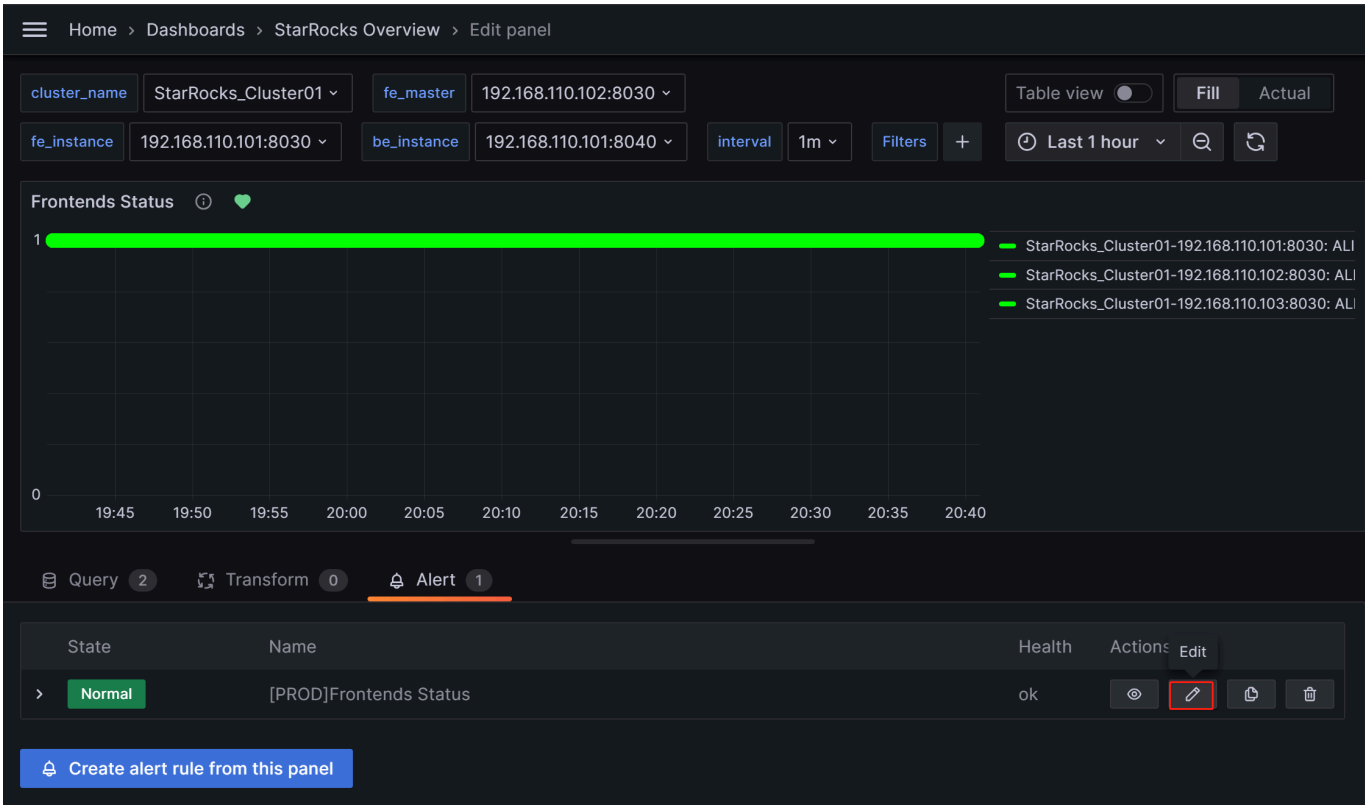
Красный: Когда продолжительность аномального состояния достигает настроенного "Duration", символ становится красным, и Grafana отправит email алерта. Символ останется красным до тех пор, пока аномальное состояние не будет решено, после чего он изменится обратно на зеленый.



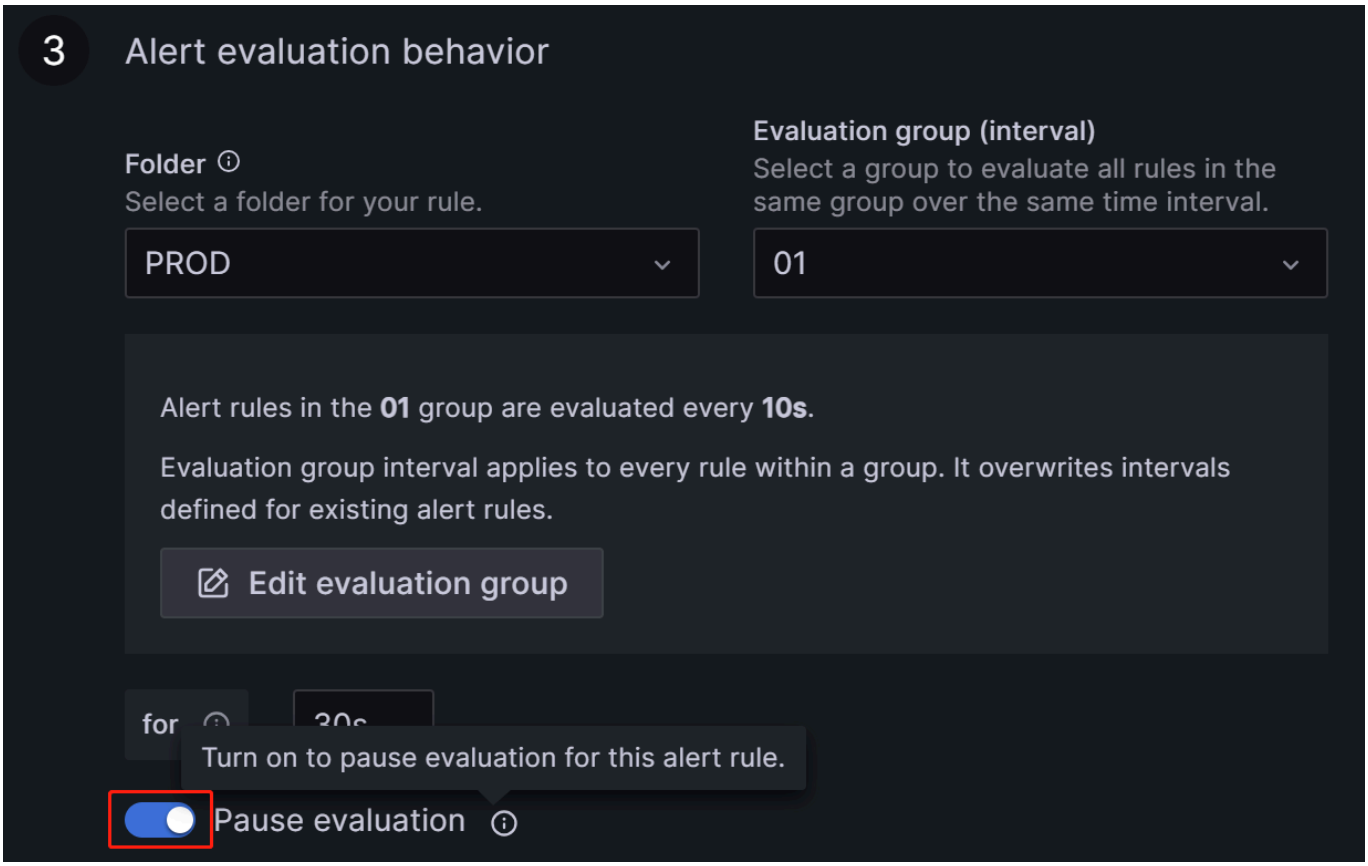
Ручная приостановка алертов

Предположим, что аномалия требует продолжительного периода для решения или алерты постоянно запускаются по причинам, отличным от аномалии. Вы можете временно приостановить оценку правила алерта, чтобы предотвратить постоянную отправку email алертов Grafana.

Перейдите на вкладку Alert, соответствующую элементу метрики на Dashboard, и нажмите значок редактирования:



В секции **Alert Evaluation Behavior** переключите переключатель **Pause Evaluation** в положение ON.



> **ПРИМЕЧАНИЕ**
>
> После приостановки оценки вы получите email, уведомляющий вас о том, что сервис восстановлен.

Настройка правила алерта для BE

Вы можете следовать приведенному выше процессу для настройки правил алертов для BE.

Редактирование конфигурации для элемента метрики Backends Status:

1. В секции **Set an alert rule name** настройте имя как "[PROD]Backends Status".
2. В секции **Set a query and alert condition** установите PromSQL как `(up{group="be"})` и используйте те же настройки, что и в правиле алерта FE для других элементов.
3. В секции **Alert evaluation behavior** выберите каталог **PROD** и группу Evaluation **01**, созданные ранее, и установите duration равной 30 секундам.
4. В секции **Add details for your alert rule** нажмите **Add annotation**, выберите **Description** и введите содержимое алерта, например, "BE node in your Selena production cluster failed, please check! Stack information for BE failure will be printed in the BE log file **be.out**. You can identify the cause based on the logs".
5. В секции **Notifications** настройте **Labels** так же, как правило алерта FE. Если Labels не настроены, Grafana будет использовать политику по умолчанию и отправлять email алертов на канал алертов "SelenaOp".

3.4.2 Правило алерта запросов

Элемент метрики для сбоев запросов - **Query Error** в **Query Statistic**.

Настройте правило алерта для элемента метрики "Query Error" следующим образом:

1. В секции **Set an alert rule name** настройте имя как "[PROD] Query Error".
2. В секции **Set a query and alert condition** удалите секцию **B**. Установите **Input** в секции **A** на **C**. В секции **C** используйте значение по умолчанию для PromQL, которое является `rate(selena_fe_query_err{job="Selena_Cluster01"}[1m])`, представляя количество неудачных запросов в минуту, деленное на 60s. Это включает как неудачные запросы, так и запросы, которые превысили лимит timeout. Затем в секции **D** настройте правило как `A IS ABOVE 0.05`.
3. В секции **Alert evaluation behavior** выберите каталог **PROD** и группу Evaluation **01**, созданные ранее, и установите duration равной 30 секундам.

4. В секции **Add details for your alert rule** нажмите **Add annotation**, выберите **Description** и введите содержимое алерта, например, "High query failure rate, please check the resource usage or configure query timeout reasonably. If queries are failing due to timeouts, you can adjust the query timeout by setting the system variable `query_timeout`".
5. В секции **Notifications** настройте **Labels** так же, как правило алерта FE. Если Labels не настроены, Grafana будет использовать политику по умолчанию и отправлять email алертов на канал алертов "SelenaOp".

3.4.3 Правило алерта сбоя пользовательской операции

Этот элемент мониторит частоту сбоев операции Schema Change, соответствующий элементу метрики **Schema Change** в **BE tasks**. Он должен быть настроен для алерта, когда больше 0.

1. В секции **Set an alert rule name** настройте имя как "[PROD] Schema Change".
2. В секции **Set a query and alert condition** удалите секцию **A**. Установите **Input** в секции **C** на **B**. В секции **B** используйте значение по умолчанию для PromQL, которое является `irate(selena_be_engine_requests_total{job="Selena_Cluster01", type="create_rollup", status="failed"}[1m])`, представляя количество неудачных задач Schema Change в минуту, деленное на 60s. Затем в секции **D** настройте правило как `C IS ABOVE 0`.
3. В секции **Alert evaluation behavior** выберите каталог **PROD** и группу Evaluation **01**, созданные ранее, и установите duration равной 30 секундам.
4. В секции **Add details for your alert rule** нажмите **Add annotation**, выберите **Description** и введите содержимое алерта, например, "Failed Schema Change tasks detected, please check promptly. You can increase the memory limit available for Schema Change by adjusting the BE configuration parameter `memory_limitation_per_thread_for_schema_change`, which is set to 2GB by default".
5. В секции **Notifications** настройте **Labels** так же, как правило алерта FE. Если Labels не настроены, Grafana будет использовать политику по умолчанию и отправлять email алертов на канал алертов "SelenaOp".

3.4.4 Правило алерта сбоя операции Selena

BE Compaction Score

Этот элемент соответствует **BE Compaction Score** в **Cluster Overview** и используется для мониторинга давления compaction на cluster.

1. В секции **Set an alert rule name** настройте имя как "[PROD] BE Compaction Score".
2. В секции **Set a query and alert condition** настройте правило в секции C как `B IS ABOVE 0`. Вы можете использовать значения по умолчанию для других элементов.
3. В секции **Alert evaluation behavior** выберите каталог **PROD** и группу Evaluation **01**, созданные ранее, и установите duration равной 30 секундам.
4. В секции **Add details for your alert rule** нажмите **Add annotation**, выберите **Description** и введите содержимое алерта, например, "High compaction pressure. Please check whether there are high-frequency or high concurrency loading tasks and reduce the loading frequency. If the cluster has sufficient CPU, memory, and I/O resources, consider adjusting the cluster compaction strategy".
5. В секции **Notifications** настройте **Labels** так же, как правило алерта FE. Если Labels не настроены, Grafana будет использовать политику по умолчанию и отправлять email алертов на канал алертов "SelenaOp".

Clone

Этот элемент соответствует **Clone** в **BE tasks** и в основном используется для мониторинга балансировки replica или операций восстановления replica внутри Selena, которые обычно не должны завершаться сбоем.

1. В секции **Set an alert rule name** настройте имя как "[PROD] Clone".
2. В секции **Set a query and alert condition** удалите секцию A. Установите **Input** в секции **C** на **B**. В секции **B** используйте значение по умолчанию для PromQL, которое является `irate(selena_be_engine_requests_total{job="Selena_Cluster01", type="clone", status="failed"}[1m])`, представляя количество неудачных задач Clone в минуту, деленное на 60s. Затем в секции **D** настройте правило как `C IS ABOVE 0`.
3. В секции **Alert evaluation behavior** выберите каталог **PROD** и группу Evaluation **01**, созданные ранее, и установите duration равной 30 секундам.
4. В секции **Add details for your alert rule** нажмите **Add annotation**, выберите **Description** и введите содержимое алерта, например, "Detected a failure in the clone task. Please check the cluster BE status, disk status, and network status".
5. В секции **Notifications** настройте **Labels** так же, как правило алерта FE. Если Labels не настроены, Grafana будет использовать политику по умолчанию и отправлять email алертов на канал алертов "SelenaOp".

3.4.5 Правило алерта доступности сервиса

Этот элемент мониторит количество логов метаданных в BDB, соответствующий элементу мониторинга **Meta Log Count** в **Cluster Overview**.

1. В секции **Set an alert rule name** настройте имя как "[PROD] Meta Log Count".
2. В секции **Set a query and alert condition** настройте правило в секции **C** как `B IS ABOVE 100000`. Вы можете использовать значения по умолчанию для других элементов.
3. В секции **Alert evaluation behavior** выберите каталог **PROD** и группу Evaluation **01**, созданные ранее, и установите duration равной 30 секундам.
4. В секции **Add details for your alert rule** нажмите **Add annotation**, выберите **Description** и введите содержимое алерта, например, "Detected that the metadata count in FE BDB is significantly higher than the expected value, which can indicate a failed Checkpoint operation. Please check whether the Xmx heap memory configuration in the FE configuration file **fe.conf** is reasonable".
5. В секции **Notifications** настройте **Labels** так же, как правило алерта FE. Если Labels не настроены, Grafana будет использовать политику по умолчанию и отправлять email алертов на канал алертов "SelenaOp".

3.4.6 Правило алерта перегрузки системы

BE CPU Idle

Этот элемент мониторит процент простоя CPU на узлах BE.

1. В секции **Set an alert rule name** настройте имя как "[PROD] BE CPU Idle".
2. В секции **Set a query and alert condition** настройте правило в секции C как `B IS BELOW 10`. Вы можете использовать значения по умолчанию для других элементов.
3. В секции **Alert evaluation behavior** выберите каталог **PROD** и группу Evaluation **01**, созданные ранее, и установите duration равной 30 секундам.
4. В секции **Add details for your alert rule** нажмите **Add annotation**, выберите **Description** и введите содержимое алерта, например, "Detected that BE CPU load is consistently high. It will impact other tasks in the cluster. Please check whether the cluster is abnormal or if there is a CPU resource bottleneck".
5. В секции **Notifications** настройте **Labels** так же, как правило алерта FE. Если Labels не настроены, Grafana будет использовать политику по умолчанию и отправлять email алертов на канал алертов "SelenaOp".

BE Memory

Этот элемент соответствует **BE Mem** в **BE**, мониторящий использование памяти на узлах BE.

1. В секции **Set an alert rule name** настройте имя как "[PROD] BE Mem".
2. В секции **Set a query and alert condition** настройте PromSQL как `selena_be_process_mem_bytes{job="Selena_Cluster01"}/(<be_mem_limit>*1024*1024*1024)`, где `<be_mem_limit>` необходимо заменить текущим лимитом доступной памяти узла BE, то есть размером памяти сервера, умноженным на значение элемента конфигурации BE `mem_limit`. Пример: `selena_be_process_mem_bytes{job="Selena_Cluster01"}/(49*1024*1024*1024)`. Затем в секции **C** настройте правило как `B IS ABOVE 0.9`.
3. В секции **Alert evaluation behavior** выберите каталог **PROD** и группу Evaluation **01**, созданные ранее, и установите duration равной 30 секундам.
4. В секции **Add details for your alert rule** нажмите **Add annotation**, выберите **Description** и введите содержимое алерта, например, "Detected that BE memory usage is consistently high. To prevent query failure, please consider expanding memory size or adding BE nodes".
5. В секции **Notifications** настройте **Labels** так же, как правило алерта FE. Если Labels не настроены, Grafana будет использовать политику по умолчанию и отправлять email алертов на канал алертов "SelenaOp".

Disks Avail Capacity

Этот элемент соответствует **Disk Usage** в **BE**, мониторящий соотношение оставшегося пространства в каталоге, где находится путь хранения BE.

1. В секции **Set an alert rule name** настройте имя как "[PROD] Disks Avail Capacity".
2. В секции **Set a query and alert condition** настройте правило в секции **C** как `B`` IS BELOW 0.2`. Вы можете использовать значения по умолчанию для других элементов.
3. В секции **Alert evaluation behavior** выберите каталог **PROD** и группу Evaluation **01**, созданные ранее, и установите duration равной 30 секундам.
4. В секции **Add details for your alert rule** нажмите **Add annotation**, выберите **Description** и введите содержимое алерта, например, "Detected that BE disk available space is below 20%, please release disk space or expand the disk".
5. В секции **Notifications** настройте **Labels** так же, как правило алерта FE. Если Labels не настроены, Grafana будет использовать политику по умолчанию и отправлять email алертов на канал алертов "SelenaOp".

FE JVM Heap Stat

Этот элемент соответствует **Cluster FE JVM Heap Stat** в **Overview**, мониторящий пропорцию использования памяти JVM FE к лимиту heap-памяти FE.

1. В секции **Set an alert rule name** настройте имя как "[PROD] FE JVM Heap Stat".
2. В секции **Set a query and alert condition** настройте правило в секции **C** как `B IS ABOVE 80`. Вы можете использовать значения по умолчанию для других элементов.
3. В секции **Alert evaluation behavior** выберите каталог **PROD** и группу Evaluation **01**, созданные ранее, и установите duration равной 30 секундам.
4. В секции **Add details for your alert rule** нажмите **Add annotation**, выберите **Description** и введите содержимое алерта, например, "Detected that FE heap memory usage is high, please adjust the heap memory limit in the FE configuration file **fe.conf**".
5. В секции **Notifications** настройте **Labels** так же, как правило алерта FE. Если Labels не настроены, Grafana будет использовать политику по умолчанию и отправлять email алертов на канал алертов "SelenaOp".

Приложение

Включение обнаружения сервисов для Prometheus

Вы можете включить обнаружение сервисов для Prometheus, чтобы он мог автоматически обнаруживать сервисы (узлы) после масштабирования cluster.

> **ПРИМЕЧАНИЕ**
>
> Следующий раздел использует AWS в качестве примера.

1. Предоставьте инстансу EC2, который размещает ваш сервис Prometheus, следующие разрешения, используя IAM Policy:

```
{
	"Version": "2012-10-17",
	"Statement": [
		{
			"Effect": "Allow",
			"Action": [
				"ec2:DescribeInstances",
				"ec2:DescribeTags"
			],
			"Resource": "*"
		}
	]
}
```

Для подробных инструкций по аутентификации к ресурсам AWS см. Authenticate to AWS resources.

С этими разрешениями Prometheus может перечислять инстансы и их tags в регионе.

2. Добавьте секции `ec2_sd_configs` и `relabel_configs` в **prometheus/prometheus.yml**.

   Пример:

```yaml
global:
scrape_interval: 15s # Установите глобальный интервал scrape равным 15s. По умолчанию 1 мин.
evaluation_interval: 15s # Установите глобальный интервал evaluation правил равным 15s. По умолчанию 1 мин.
scrape_configs:
- job_name: 'Selena_Cluster01'
   metrics_path: '/metrics'
   ec2_sd_configs:
      - region: us-west-2
      port: 8030
      filters:
         - name: tag:ClusterName
            values: ['test-stage-20251021']
         - name: tag:ProcessType
            values: ['FE']
      - region: us-west-2
      port: 8040
      filters:
         - name: tag:ClusterName
            values: ['test-stage-20251021']
         - name: tag:ProcessType
            values: ['BE']
   relabel_configs:
      - source_labels: [__meta_ec2_tag_ClusterName]
      regex: test-stage-20251021
      target_label: cluster
      replacement: test-stage-20251021
      - source_labels: [__meta_ec2_tag_ProcessType]
      regex: FE
      target_label: group
      replacement: fe
      - source_labels: [__meta_ec2_tag_ProcessType]
      regex: BE
      target_label: group
      replacement: be
```

Q&A

Q: Почему Dashboard не может обнаружить аномалии?

A: Grafana Dashboard полагается на системное время сервера, на котором он размещен, для получения значений для элементов мониторинга. Если страница Grafana Dashboard

остается неизменной после аномалии cluster, вы можете проверить, синхронизированы ли системные часы серверов, а затем выполнить калибровку времени cluster.

Q: Как я могу реализовать градацию алертов?

A: Возьмем элемент Query Error в качестве примера, вы можете создать два правила алертов для него с разными порогами алертов. Например:

- **Уровень риска**: Установите частоту сбоев больше 0.05, указывающую на риск. Отправьте алерт команде разработки.
- **Уровень серьезности**: Установите частоту сбоев больше 0.20, указывающую на серьезность. В этот момент уведомление об алерте будет отправлено как команде разработки, так и команде operations одновременно.

Q: Как я могу получить более подробные метрики, включая метрики на уровне таблицы, метрики materialized view и статистику соединений с labels пользователей?

A: По умолчанию endpoint `/metrics` собирает метрики в минимизированном режиме для минимизации влияния на производительность. Чтобы получить подробные метрики, вам необходимо добавить определенные параметры к запросу и предоставить учетные данные Basic Authentication для пользователя с привилегиями ADMIN.

Поддерживаемые параметры:

- `with_table_metrics=all`: Собирает все метрики на уровне таблицы.
- `with_materialized_view_metrics=all`: Собирает все метрики materialized view.
- `with_user_connections=all`: Собирает статистику соединений, категоризированную по labels пользователей.

Требование аутентификации:

Эти параметры вступают в силу только тогда, когда запрос включает действительные учетные данные Basic Authentication для пользователя ADMIN. Если запрос анонимный или у пользователя отсутствуют привилегии ADMIN, эти параметры игнорируются, и возвращаются только метрики по умолчанию.

Пример команды Curl:

```
curl -u <admin_username>:<admin_password> \
"http://<fe_host>:<fe_http_port>/metrics?
with_table_metrics=all&with_materialized_view_metrics=all&with_user_connectio
```

Пример конфигурации Prometheus:

Чтобы включить сбор подробных метрик в Prometheus, настройте `params` и `basic_auth` в вашем `prometheus.yml`:

```
scrape_configs:
  - job_name: 'Selena_Detailed_Metrics'
    metrics_path: '/metrics'
    params:
      with_table_metrics: ['all']
      with_materialized_view_metrics: ['all']
      with_user_connections: ['all']
    basic_auth:
      username: '<admin_username>'
      password: '<admin_password>'
    static_configs:
      - targets: ['<fe_host>:<fe_http_port>']
```

> **ПРИМЕЧАНИЕ**
>
> Сбор всех метрик таблиц и materialized view может увеличить нагрузку на узел FE. Используйте эти параметры с осторожностью в крупномасштабных окружениях.

Общие метрики мониторинга

В этом разделе представлены некоторые важные общие метрики Selena.

Для специализированных метрик для materialized views и shared-data clusters, пожалуйста, обратитесь к соответствующим разделам:

- Метрики для асинхронных materialized view
- Метрики для Shared-data Dashboard и Starlet Dashboard

Для дополнительной информации о том, как построить сервис мониторинга для вашего cluster Selena, см. Мониторинг и алертинг.

Элементы метрик

be_broker_count

- Unit: Count
- Type: Average
- Description: Количество brokers.

be_brpc_endpoint_count

- Unit: Count
- Type: Average
- Description: Количество StubCache в bRPC.

be_bytes_read_per_second

- Unit: Bytes/s
- Type: Average
- Description: Скорость чтения BE.

be_bytes_written_per_second

- Unit: Bytes/s
- Type: Average
- Description: Скорость записи BE.

be_base_compaction_bytes_per_second

- Unit: Bytes/s
- Type: Average
- Description: Скорость base compaction BE.

be_cumulative_compaction_bytes_per_second

- Unit: Bytes/s
- Type: Average
- Description: Скорость cumulative compaction BE.

be_base_compaction_rowsets_per_second

- Unit: Count
- Type: Average
- Description: Скорость base compaction rowsets BE.

be_cumulative_compaction_rowsets_per_second

- Unit: Count
- Type: Average
- Description: Скорость cumulative compaction rowsets BE.

be_base_compaction_failed

- Unit: Count/s
- Type: Average
- Description: Сбой base compaction BE.

be_clone_failed

- Unit: Count/s
- Type: Average
- Description: Сбой клонирования BE.

be_create_rollup_failed

- Unit: Count/s
- Type: Average
- Description: Сбой создания materialized view BE.

be_create_tablet_failed

- Unit: Count/s
- Type: Average
- Description: Сбой создания tablet BE.

be_cumulative_compaction_failed

- Unit: Count/s
- Type: Average
- Description: Сбой cumulative compaction BE.

be_delete_failed

- Unit: Count/s
- Type: Average
- Description: Сбой удаления BE.

be_finish_task_failed

- Unit: Count/s

- Type: Average
- Description: Сбой задачи BE.

be_publish_failed

- Unit: Count/s
- Type: Average
- Description: Сбой публикации версии BE.

be_report_tables_failed

- Unit: Count/s
- Type: Average
- Description: Сбой отчета таблицы BE.

be_report_disk_failed

- Unit: Count/s
- Type: Average
- Description: Сбой отчета диска BE.

be_report_tablet_failed

- Unit: Count/s
- Type: Average
- Description: Сбой отчета tablet BE.

be_report_task_failed

- Unit: Count/s
- Type: Average
- Description: Сбой отчета задачи BE.

be_schema_change_failed

- Unit: Count/s
- Type: Average
- Description: Сбой schema change BE.

be_base_compaction_requests

- Unit: Count/s
- Type: Average
- Description: Запрос base compaction BE.

be_clone_total_requests

- Unit: Count/s
- Type: Average
- Description: Запрос клонирования BE.

be_create_rollup_requests

- Unit: Count/s
- Type: Average
- Description: Запрос создания materialized view BE.

be_create_tablet_requests

- Unit: Count/s
- Type: Average
- Description: Запрос создания tablet BE.

be_cumulative_compaction_requests

- Unit: Count/s
- Type: Average
- Description: Запрос cumulative compaction BE.

be_delete_requests

- Unit: Count/s
- Type: Average
- Description: Запрос удаления BE.

be_finish_task_requests

- Unit: Count/s
- Type: Average
- Description: Запрос завершения задачи BE.

be_publish_requests

- Unit: Count/s
- Type: Average
- Description: Запрос публикации версии BE.

be_report_tablets_requests

- Unit: Count/s

- Type: Average
- Description: Запрос отчета tablet BE.

be_report_disk_requests

- Unit: Count/s
- Type: Average
- Description: Запрос отчета диска BE.

be_report_tablet_requests

- Unit: Count/s
- Type: Average
- Description: Запрос отчета tablet BE.

be_report_task_requests

- Unit: Count/s
- Type: Average
- Description: Запрос отчета задачи BE.

be_schema_change_requests

- Unit: Count/s
- Type: Average
- Description: Запрос отчета schema change BE.

be_storage_migrate_requests

- Unit: Count/s
- Type: Average
- Description: Запрос миграции BE.

be_fragment_endpoint_count

- Unit: Count
- Type: Average
- Description: Количество BE DataStream.

be_fragment_request_latency_avg

- Unit: ms
- Type: Average
- Description: Задержка запросов fragment.

be_fragment_requests_per_second

- Unit: Count/s
- Type: Average
- Description: Количество запросов fragment.

be_http_request_latency_avg

- Unit: ms
- Type: Average
- Description: Задержка HTTP-запросов.

be_http_requests_per_second

- Unit: Count/s
- Type: Average
- Description: Количество HTTP-запросов.

be_http_request_send_bytes_per_second

- Unit: Bytes/s
- Type: Average
- Description: Количество байтов, отправленных для HTTP-запросов.

fe_connections_per_second

- Unit: Count/s
- Type: Average
- Description: Частота новых соединений FE.

fe_connection_total

- Unit: Count
- Type: Cumulative
- Description: Общее количество соединений FE.

fe_edit_log_read

- Unit: Count/s
- Type: Average
- Description: Скорость чтения edit log FE.

fe_edit_log_size_bytes

- Unit: Bytes/s

- Type: Average
- Description: Размер edit log FE.

fe_edit_log_write

- Unit: Bytes/s
- Type: Average
- Description: Скорость записи edit log FE.

fe_checkpoint_push_per_second

- Unit: Count/s
- Type: Average
- Description: Количество checkpoints FE.

fe_pending_hadoop_load_job

- Unit: Count
- Type: Average
- Description: Количество ожидающих hadoop jobs.

fe_committed_hadoop_load_job

- Unit: Count
- Type: Average
- Description: Количество зафиксированных hadoop jobs.

fe_loading_hadoop_load_job

- Unit: Count
- Type: Average
- Description: Количество загружаемых hadoop jobs.

fe_finished_hadoop_load_job

- Unit: Count
- Type: Average
- Description: Количество завершенных hadoop jobs.

fe_cancelled_hadoop_load_job

- Unit: Count
- Type: Average
- Description: Количество отмененных hadoop jobs.

fe_pending_insert_load_job

- Unit: Count
- Type: Average
- Description: Количество ожидающих insert jobs.

fe_loading_insert_load_job

- Unit: Count
- Type: Average
- Description: Количество загружаемых insert jobs.

fe_committed_insert_load_job

- Unit: Count
- Type: Average
- Description: Количество зафиксированных insert jobs.

fe_finished_insert_load_job

- Unit: Count
- Type: Average
- Description: Количество завершенных insert jobs.

fe_cancelled_insert_load_job

- Unit: Count
- Type: Average
- Description: Количество отмененных insert jobs.

fe_pending_broker_load_job

- Unit: Count
- Type: Average
- Description: Количество ожидающих broker jobs.

fe_loading_broker_load_job

- Unit: Count
- Type: Average
- Description: Количество загружаемых broker jobs.

fe_committed_broker_load_job

- Unit: Count

- Type: Average
- Description: Количество зафиксированных broker jobs.

fe_finished_broker_load_job

- Unit: Count
- Type: Average
- Description: Количество завершенных broker jobs.

fe_cancelled_broker_load_job

- Unit: Count
- Type: Average
- Description: Количество отмененных broker jobs.

fe_pending_delete_load_job

- Unit: Count
- Type: Average
- Description: Количество ожидающих delete jobs.

fe_loading_delete_load_job

- Unit: Count
- Type: Average
- Description: Количество загружаемых delete jobs.

fe_committed_delete_load_job

- Unit: Count
- Type: Average
- Description: Количество зафиксированных delete jobs.

fe_finished_delete_load_job

- Unit: Count
- Type: Average
- Description: Количество завершенных delete jobs.

fe_cancelled_delete_load_job

- Unit: Count
- Type: Average
- Description: Количество отмененных delete jobs.

fe_rollup_running_alter_job

- Unit: Count
- Type: Average
- Description: Количество jobs, созданных в rollup.

fe_schema_change_running_job

- Unit: Count
- Type: Average
- Description: Количество jobs в schema change.

cpu_util

- Unit: -
- Type: Average
- Description: Процент использования CPU.

cpu_system

- Unit: -
- Type: Average
- Description: Процент использования cpu_system.

cpu_user

- Unit: -
- Type: Average
- Description: Процент использования cpu_user.

cpu_idle

- Unit: -
- Type: Average
- Description: Процент использования cpu_idle.

cpu_guest

- Unit: -
- Type: Average
- Description: Процент использования cpu_guest.

cpu_iowait

- Unit: -

- Type: Average
- Description: Процент использования cpu_iowait.

cpu_irq

- Unit: -
- Type: Average
- Description: Процент использования cpu_irq.

cpu_nice

- Unit: -
- Type: Average
- Description: Процент использования cpu_nice.

cpu_softirq

- Unit: -
- Type: Average
- Description: Процент использования cpu_softirq.

cpu_steal

- Unit: -
- Type: Average
- Description: Процент использования cpu_steal.

disk_free

- Unit: Bytes
- Type: Average
- Description: Свободная емкость диска.

disk_io_svctm

- Unit: ms
- Type: Average
- Description: Время обслуживания I/O диска.

disk_io_util

- Unit: -
- Type: Average
- Description: Использование диска.

disk_used

- Unit: Bytes
- Type: Average
- Description: Использованная емкость диска.

encryption_keys_created

- Unit: Count
- Type: Cumulative
- Description: количество ключей шифрования файлов, созданных для шифрования файлов

encryption_keys_unwrapped

- Unit: Count
- Type: Cumulative
- Description: количество развернутых мета шифрования для дешифрования файлов

encryption_keys_in_cache

- Unit: Count
- Type: Instantaneous
- Description: количество ключей шифрования, в настоящее время находящихся в кеше ключей

selena_fe_meta_log_count

- Unit: Count
- Type: Instantaneous
- Description: Количество Edit Logs без checkpoint. Значение в пределах `100000` считается разумным.

selena_fe_query_resource_group

- Unit: Count
- Type: Cumulative
- Description: Количество запросов для каждой resource group.

selena_fe_query_resource_group_latency

- Unit: Seconds
- Type: Average
- Description: процентиль задержки запросов для каждой resource group.

selena_fe_query_resource_group_err

- Unit: Count
- Type: Cumulative
- Description: Количество неправильных запросов для каждой resource group.

selena_be_resource_group_cpu_limit_ratio

- Unit: -
- Type: Instantaneous
- Description: Мгновенное значение соотношения квоты cpu resource group.

selena_be_resource_group_cpu_use_ratio

- Unit: -
- Type: Average
- Description: Соотношение времени CPU, используемого resource group, к времени CPU всех resource groups.

selena_be_resource_group_mem_limit_bytes

- Unit: Bytes
- Type: Instantaneous
- Description: Мгновенное значение квоты памяти resource group.

selena_be_resource_group_mem_allocated_bytes

- Unit: Bytes
- Type: Instantaneous
- Description: Мгновенное значение использования памяти resource group.

selena_be_pipe_prepare_pool_queue_len

- Unit: Count
- Type: Instantaneous
- Description: Мгновенное значение длины очереди задач пула потоков prepare pipeline.

selena_fe_safe_mode

- Unit: -
- Type: Instantaneous
- Description: Указывает, включен ли Safe Mode. Допустимые значения: `0` (отключен) и `1` (включен). Когда Safe Mode включен, cluster больше не принимает запросы на загрузку.

selena_fe_unfinished_backup_job

- Unit: Count
- Type: Instantaneous
- Description: Указывает количество выполняющихся задач BACKUP в конкретном warehouse. Для shared-nothing cluster этот элемент мониторит только warehouse по умолчанию. Для shared-data cluster это значение всегда `0`.

selena_fe_unfinished_restore_job

- Unit: Count
- Type: Instantaneous
- Description: Указывает количество выполняющихся задач RESTORE в конкретном warehouse. Для shared-nothing cluster этот элемент мониторит только warehouse по умолчанию. Для shared-data cluster это значение всегда `0`.

selena_fe_memory_usage

- Unit: Bytes или Count
- Type: Instantaneous
- Description: Указывает статистику памяти для различных модулей в конкретном warehouse. Для shared-nothing cluster этот элемент мониторит только warehouse по умолчанию.

selena_fe_unfinished_query

- Unit: Count
- Type: Instantaneous
- Description: Указывает количество запросов, в настоящее время выполняющихся в конкретном warehouse. Для shared-nothing cluster этот элемент мониторит только warehouse по умолчанию.

selena_fe_last_finished_job_timestamp

- Unit: ms
- Type: Instantaneous
- Description: Указывает время окончания последнего запроса или загрузки в конкретном warehouse. Для shared-nothing cluster этот элемент мониторит только warehouse по умолчанию.

selena_fe_query_resource_group

- Unit: Count

- Type: Cumulative
- Description: Указывает общее количество запросов, выполненных в конкретной resource group.

selena_fe_query_resource_group_err

- Unit: Count
- Type: Cumulative
- Description: Указывает количество неудачных запросов в конкретной resource group.

selena_fe_query_resource_group_latency

- Unit: ms
- Type: Cumulative
- Description: Указывает статистику задержки для запросов в конкретной resource group.

selena_fe_tablet_num

- Unit: Count
- Type: Instantaneous
- Description: Указывает количество tablets на каждом узле BE.

selena_fe_tablet_max_compaction_score

- Unit: Count
- Type: Instantaneous
- Description: Указывает наивысший Compaction Score на каждом узле BE.

selena_fe_slow_lock_held_time_ms

- Unit: ms
- Type: Summary
- Description: Гистограмма, отслеживающая время удержания блокировки (в миллисекундах), когда обнаруживаются медленные блокировки. Эта метрика обновляется, когда время ожидания блокировки превышает параметр конфигурации `slow_lock_threshold_ms`. Она отслеживает максимальное время удержания блокировки среди всех владельцев блокировки при обнаружении события медленной блокировки. Каждая метрика включает значения квантилей (0.75, 0.95, 0.98, 0.99, 0.999), `_sum` и `_count` выводы. Примечание: Эта метрика может не точно отражать точное время удержания блокировки при высокой конкуренции, потому что метрика обновляется, как только время ожидания превышает порог, но время удержания может продолжать увеличиваться до тех пор, пока владелец не завершит свою

операцию и не освободит блокировку. Однако эта метрика все еще может быть обновлена даже при возникновении взаимной блокировки.

selena_fe_slow_lock_wait_time_ms

- Unit: ms
- Type: Summary
- Description: Гистограмма, отслеживающая время ожидания блокировки (в миллисекундах), когда обнаруживаются медленные блокировки. Эта метрика обновляется, когда время ожидания блокировки превышает параметр конфигурации `slow_lock_threshold_ms`. Она точно отслеживает, как долго потоки ждут получения блокировок во время сценариев конкуренции блокировки. Каждая метрика включает значения квантилей (0.75, 0.95, 0.98, 0.99, 0.999), `_sum` и `_count` выводы. Эта метрика обеспечивает точные измерения времени ожидания. Примечание: Эта метрика не может быть обновлена при возникновении взаимной блокировки, поэтому она не может использоваться для обнаружения ситуаций взаимной блокировки.

update_compaction_outputs_total

- Unit: Count
- Description: Общее количество compactions таблиц Primary Key.

update_del_vector_bytes_total

- Unit: Bytes
- Description: Общая память, используемая для кеширования DELETE vectors в таблицах Primary Key.

push_request_duration_us

- Unit: us
- Description: Общее время, затраченное на Spark Load.

writable_blocks_total (Устарело)

disks_data_used_capacity

- Description: Использованная емкость каждого диска (представленная путем хранения).

query_scan_rows

- Unit: Count
- Description: Общее количество просканированных строк.

update_primary_index_num

- Unit: Count
- Description: Количество индексов Primary Key, кешированных в памяти.

result_buffer_block_count

- Unit: Count
- Description: Количество блоков в буфере результатов.

query_scan_bytes

- Unit: Bytes
- Description: Общее количество просканированных байтов.

selena_be_files_scan_num_files_read

- Unit: Count
- Description: Количество файлов, прочитанных из внешнего хранилища (CSV, Parquet, ORC, JSON, Avro). Labels: `file_format`, `scan_type`.

selena_be_files_scan_num_bytes_read

- Unit: Bytes
- Description: Общее количество байтов, прочитанных из внешнего хранилища. Labels: `file_format`, `scan_type`.

selena_be_files_scan_num_raw_rows_read

- Unit: Count
- Description: Общее количество raw строк, прочитанных из внешнего хранилища перед проверкой формата и фильтрацией предикатов. Labels: `file_format`, `scan_type`.

selena_be_files_scan_num_valid_rows_read

- Unit: Count
- Description: Количество действительных строк, прочитанных (исключая строки с недопустимым форматом). Labels: `file_format`, `scan_type`.

selena_be_files_scan_num_rows_return

- Unit: Count
- Description: Количество строк, возвращенных после фильтрации предикатов. Labels: `file_format`, `scan_type`.

disk_reads_completed

- Unit: Count
- Description: Количество успешно завершенных чтений диска.

query_cache_hit_count

- Unit: Count
- Description: Количество попаданий кеша запросов.

jemalloc_resident_bytes

- Unit: Bytes
- Description: Максимальное количество байтов в физически резидентных страницах данных, отображенных allocator, включая все страницы, выделенные для метаданных allocator, страницы, поддерживающие активные выделения, и неиспользуемые грязные страницы.

blocks_open_writing (Устарело)

disk_io_time_weigthed

- Unit: ms
- Description: Взвешенное время, затраченное на I/Os.

update_compaction_task_byte_per_second

- Unit: Bytes/s
- Description: Расчетная скорость compactions таблиц Primary Key.

blocks_open_reading (Устарело)

tablet_update_max_compaction_score

- Unit: -
- Description: Наивысший compaction score tablets в таблицах Primary Key в текущем BE.

segment_read

- Unit: Count
- Description: Общее количество чтений segment.

disk_io_time_ms

- Unit: ms
- Description: Время, затраченное на I/Os.

load_mem_bytes

- Unit: Bytes
- Description: Стоимость памяти загрузки данных.

delta_column_group_get_non_pk_total

- Unit: Count
- Description: Общее количество раз получения delta column group (только для таблиц, не являющихся Primary Key).

query_scan_bytes_per_second

- Unit: Bytes/s
- Description: Расчетная скорость просканированных байтов в секунду.

active_scan_context_count

- Unit: Count
- Description: Общее количество задач сканирования, созданных Flink/Spark SQL.

fd_num_limit

- Unit: Count
- Description: Максимальное количество дескрипторов файлов.

update_compaction_task_cost_time_ns

- Unit: ns
- Description: Общее время, затраченное на compactions таблиц Primary Key.

delta_column_group_get_hit_cache

- Unit: Count
- Description: Общее количество попаданий кеша delta column group (только для таблиц Primary Key).

data_stream_receiver_count

- Unit: Count
- Description: Совокупное количество инстансов, выступающих в качестве приемников Exchange в BE.

bytes_written_total

- Unit: Bytes

- Description: Общее количество записанных байтов (sectors write * 512).

transaction_streaming_load_bytes

- Unit: Bytes
- Description: Общее количество загруженных байтов transaction load.

running_cumulative_compaction_task_num

- Unit: Count
- Description: Общее количество выполняющихся cumulative compactions.

transaction_streaming_load_requests_total

- Unit: Count
- Description: Общее количество запросов transaction load.

cpu

- Unit: -
- Description: Информация об использовании CPU, возвращенная `/proc/stat`.

update_del_vector_num

- Unit: Count
- Description: Количество элементов кеша DELETE vector в таблицах Primary Key.

disks_avail_capacity

- Description: Доступная емкость конкретного диска.

clone_mem_bytes

- Unit: Bytes
- Description: Память, используемая для клонирования replica.

fragment_requests_total

- Unit: Count
- Description: Общее количество инстансов fragment, выполняющихся на BE (для не-pipeline движка).

disk_write_time_ms

- Unit: ms
- Description: Время, затраченное на запись диска.

schema_change_mem_bytes

- Unit: Bytes
- Description: Память, используемая для Schema Change.

thrift_connections_total

- Unit: Count
- Description: Общее количество thrift-соединений (включая завершенные соединения).

thrift_opened_clients

- Unit: Count
- Description: Количество в настоящее время открытых thrift-клиентов.

thrift_used_clients

- Unit: Count
- Description: Количество thrift-клиентов, используемых в настоящее время.

thrift_current_connections (Устарело)

disk_bytes_read

- Unit: Bytes
- Description: Общее количество байтов, прочитанных с диска (sectors read * 512).

base_compaction_task_cost_time_ms

- Unit: ms
- Description: Общее время, затраченное на base compactions.

update_primary_index_bytes_total

- Unit: Bytes
- Description: Общая стоимость памяти индекса Primary Key.

compaction_deltas_total

- Unit: Count
- Description: Общее количество объединенных rowsets из base compactions и cumulative compactions.

max_network_receive_bytes_rate

- Unit: Bytes

- Description: Общее количество байтов, полученных по сети (максимальное значение среди всех сетевых интерфейсов).

max_network_send_bytes_rate

- Unit: Bytes
- Description: Общее количество байтов, отправленных по сети (максимальное значение среди всех сетевых интерфейсов).

chunk_allocator_mem_bytes

- Unit: Bytes
- Description: Память, используемая chunk allocator.

query_cache_usage_ratio

- Unit: -
- Description: Текущее соотношение использования кеша запросов.

process_fd_num_limit_soft

- Unit: Count
- Description: Мягкий лимит на максимальное количество дескрипторов файлов. Обратите внимание, что этот элемент указывает мягкий лимит. Вы можете установить жесткий лимит с помощью команды `ulimit`.

query_cache_hit_ratio

- Unit: -
- Description: Коэффициент попаданий кеша запросов.

tablet_metadata_mem_bytes

- Unit: Bytes
- Description: Память, используемая метаданными tablet.

jemalloc_retained_bytes

- Unit: Bytes
- Description: Общее количество байтов отображений виртуальной памяти, которые были сохранены, а не возвращены операционной системе через операции, такие как munmap(2).

unused_rowsets_count

- Unit: Count
- Description: Общее количество неиспользуемых rowsets. Обратите внимание, что эти rowsets будут восстановлены позже.

update_rowset_commit_apply_total

- Unit: Count
- Description: Общее количество COMMIT и APPLY для таблиц Primary Key.

segment_zonemap_mem_bytes

- Unit: Bytes
- Description: Память, используемая segment zonemap.

base_compaction_task_byte_per_second

- Unit: Bytes/s
- Description: Расчетная скорость base compactions.

transaction_streaming_load_duration_ms

- Unit: ms
- Description: Общее время, затраченное на Stream Load transaction Interface.

memory_pool_bytes_total

- Unit: Bytes
- Description: Память, используемая пулом памяти.

short_key_index_mem_bytes

- Unit: Bytes
- Description: Память, используемая индексом short key.

disk_bytes_written

- Unit: Bytes
- Description: Общее количество байтов, записанных на диск.

segment_flush_queue_count

- Unit: Count
- Description: Количество задач в очереди в пуле потоков segment flush.

jemalloc_metadata_bytes

- Unit: Bytes
- Description: Общее количество байтов, выделенных для метаданных, включая базовые выделения, используемые для чувствительных к bootstrap структур метаданных allocator и внутренних выделений. Использование transparent huge pages не включено в этот элемент.

network_send_bytes

- Unit: Bytes
- Description: Количество байтов, отправленных по сети.

update_mem_bytes

- Unit: Bytes
- Description: Память, используемая задачами APPLY таблиц Primary Key и индексом Primary Key.

blocks_created_total (Устарело)

query_cache_usage

- Unit: Bytes
- Description: Текущее использование кеша запросов.

memtable_flush_duration_us

- Unit: us
- Description: Общее время, затраченное на flush memtable.

push_request_write_bytes

- Unit: Bytes
- Description: Общее количество байтов, записанных через Spark Load.

jemalloc_active_bytes

- Unit: Bytes
- Description: Общее количество байтов в активных страницах, выделенных приложением.

load_rpc_threadpool_size

- Unit: Count

- Description: Текущий размер пула потоков RPC, который используется для обработки Routine Load и загрузки через табличные функции. Значение по умолчанию - 10, с максимальным значением 1000. Это значение динамически настраивается на основе использования пула потоков.

publish_version_queue_count

- Unit: Count
- Description: Количество задач в очереди в пуле потоков Publish Version.

chunk_pool_system_free_count

- Unit: Count
- Description: Метрики выделения/кеша chunk памяти.

chunk_pool_system_free_cost_ns

- Unit: ns
- Description: Метрики выделения/кеша chunk памяти.

chunk_pool_system_alloc_count

- Unit: Count
- Description: Метрики выделения/кеша chunk памяти.

chunk_pool_system_alloc_cost_ns

- Unit: ns
- Description: Метрики выделения/кеша chunk памяти.

chunk_pool_local_core_alloc_count

- Unit: Count
- Description: Метрики выделения/кеша chunk памяти.

chunk_pool_other_core_alloc_count

- Unit: Count
- Description: Метрики выделения/кеша chunk памяти.

fragment_endpoint_count

- Unit: Count
- Description: Совокупное количество инстансов, выступающих в качестве отправителей Exchange в BE.

disk_sync_total (Устарело)

max_disk_io_util_percent

- Unit: -
- Description: Максимальный процент использования I/O диска.

network_receive_bytes

- Unit: Bytes
- Description: Общее количество байтов, полученных через сеть.

storage_page_cache_mem_bytes

- Unit: Bytes
- Description: Память, используемая кешем страниц хранилища.

jit_cache_mem_bytes

- Unit: Bytes
- Description: Память, используемая кешем jit-скомпилированных функций.

column_partial_update_apply_total

- Unit: Count
- Description: Общее количество APPLY для частичных обновлений по столбцу (режим Column)

disk_writes_completed

- Unit: Count
- Description: Общее количество успешно завершенных записей диска.

memtable_flush_total

- Unit: Count
- Description: Общее количество flush memtable.

page_cache_hit_count

- Unit: Count
- Description: Общее количество попаданий в кеш страниц хранилища.

page_cache_insert_count

- Unit: Count
- Description: Общее количество операций вставки в кеш страниц хранилища.

page_cache_insert_evict_count

- Unit: Count
- Description: Общее количество записей кеша, вытесненных во время операций вставки из-за ограничений емкости.

page_cache_release_evict_count

- Unit: Count
- Description: Общее количество записей кеша, вытесненных во время операций освобождения, когда использование кеша превышает емкость.

bytes_read_total (Устарело)

update_rowset_commit_request_total

- Unit: Count
- Description: Общее количество запросов COMMIT rowset в таблицах Primary Key.

tablet_cumulative_max_compaction_score

- Unit: -
- Description: Наивысший cumulative compaction score tablets в этом BE.

column_zonemap_index_mem_bytes

- Unit: Bytes
- Description: Память, используемая column zonemaps.

push_request_write_bytes_per_second

- Unit: Bytes/s
- Description: Скорость записи данных Spark Load.

process_thread_num

- Unit: Count
- Description: Общее количество потоков в этом процессе.

query_cache_lookup_count

- Unit: Count
- Description: Общее количество поисков кеша запросов.

http_requests_total (Устарело)

http_request_send_bytes (Устарело)

cumulative_compaction_task_cost_time_ms

- Unit: ms
- Description: Общее время, затраченное на cumulative compactions.

column_metadata_mem_bytes

- Unit: Bytes
- Description: Память, используемая метаданными столбцов.

plan_fragment_count

- Unit: Count
- Description: Количество в настоящее время выполняющихся фрагментов плана запросов.

page_cache_lookup_count

- Unit: Count
- Description: Общее количество поисков кеша страниц хранилища.

query_mem_bytes

- Unit: Bytes
- Description: Память, используемая запросами.

load_channel_count

- Unit: Count
- Description: Общее количество каналов загрузки.

push_request_write_rows

- Unit: Count
- Description: Общее количество строк, записанных через Spark Load.

running_base_compaction_task_num

- Unit: Count
- Description: Общее количество выполняющихся задач base compaction.

fragment_request_duration_us

- Unit: us

- Description: Совокупное время выполнения инстансов fragment (для не-pipeline движка).

process_mem_bytes

- Unit: Bytes
- Description: Память, используемая этим процессом.

broker_count

- Unit: Count
- Description: Общее количество созданных brokers файловой системы (по адресу хоста).

segment_replicate_queue_count

- Unit: Count
- Description: Количество задач в очереди в пуле потоков Segment Replicate.

brpc_endpoint_stub_count

- Unit: Count
- Description: Общее количество bRPC stubs (по адресу).

readable_blocks_total (Устарело)

delta_column_group_get_non_pk_hit_cache

- Unit: Count
- Description: Общее количество попаданий в кеш delta column group (для таблиц, не являющихся Primary Key).

update_del_vector_deletes_new

- Unit: Count
- Description: Общее количество вновь сгенерированных DELETE vectors, используемых в таблицах Primary Key.

compaction_bytes_total

- Unit: Bytes
- Description: Общее количество объединенных байтов из base compactions и cumulative compactions.

segment_metadata_mem_bytes

- Unit: Bytes
- Description: Память, используемая метаданными segment.

column_partial_update_apply_duration_us

- Unit: us
- Description: Общее время, затраченное на задачи APPLY частичных обновлений столбцов (режим Column).

bloom_filter_index_mem_bytes

- Unit: Bytes
- Description: Память, используемая индексами Bloomfilter.

routine_load_task_count

- Unit: Count
- Description: Количество в настоящее время выполняющихся задач Routine Load.

delta_column_group_get_total

- Unit: Count
- Description: Общее количество раз получения delta column groups (для таблиц Primary Key).

disk_read_time_ms

- Unit: ms
- Description: Время, затраченное на чтение с диска.

update_compaction_outputs_bytes_total

- Unit: Bytes
- Description: Общее количество байтов, записанных compactions таблиц Primary Key.

memtable_flush_queue_count

- Unit: Count
- Description: Количество задач в очереди в пуле потоков flush memtable.

query_cache_capacity

- Description: Емкость кеша запросов.

streaming_load_duration_ms

- Unit: ms
- Description: Общее время, затраченное на Stream Load.

streaming_load_current_processing

- Unit: Count
- Description: Количество в настоящее время выполняющихся задач Stream Load.

streaming_load_bytes

- Unit: Bytes
- Description: Общее количество байтов, загруженных Stream Load.

load_rows

- Unit: Count
- Description: Общее количество загруженных строк.

load_bytes

- Unit: Bytes
- Description: Общее количество загруженных байтов.

meta_request_total

- Unit: Count
- Description: Общее количество запросов чтения/записи meta.

meta_request_duration

- Unit: us
- Description: Общая продолжительность чтения/записи meta.

tablet_base_max_compaction_score

- Unit: -
- Description: Наивысший base compaction score tablets в этом BE.

page_cache_capacity

- Description: Емкость кеша страниц хранилища.

cumulative_compaction_task_byte_per_second

- Unit: Bytes/s

- Description: Скорость обработанных байтов во время cumulative compactions.

stream_load

- Unit: -
- Description: Общее количество загруженных строк и полученных байтов.

stream_load_pipe_count

- Unit: Count
- Description: Количество в настоящее время выполняющихся задач Stream Load.

binary_column_pool_bytes

- Unit: Bytes
- Description: Память, используемая пулом столбцов BINARY.

int16_column_pool_bytes

- Unit: Bytes
- Description: Память, используемая пулом столбцов INT16.

decimal_column_pool_bytes

- Unit: Bytes
- Description: Память, используемая пулом столбцов DECIMAL.

double_column_pool_bytes

- Unit: Bytes
- Description: Память, используемая пулом столбцов DOUBLE.

int128_column_pool_bytes

- Unit: Bytes
- Description: Память, используемая пулом столбцов INT128.

date_column_pool_bytes

- Unit: Bytes
- Description: Память, используемая пулом столбцов DATE.

int64_column_pool_bytes

- Unit: Bytes
- Description: Память, используемая пулом столбцов INT64.

int8_column_pool_bytes

- Unit: Bytes
- Description: Память, используемая пулом столбцов INT8.

datetime_column_pool_bytes

- Unit: Bytes
- Description: Память, используемая пулом столбцов DATETIME.

int32_column_pool_bytes

- Unit: Bytes
- Description: Память, используемая пулом столбцов INT32.

float_column_pool_bytes

- Unit: Bytes
- Description: Память, используемая пулом столбцов FLOAT.

uint8_column_pool_bytes

- Unit: Bytes
- Description: Байты, используемые пулом столбцов UINT8.

column_pool_mem_bytes

- Unit: Bytes
- Description: Память, используемая пулами столбцов.

local_column_pool_bytes (Устарело)

total_column_pool_bytes (Устарело)

central_column_pool_bytes (Устарело)

wait_base_compaction_task_num

- Unit: Count
- Description: Количество задач base compaction, ожидающих выполнения.

wait_cumulative_compaction_task_num

- Unit: Count
- Description: Количество задач cumulative compaction, ожидающих выполнения.

jemalloc_allocated_bytes

- Unit: Bytes
- Description: Общее количество байтов, выделенных приложением.

pk_index_compaction_queue_count

- Unit: Count
- Description: Количество задач в очереди в пуле потоков compaction индекса Primary Key.

disks_total_capacity

- Description: Общая емкость диска.

disks_state

- Unit: -
- Description: Состояние каждого диска. `1` указывает, что диск используется, а `0` указывает, что он не используется.

update_del_vector_deletes_total (Устарело)

update_del_vector_dels_num (Устарело)

result_block_queue_count

- Unit: Count
- Description: Количество результатов в очереди блоков результатов.

engine_requests_total

- Unit: Count
- Description: Общее количество всех типов запросов между BE и FE, включая CREATE TABLE, Publish Version и клонирование tablet.

snmp

- Unit: -
- Description: Метрики, возвращенные `/proc/net/snmp`.

compaction_mem_bytes

- Unit: Bytes
- Description: Память, используемая compactions.

txn_request

- Unit: -
- Description: Транзакционные запросы BEGIN, COMMIT, ROLLBACK и EXEC.

small_file_cache_count

- Unit: Count
- Description: Количество кешей небольших файлов.

rowset_metadata_mem_bytes

- Unit: Bytes
- Description: Общее количество байтов для метаданных rowset.

update_apply_queue_count

- Unit: Count
- Description: Количество задач в очереди в пуле потоков APPLY транзакции таблиц Primary Key.

async_delta_writer_queue_count

- Unit: Count
- Description: Количество задач в очереди в пуле потоков delta writer tablet.

update_compaction_duration_us

- Unit: us
- Description: Общее время, затраченное на compactions таблиц Primary Key.

transaction_streaming_load_current_processing

- Unit: Count
- Description: Количество в настоящее время выполняющихся задач транзакционного Stream Load.

ordinal_index_mem_bytes

- Unit: Bytes
- Description: Память, используемая ordinal индексами.

consistency_mem_bytes

- Unit: Bytes
- Description: Память, используемая проверками согласованности replica.

tablet_schema_mem_bytes

- Unit: Bytes
- Description: Память, используемая schema tablet.

fd_num_used

- Unit: Count
- Description: Количество дескрипторов файлов, используемых в настоящее время.

running_update_compaction_task_num

- Unit: Count
- Description: Общее количество в настоящее время выполняющихся задач compaction таблиц Primary Key.

rowset_count_generated_and_in_use

- Unit: Count
- Description: Количество ID rowset, используемых в настоящее время.

bitmap_index_mem_bytes

- Unit: Bytes
- Description: Память, используемая bitmap индексами.

update_rowset_commit_apply_duration_us

- Unit: us
- Description: Общее время, затраченное на задачи APPLY таблиц Primary Key.

process_fd_num_used

- Unit: Count
- Description: Количество дескрипторов файлов, используемых в настоящее время в этом процессе BE.

network_send_packets

- Unit: Count
- Description: Общее количество пакетов, отправленных через сеть.

network_receive_packets

- Unit: Count
- Description: Общее количество пакетов, полученных через сеть.

metadata_mem_bytes (Устарело)

push_requests_total

- Unit: Count
- Description: Общее количество успешных и неудачных запросов Spark Load.

blocks_deleted_total (Устарело)

jemalloc_mapped_bytes

- Unit: Bytes
- Description: Общее количество байтов в активных extents, отображенных allocator.

process_fd_num_limit_hard

- Unit: Count
- Description: Жесткий лимит на максимальное количество дескрипторов файлов.

jemalloc_metadata_thp

- Unit: Count
- Description: Количество Transparent Huge Pages, используемых для метаданных.

update_rowset_commit_request_failed

- Unit: Count
- Description: Общее количество неудачных запросов COMMIT rowset в таблицах Primary Key.

streaming_load_requests_total

- Unit: Count
- Description: Общее количество запросов Stream Load.

resource_group_running_queries

- Unit: Count
- Description: Количество запросов, в настоящее время выполняющихся в каждой resource group. Это мгновенное значение.

resource_group_total_queries

- Unit: Count
- Description: Общее количество запросов, выполненных в каждой resource group, включая те, которые в настоящее время выполняются. Это мгновенное значение.

resource_group_bigquery_count

- Unit: Count
- Description: Количество запросов в каждой resource group, которые вызвали лимит для больших запросов. Это мгновенное значение.

resource_group_concurrency_overflow_count

- Unit: Count
- Description: Количество запросов в каждой resource group, которые вызвали лимит concurrency. Это мгновенное значение.

resource_group_mem_limit_bytes

- Unit: Bytes
- Description: Лимит памяти для каждой resource group, измеренный в байтах. Это мгновенное значение.

resource_group_mem_inuse_bytes

- Unit: Bytes
- Description: В настоящее время используемая память каждой resource group, измеренная в байтах. Это мгновенное значение.

resource_group_cpu_limit_ratio

- Unit: -
- Description: Соотношение лимита ядер CPU для каждой resource group к общему лимиту ядер CPU для всех resource groups. Это мгновенное значение.

resource_group_inuse_cpu_cores

- Unit: Count
- Description: Расчетное количество ядер CPU, в настоящее время используемых каждой resource group. Это среднее значение за интервал времени между двумя получениями метрик.

resource_group_cpu_use_ratio (Устарело)

- Unit: -
- Description: Соотношение временных срезов потока pipeline, используемых каждой resource group, к общему количеству, используемому всеми resource groups. Это среднее значение за интервал времени между двумя получениями метрик.

resource_group_connector_scan_use_ratio (Устарело)

- Unit: -
- Description: Соотношение временных срезов потока сканирования внешних таблиц, используемых каждой resource group, к общему количеству, используемому всеми resource groups. Это среднее значение за интервал времени между двумя получениями метрик.

resource_group_scan_use_ratio (Устарело)

- Unit: -
- Description: Соотношение временных срезов потока сканирования внутренних таблиц, используемых каждой resource group, к общему количеству, используемому всеми resource groups. Это среднее значение за интервал времени между двумя получениями метрик.

pipe_poller_block_queue_len

- Unit: Count
- Description: Текущая длина очереди блоков PipelineDriverPoller в pipeline движке.

pip_query_ctx_cnt

- Unit: Count
- Description: Общее количество в настоящее время выполняющихся запросов в BE.

pipe_driver_schedule_count

- Unit: Count
- Description: Совокупное количество раз планирования driver для pipeline executors в BE.

pipe_scan_executor_queuing

- Unit: Count
- Description: Текущее количество ожидающих асинхронных задач I/O, запущенных Scan Operators.

pipe_driver_queue_len

- Unit: Count
- Description: Текущее количество готовых drivers в готовой очереди, ожидающих планирования в BE.

pipe_driver_execution_time

- Description: Совокупное время, затраченное executors PipelineDriver на обработку PipelineDrivers.

pipe_prepare_pool_queue_len

- Unit: Count
- Description: Количество задач в очереди в пуле потоков PREPARE pipeline. Это мгновенное значение.

selena_be_exec_state_report_active_threads

- Unit: Count
- Type: Instantaneous
- Description: Количество задач, выполняемых в пуле потоков, который отчитывается о состоянии выполнения инстанса Fragment.

selena_be_exec_state_report_running_threads

- Unit: Count
- Type: Instantaneous
- Description: Количество потоков в пуле потоков, который отчитывается о состоянии выполнения инстанса Fragment, с минимумом 1 и максимумом 2.

selena_be_exec_state_report_threadpool_size

- Unit: Count
- Type: Instantaneous
- Description: Максимальное количество потоков в пуле потоков, который отчитывается о состоянии выполнения инстанса Fragment, по умолчанию 2.

selena_be_exec_state_report_queue_count

- Unit: Count
- Type: Instantaneous
- Description: Количество задач в очереди в пуле потоков, который отчитывается о состоянии выполнения инстанса Fragment, до максимума 1000.

selena_be_priority_exec_state_report_active_threads

- Unit: Count
- Type: Instantaneous

- Description: Количество задач, выполняемых в пуле потоков, который отчитывается о финальном состоянии выполнения инстанса Fragment.

selena_be_priority_exec_state_report_running_threads

- Unit: Count
- Type: Instantaneous
- Description: Количество потоков в пуле потоков, который отчитывается о финальном состоянии выполнения инстанса Fragment, с минимумом 1 и максимумом 2.

selena_be_priority_exec_state_report_threadpool_size

- Unit: Count
- Type: Instantaneous
- Description: Максимальное количество потоков в пуле потоков, который отчитывается о финальном состоянии выполнения инстанса Fragment, по умолчанию 2.

selena_be_priority_exec_state_report_queue_count

- Unit: Count
- Type: Instantaneous
- Description: Количество задач в очереди в пуле потоков, который отчитывается о финальном состоянии выполнения инстанса Fragment, до максимума 2147483647.

selena_fe_routine_load_jobs

- Unit: Count

- Description: Общее количество задач Routine Load в различных состояниях. Например:

```
selena_fe_routine_load_jobs{state="NEED_SCHEDULE"} 0
selena_fe_routine_load_jobs{state="RUNNING"} 1
selena_fe_routine_load_jobs{state="PAUSED"} 0
selena_fe_routine_load_jobs{state="STOPPED"} 0
selena_fe_routine_load_jobs{state="CANCELLED"} 1
selena_fe_routine_load_jobs{state="UNSTABLE"} 0
```

selena_fe_routine_load_paused

- Unit: Count
- Description: Общее количество раз, когда задачи Routine Load приостанавливаются.

selena_fe_routine_load_rows

- Unit: Count
- Description: Общее количество строк, загруженных всеми задачами Routine Load.

selena_fe_routine_load_receive_bytes

- Unit: Byte
- Description: Общее количество данных, загруженных всеми задачами Routine Load.

selena_fe_routine_load_error_rows

- Unit: Count
- Description: Общее количество строк с ошибками, встреченных во время загрузки данных всеми задачами Routine Load.

selena_fe_routine_load_max_lag_of_partition

- Unit: -
- Description: Максимальное отставание offset partition Kafka для каждой задачи Routine Load. Собирается только тогда, когда конфигурация FE `enable_routine_load_lag_metrics` установлена на `true`, а отставание offset больше или равно конфигурации FE `min_routine_load_lag_for_metrics`. По умолчанию `enable_routine_load_lag_metrics` равно `false`, а `min_routine_load_lag_for_metrics` равно `10000`.

selena_fe_routine_load_max_lag_time_of_partition

- Unit: Seconds
- Description: Максимальное отставание timestamp offset partition Kafka для каждой задачи Routine Load. Собирается только тогда, когда конфигурация FE `enable_routine_load_lag_time_metrics` установлена на `true`. По умолчанию `enable_routine_load_lag_time_metrics` равно `false`.

selena_fe_sql_block_hit_count

- Unit: Count
- Description: Количество раз, когда sql из черного списка были перехвачены.

selena_fe_scheduled_pending_tablet_num

- Unit: Count
- Type: Instantaneous

- Description: Количество задач Clone в состоянии Pending, запланированных FE, включая как типы BALANCE, так и REPAIR.

selena_fe_scheduled_running_tablet_num

- Unit: Count
- Type: Instantaneous
- Description: Количество задач Clone в состоянии Running, запланированных FE, включая как типы BALANCE, так и REPAIR.

selena_fe_clone_task_total

- Unit: Count
- Type: Cumulative
- Description: Общее количество задач Clone в cluster.

selena_fe_clone_task_success

- Unit: Count
- Type: Cumulative
- Description: Количество успешно выполненных задач Clone в cluster.

selena_fe_clone_task_copy_bytes

- Unit: Bytes
- Type: Cumulative
- Description: Общий размер файлов, скопированных задачами Clone в cluster, включая как типы INTER_NODE, так и INTRA_NODE.

selena_fe_clone_task_copy_duration_ms

- Unit: ms
- Type: Cumulative
- Description: Общее время для копирования, потребленное задачами Clone в cluster, включая как типы INTER_NODE, так и INTRA_NODE.

selena_be_clone_task_copy_bytes

- Unit: Bytes
- Type: Cumulative
- Description: Общий размер файлов, скопированных задачами Clone на узле BE, включая как типы INTER_NODE, так и INTRA_NODE.

selena_be_clone_task_copy_duration_ms

- Unit: ms
- Type: Cumulative
- Description: Общее время для копирования, потребленное задачами Clone на узле BE, включая как типы INTER_NODE, так и INTRA_NODE.

Метрики задержки транзакций

Следующие метрики являются метриками типа `summary`, которые предоставляют распределения задержки для различных фаз транзакции. Эти метрики отчитываются исключительно узлом Leader FE.

Каждая метрика включает следующие выводы:

- **Quantiles**: Значения задержки на различных процентильных границах. Они предоставляются через label `quantile`, который может иметь значения `0.75`, `0.95`, `0.98`, `0.99` и `0.999`.
- **`<metric_name>_sum`**: Общее совокупное время, затраченное на эту фазу, например, `selena_fe_txn_total_latency_ms_sum`.
- **`<metric_name>_count`**: Общее количество транзакций, записанных для этой фазы, например, `selena_fe_txn_total_latency_ms_count`.

Все метрики транзакций имеют следующие labels:

- `type`: Категоризирует транзакции по типу источника задачи загрузки (например, `all`, `stream_load`, `routine_load`). Это позволяет мониторить как общую производительность транзакций, так и производительность конкретных типов загрузки. Отчетные группы можно настроить через параметр FE `txn_latency_metric_report_groups`.
- `is_leader`: Указывает, является ли отчитывающийся узел FE Leader. Только Leader FE (`is_leader="true"`) отчитывает фактические значения метрик. Followers будут иметь `is_leader="false"` и не отчитывать данные.

selena_fe_txn_total_latency_ms

- Unit: ms
- Type: Summary
- Description: Общая задержка для завершения транзакции, измеренная от времени `prepare` до времени `finish`. Эта метрика представляет полную end-to-end

продолжительность транзакции.

selena_fe_txn_write_latency_ms

- Unit: ms
- Type: Summary
- Description: Задержка фазы `write` транзакции, от времени `prepare` до времени `commit`. Эта метрика изолирует производительность этапа записи данных и подготовки перед тем, как транзакция будет готова к публикации.

selena_fe_txn_publish_latency_ms

- Unit: ms
- Type: Summary
- Description: Задержка фазы `publish`, от времени `commit` до времени `finish`. Это продолжительность, которую требуется для того, чтобы зафиксированная транзакция стала видимой для запросов. Это сумма подфаз `schedule`, `execute` и `ack`.

selena_fe_txn_publish_schedule_latency_ms

- Unit: ms
- Type: Summary
- Description: Время, которое транзакция проводит в ожидании публикации после того, как она была зафиксирована, измеренное от времени `commit` до момента, когда задача публикации принята. Эта метрика отражает задержки планирования или время ожидания в очереди в pipeline `publish`.

selena_fe_txn_publish_execute_latency_ms

- Unit: ms
- Type: Summary
- Description: Активное время выполнения задачи `publish`, от момента, когда задача принята, до момента ее завершения. Эта метрика представляет фактическое время, затрачиваемое на то, чтобы сделать изменения транзакции видимыми.

selena_fe_txn_publish_ack_latency_ms

- Unit: ms
- Type: Summary
- Description: Финальная задержка подтверждения, от момента завершения задачи `publish` до финального времени `finish`, когда транзакция помечается как `VISIBLE`. Эта метрика включает любые финальные шаги или подтверждения, необходимые.

Метрики Merge Commit BE

merge_commit_request_total

- Unit: Count
- Type: Cumulative
- Description: Общее количество запросов merge commit, полученных BE.

merge_commit_request_bytes

- Unit: Bytes
- Type: Cumulative
- Description: Общее количество байтов данных, полученных через запросы merge commit.

merge_commit_success_total

- Unit: Count
- Type: Cumulative
- Description: Запросы merge commit, завершенные успешно.

merge_commit_fail_total

- Unit: Count
- Type: Cumulative
- Description: Запросы merge commit, завершенные сбоем.

merge_commit_pending_total

- Unit: Count
- Type: Instantaneous
- Description: Задачи merge commit, в настоящее время ожидающие в очереди выполнения.

merge_commit_pending_bytes

- Unit: Bytes
- Type: Instantaneous
- Description: Общее количество байтов данных, удерживаемых ожидающими задачами merge commit.

merge_commit_send_rpc_total

- Unit: Count
- Type: Cumulative

- Description: RPC-запросы, отправленные на FE для запуска операций merge commit.

merge_commit_register_pipe_total

- Unit: Count
- Type: Cumulative
- Description: Pipes загрузки stream, зарегистрированные для операций merge commit.

merge_commit_unregister_pipe_total

- Unit: Count
- Type: Cumulative
- Description: Pipes загрузки stream, отмененные от операций merge commit.

Метрики задержки предоставляют серии процентилей, такие как `merge_commit_request_latency_99` и `merge_commit_request_latency_90`, отчитываемые в микросекундах. End-to-end задержка соответствует:

`merge_commit_request = merge_commit_pending + merge_commit_wait_plan + merge_commit_append_pipe + merge_commit_wait_finish`

> **Примечание**: До v1.5.2 и v2.0.0 эти метрики задержки отчитывались в наносекундах.

merge_commit_request

- Unit: микросекунда
- Type: Summary
- Description: End-to-end задержка обработки для запросов merge commit.

merge_commit_pending

- Unit: микросекунда
- Type: Summary
- Description: Время, которое задачи merge commit проводят в ожидании в ожидающей очереди перед выполнением.

merge_commit_wait_plan

- Unit: микросекунда
- Type: Summary
- Description: Комбинированная задержка для RPC-запроса и ожидания доступности pipe загрузки stream.

merge_commit_append_pipe

- Unit: микросекунда
- Type: Summary
- Description: Время, затраченное на добавление данных в pipe загрузки stream во время merge commit.

merge_commit_wait_finish

- Unit: микросекунда
- Type: Summary
- Description: Время, затраченное на ожидание завершения операций загрузки merge commit.

Метрики мониторинга для асинхронных материализованных представлений

Начиная с версии v1.5.2, Selena поддерживает метрики для асинхронных материализованных представлений.

Чтобы разрешить Prometheus доступ к метаданным материализованных представлений в вашем cluster, вы должны добавить следующие конфигурации в файл конфигурации Prometheus **prometheus/prometheus.yml**:

```
global:
....
scrape_configs:

  - job_name: 'dev'
    metrics_path: '/metrics'
    # Добавьте следующие конфигурации.
    basic_auth:
      username: 'root'
      password: ''
    params:
      'with_materialized_view_metrics' : ['all']
....
```

- `username`: Имя пользователя, используемое для входа в ваш cluster Selena. Если не используется учётная запись root, пользователю должны быть предоставлены роли `user_admin` и `db_admin`.

- `password`: Пароль, используемый для входа в ваш cluster Selena.
- `'with_materialized_view_metrics'`: Область собираемых метрик. Допустимые значения:
  - `'all'`: Собираются все метрики, относящиеся к материализованным представлениям.
  - `'minified'`: Метрики типа Gauge и метрики со значением `0` не собираются.

Элементы метрик

mv_refresh_jobs

- Тип: Counter
- Описание: Общее количество заданий обновления материализованного представления.

mv_refresh_total_success_jobs

- Тип: Counter
- Описание: Количество успешных заданий обновления материализованного представления.

mv_refresh_total_failed_jobs

- Тип: Counter
- Описание: Количество неудачных заданий обновления материализованного представления.

mv_refresh_total_empty_jobs

- Тип: Counter
- Описание: Количество отменённых заданий обновления материализованного представления, потому что данные для обновления пусты.

mv_refresh_total_retry_meta_count

- Тип: Counter
- Описание: Количество раз, когда задание обновления материализованного представления проверяет, обновлена ли базовая таблица.

mv_query_total_count

- Тип: Counter
- Описание: Количество раз, когда материализованное представление используется в предварительной обработке запроса.

mv_query_total_hit_count

- Тип: Counter
- Описание: Количество раз, когда материализованное представление считается способным переписать запрос в плане запроса. Это значение может казаться выше, потому что окончательный план запроса может пропустить переписывание из-за высокой стоимости.

mv_query_total_considered_count

- Тип: Counter
- Описание: Количество раз, когда материализованное представление переписывает запрос (исключая прямые запросы к материализованному представлению).

mv_query_total_matched_count

- Тип: Counter
- Описание: Количество раз, когда материализованное представление участвует в окончательном плане запроса (включая прямые запросы к материализованному представлению).

mv_refresh_pending_jobs

- Тип: Gauge
- Описание: Количество ожидающих заданий обновления материализованного представления в данный момент.

mv_refresh_running_jobs

- Тип: Gauge
- Описание: Количество выполняющихся заданий обновления материализованного представления в данный момент.

mv_row_count

- Тип: Gauge
- Описание: Количество строк материализованного представления.

mv_storage_size

- Тип: Gauge
- Описание: Размер материализованного представления. Единица: Байт.

mv_inactive_state

- Тип: Gauge
- Описание: Статус материализованного представления. Допустимые значения: `0`(активно) и `1`(неактивно).

mv_partition_count

- Тип: Gauge
- Описание: Количество партиций в материализованном представлении. Значение равно `0`, если материализованное представление не партиционировано.

mv_refresh_duration

- Тип: Histogram
- Описание: Продолжительность успешных заданий обновления материализованного представления.

Метрики мониторинга для cluster с разделяемыми данными

Selena предоставляет два шаблона Dashboard для cluster с разделяемыми данными:

- Shared-data Dashboard
- Starlet Dashboard

Shared-data Dashboard

Shared-data Dashboard включает следующие категории метрик мониторинга:

- Publish Version
- Metadata
- Metacache
- Vacuum
- Loading

Publish Version

Latency / QPS

- Описание: Квантильная задержка, средняя задержка и QPS задач Public Version.

Queued Tasks

- Описание: Количество задач Public Version в очереди.

Metadata

Get Tablet Metadata

- Описание: Квантильная задержка, средняя задержка и QPS задач Get Tablet Metadata.

Put Tablet Metadata

- Описание: Квантильная задержка, средняя задержка и QPS задач Put Tablet Metadata.

Get Txn Log

- Описание: Квантильная задержка, средняя задержка и QPS задач Get Txn Log.

Put Txn Log

- Описание: Квантильная задержка, средняя задержка и QPS задач Put Txn Log.

Metacache

Metacache Usage

- Описание: Коэффициент использования Metacache.

Delvec Cache Miss Per Minute

- Описание: Количество промахов кэша Delvec Cache в минуту.

Metadata Cache Miss Per Minute

- Описание: Количество промахов кэша Metadata Cache в минуту.

Txn Log Cache Miss Per Minute

- Описание: Количество промахов кэша Txn Log Cache в минуту.

Segment Cache Miss Per Minute

- Описание: Количество промахов кэша Segment Cache в минуту.

Vacuum

Vacuum Deletes

- Описание: Квантильная задержка, средняя задержка и QPS задач Vacuum Deletes.

Errors

- Описание: Количество неудачных операций Vacuum Deletes.

© ООО «Платформа Данных Селена», 2026

Loading

Queue Size

- Описание: Размер очереди BE Async Delta Writer.

Starlet Dashboard

Starlet Dashboard включает следующие категории метрик мониторинга:

- FSLIB READ IO METRICS
- FSLIB WRITE IO METRICS
- S3 IO METRICS
- FSLIB CACHE METRICS
- FSLIB FS METRICS

FSLIB READ IO METRICS

fslib read io_latency (quantile)

- Тип: Histogram
- Описание: Квантильная задержка для чтения S3.

fslib read io_latency (average)

- Тип: Counter
- Описание: Средняя задержка для чтения S3.

fslib total read data

- Тип: Counter
- Описание: Общий размер данных для чтения S3.

fslib read iosize (quantile)

- Тип: Histogram
- Описание: Квантильный размер I/O для чтения S3.

fslib read iosize (average)

- Тип: Counter
- Описание: Средний размер I/O для чтения S3.

fslib read throughput

- Тип: Counter
- Описание: Пропускная способность I/O в секунду для чтения S3.

fslib read iops

- Тип: Counter
- Описание: Количество операций I/O в секунду для чтения S3.

FSLIB WRITE IO METRICS

fslib write io_latency (quantile)

- Тип: Histogram
- Описание: Квантильная задержка для записи приложения. Обратите внимание, что это значение может казаться ниже, потому что эта метрика отслеживает только данные, записанные в буфер.

fslib write io_latency (average)

- Тип: Counter
- Описание: Средняя задержка для записи приложения. Обратите внимание, что это значение может казаться ниже, потому что эта метрика отслеживает только данные, записанные в буфер.

fslib total write data

- Тип: Counter
- Описание: Общий размер данных для записи приложения.

fslib write iosize (quantile)

- Тип: Histogram
- Описание: Квантильный размер I/O для записи приложения.

fslib write iosize (average)

- Тип: Counter
- Описание: Средний размер I/O для записи приложения.

fslib write throughput

- Тип: Counter
- Описание: Пропускная способность I/O в секунду для записи приложения.

S3 IO METRICS

fslib s3 single upload iops

- Тип: Counter
- Описание: Количество вызовов в секунду для S3 Put Object.

fslib s3 single upload iosize (quantile)

- Тип: Histogram
- Описание: Квантильный размер I/O для S3 Put Object.

fslib s3 single upload latency (quantile)

- Тип: Histogram
- Описание: Квантильная задержка для S3 Put Object.

fslib s3 multi upload iops

- Тип: Counter
- Описание: Количество вызовов в секунду для S3 Multi Upload Object.

fslib s3 multi upload iosize (quantile)

- Тип: Histogram
- Описание: Квантильный размер I/O для S3 Multi Upload Object.

fslib s3 multi upload latency (quantile)

- Тип: Histogram
- Описание: Квантильная задержка для S3 Multi Upload Object.

fslib s3 complete multi upload latency (quantile)

- Тип: Histogram
- Описание: Квантильная задержка для S3 Complete Multi Upload Object.

FSLIB CACHE METRICS

fslib cache hit ratio

- Тип: Counter
- Описание: Коэффициент попадания в кэш.

fslib cache hits/misses

- Тип: Counter
- Описание: Количество попаданий/промахов кэша в секунду.

FSLIB FS METRICS

fslib alive fs instances count

- Тип: Gauge
- Описание: Количество активных экземпляров файловой системы.

fslib open files

- Тип: Counter
- Описание: Накопленное количество открытых файлов.

fslib create files

- Тип: Counter
- Описание: Среднее количество созданных файлов в секунду.

filesystem meta operations

- Тип: Counter
- Описание: Среднее количество операций листинга директорий в секунду.

fslib async caches

- Тип: Counter
- Описание: Накопленное количество файлов в асинхронном кэше.

fslib create files (TOTAL)

- Тип: Counter
- Описание: Накопленное количество созданных файлов.

fslib async tasks

- Тип: Counter
- Описание: Накопленное количество асинхронных задач в очереди.

Управление алертами

В этом разделе представлены различные элементы алертов с разных измерений, включая непрерывность бизнеса, доступность cluster и нагрузку машины, и предоставляются соответствующие решения.

> **ПРИМЕЧАНИЕ**
>
> В следующих примерах все переменные имеют префикс `$`. Они должны быть заменены в соответствии с вашей бизнес-средой. Например, `$job_name` должен быть заменен соответствующим именем Job в конфигурации Prometheus, а `$fe_leader` должен быть заменен IP-адресом Leader FE.

Алерты приостановки сервиса

Приостановка сервиса FE

PromSQL

```
count(up{group="fe", job="$job_name"}) >= 3
```

Описание алерта

Алерт запускается, когда количество активных узлов FE падает ниже указанного значения. Вы можете настроить это значение на основе фактического количества узлов FE.

Решение

Попытайтесь перезапустить приостановленный узел FE.

Приостановка сервиса BE

PromSQL

```
node_info{type="be_node_num", job="$job_name",state="dead"} > 1
```

Описание алерта

Алерт запускается, когда более одного узла BE приостановлено.

Решение

Попытайтесь перезапустить приостановленный узел BE.

Алерты нагрузки машины

Алерт CPU BE

PromSQL

```
(1-(sum(rate(selena_be_cpu{mode="idle", job="$job_name",instance=~".*"}
[5m])) by (job, instance)) /
(sum(rate(selena_be_cpu{job="$job_name",host=~".*"}[5m])) by (job,
instance))) * 100 > 90
```

Описание алерта

Алерт запускается, когда использование CPU BE превышает 90%.

Решение

Проверьте, есть ли большие запросы или крупномасштабная загрузка данных, и передайте детали команде поддержки для дальнейшего исследования.

1. Используйте команду `top` для проверки использования ресурсов процессами.

   ```
   top -Hp $be_pid
   ```

2. Используйте команду `perf` для сбора и анализа данных производительности.

   ```
   # Выполните команду в течение 1-2 минут и завершите ее, нажав CTRL+C.
   sudo perf top -p $be_pid -g >/tmp/perf.txt
   ```

> **ПРИМЕЧАНИЕ**
>
> В чрезвычайных ситуациях для быстрого восстановления сервиса вы можете попытаться перезапустить соответствующий узел BE после сохранения стека. Чрезвычайная ситуация здесь относится к ситуации, когда использование CPU узла BE остается аномально высоким, и нет эффективных средств для снижения использования CPU.

Алерт памяти

PromSQL

```
(1-
node_memory_MemAvailable_bytes{instance=~".*"}/node_memory_MemTotal_bytes{ins
> 90
```

Описание алерта

Алерт запускается, когда использование памяти превышает 90%.

Решение

Обратитесь к Get Heap Profile для устранения неполадок.

> **ПРИМЕЧАНИЕ**
>
> - В чрезвычайных ситуациях вы можете попытаться перезапустить соответствующий сервис BE для восстановления сервиса. Чрезвычайная ситуация здесь относится к ситуации, когда использование памяти узла BE остается аномально высоким, и нет эффективных средств для снижения использования памяти.
> - Если другие смешанно развернутые сервисы влияют на систему, вы можете рассмотреть возможность завершения этих сервисов в чрезвычайных ситуациях.

Алерты диска

Алерт нагрузки диска

PromSQL

```
rate(node_disk_io_time_seconds_total{instance=~".*"}[1m]) * 100 > 90
```

Описание алерта

Алерт запускается, когда нагрузка диска превышает 90%.

Решение

Если cluster запускает алерт `node_disk_io_time_seconds_total`, сначала проверьте, есть ли какие-либо изменения в бизнесе. Если да, рассмотрите возможность отката

изменений для поддержания предыдущего баланса ресурсов. Если изменения не идентифицированы или откат невозможен, рассмотрите, приводит ли нормальный рост бизнеса к необходимости расширения ресурсов. Вы можете использовать инструмент `iotop` для анализа использования I/O диска. `iotop` имеет UI, похожий на `top`, и включает информацию, такую как `pid`, `user` и I/O.

Вы также можете использовать следующий SQL-запрос для идентификации tablets, потребляющих значительное I/O, и отследить их до конкретных задач и таблиц.

```
-- "all" указывает на все сервисы. 10 указывает, что сбор длится 10 секунд.
3 указывает на получение топ-3 результатов.
ADMIN EXECUTE ON $backend_id
'System.print(ExecEnv.io_profile_and_get_topn_stats("all", 10, 3))';
```

Алерт емкости корневого пути

PromSQL

```
node_filesystem_free_bytes{mountpoint="/"} /1024/1024/1024 < 5
```

Описание алерта

Алерт запускается, когда доступное пространство в корневом каталоге меньше 5 ГБ.

Решение

Общие каталоги, которые могут занимать значительное пространство, включают **/var**, **/**opt и **/tmp**. Используйте следующую команду для проверки больших файлов и очистки ненужных файлов.

```
du -sh / --max-depth=1
```

Алерт емкости диска данных

PromSQL

```
(SUM(selena_be_disks_total_capacity{job="$job"}) by (host, path) -
SUM(selena_be_disks_avail_capacity{job="$job"}) by (host, path)) /
SUM(selena_be_disks_total_capacity{job="$job"}) by (host, path) * 100 > 90
```

Описание алерта

Алерт запускается, когда использование емкости диска превышает 90%.

Решение

1. Проверьте, были ли изменения в объеме загруженных данных.

   Мониторьте метрику `load_bytes` в Grafana. Если произошло значительное увеличение объема загрузки данных, вам может потребоваться масштабировать ресурсы системы.

2. Проверьте наличие операций DROP.

   Если объем загрузки данных не сильно изменился, выполните `SHOW BACKENDS`. Если отчетное использование диска не соответствует фактическому использованию, проверьте FE Audit Log на наличие недавних операций DROP DATABASE, TABLE или PARTITION.

   Метаданные для этих операций остаются в памяти FE в течение одного дня, позволяя вам восстановить данные с помощью инструкции RECOVER в течение 24 часов для избежания ошибочных операций. После восстановления фактическое использование диска может превышать то, что показано в `SHOW BACKENDS`.

   Период хранения удаленных данных в памяти можно настроить с помощью динамического параметра FE `catalog_trash_expire_second` (значение по умолчанию: 86400).

   ```
   ADMIN SET FRONTEND CONFIG ("catalog_trash_expire_second"="86400");
   ```

   Чтобы сохранить это изменение, добавьте элемент конфигурации в конфигурационный файл FE **fe.conf**.

После этого удаленные данные будут перемещены в каталог **trash** на узлах BE (`$storage_root_path/trash`). По умолчанию удаленные данные хранятся в каталоге **trash** в течение одного дня, что также может привести к тому, что фактическое использование диска превысит то, что показано в `SHOW BACKENDS`.

Время хранения удаленных данных в каталоге **trash** можно настроить с помощью динамического параметра BE `trash_file_expire_time_sec` (значение по умолчанию: 86400).

```
curl http://$be_ip:$be_http_port/api/update_config?
trash_file_expire_time_sec=86400
```

Алерт емкости диска метаданных FE

PromSQL

```
node_filesystem_free_bytes{mountpoint="${meta_path}"} /1024/1024/1024 < 10
```

Описание алерта

Алерт запускается, когда доступное дисковое пространство для метаданных FE меньше 10 ГБ.

Решение

Используйте следующие команды для проверки каталогов, занимающих большие объемы пространства, и очистите ненужные файлы. Путь метаданных указывается конфигурацией `meta_dir` в **fe.conf**.

```
du -sh /${meta_dir} --max-depth=1
```

Если каталог метаданных занимает много пространства, это обычно потому, что каталог **bdb** большой, возможно, из-за сбоя CheckPoint. Обратитесь к Алерт сбоя CheckPoint для устранения неполадок. Если этот метод не решит проблему, обратитесь в команду технической поддержки.

Алерты исключений сервиса cluster

Алерты сбоя Compaction

Алерт сбоя Cumulative Compaction

PromSQL

```
increase(selena_be_engine_requests_total{job="$job_name"
,status="failed",type="cumulative_compaction"}[1m]) > 3
increase(selena_be_engine_requests_total{job="$job_name"
,status="failed",type="base_compaction"}[1m]) > 3
```

Описание алерта

Алерт запускается, когда происходят три сбоя в Cumulative Compaction или Base Compaction в течение последней минуты.

Решение

Найдите лог соответствующего узла BE для следующих ключевых слов для идентификации вовлеченного tablet.

```
grep -E 'compaction' be.INFO | grep failed
```

Запись лога, подобная следующей, указывает на сбой Compaction.

```
W0924 17:52:56:537041 123639 comaction_task_cpp:193] compaction task:8482.
tablet:8423674 failed.
```

Вы можете проверить контекст лога для анализа сбоя. Обычно сбой мог быть вызван операцией DROP TABLE или PARTITION во время процесса Compaction. Система имеет внутренний механизм повтора для Compaction, и вы также можете вручную установить статус tablet как BAD и запустить задачу Clone для его восстановления.

> **ПРИМЕЧАНИЕ**
>
> Перед выполнением следующей операции убедитесь, что таблица имеет по крайней мере три полные replicas.

```
ADMIN SET REPLICA STATUS PROPERTIES("tablet_id" = "$tablet_id",
"backend_id" = "$backend_id", "status" = "bad");
```

Алерт высокого давления Compaction

PromSQL

```
selena_fe_max_tablet_compaction_score{job="$job_name",instance="$fe_leader"}
> 100
```

Описание алерта

Алерт запускается, когда наивысший Compaction Score превышает 100, указывая на высокое давление Compaction.

Решение

Этот алерт обычно вызван частой загрузкой, `INSERT INTO VALUES` или операциями `DELETE` (со скоростью 1 в секунду). Рекомендуется установить интервал между задачами загрузки или DELETE более 5 секунд и избегать отправки задач DELETE с высокой concurrency.

Алерт превышения количества версий

PromSQL

```
selena_be_max_tablet_rowset_num{job="$job_name"} > 700
```

Описание алерта

Алерт запускается, когда tablet на узле BE имеет более 700 версий данных.

Решение

Используйте следующую команду для проверки tablet с превышенными версиями:

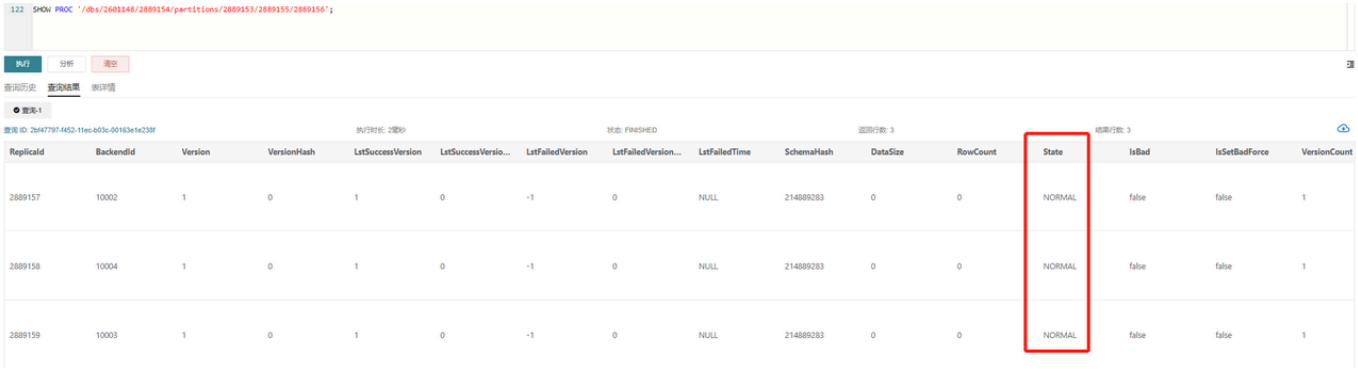
```
SELECT BE_ID,TABLET_ID FROM information_schema.be_tablets WHERE
NUM_ROWSET>700;
```

Пример для Tablet с ID `2889156`:

```
SHOW TABLET 2889156;
```

Выполните команду, возвращенную в поле `DetailCmd`:

```
SHOW PROC '/dbs/2601148/2889154/partitions/2889153/2889155/2889156';
```



122 SHOW PROC '/dbs/2601148/2889154/partitions/2889153/2889155/2889156';

执行 分析 清空

查询历史 查询结果 表详情

查询-1

查询 ID: 2bf47797-f452-11ec-b03c-00163e1e238f 执行时长: 2毫秒 状态: FINISHED 返回行数: 3 结果行数: 3

ReplicaId	BackendId	Version	VersionHash	LstSuccessVersion	LstSuccessVersio...	LstFailedVersion	LstFailedVersion...	LstFailedTime	SchemaHash	DataSize	RowCount	State	IsBad	IsSetBadForce	VersionCount
2889157	10002	1	0	1	0	-1	0	NULL	214889283	0	0	NORMAL	false	false	1
2889158	10004	1	0	1	0	-1	0	NULL	214889283	0	0	NORMAL	false	false	1
2889159	10003	1	0	1	0	-1	0	NULL	214889283	0	0	NORMAL	false	false	1

В нормальных обстоятельствах, как показано, все три replicas должны быть в статусе `NORMAL`, и другие метрики, такие как `RowCount` и `DataSize`, должны оставаться согласованными. Если только одна replica превышает лимит версий 700, вы можете запустить задачу Clone на основе других replicas с помощью следующей команды:

```
ADMIN SET REPLICA STATUS PROPERTIES("tablet_id" = "$tablet_id",
"backend_id" = "$backend_id", "status" = "bad");
```

Если две или более replicas превышают лимит версий, вы можете временно увеличить лимит количества версий:

```
# Замените be_ip на IP узла BE, который хранит tablet, превышающий лимит
версий.
# По умолчанию be_http_port равен 8040.
# Значение по умолчанию tablet_max_versions равно 1000.
curl -XPOST http://$be_ip:$be_http_port/api/update_config?
tablet_max_versions=2000
```

Алерт сбоя CheckPoint

PromSQL

```
selena_fe_meta_log_count{job="$job_name",instance="$fe_master"} > 100000
```

Описание алерта

Алерт запускается, когда количество логов BDB узла FE превышает 100 000. По умолчанию система выполняет CheckPoint, когда количество логов BDB превышает 50 000, а затем сбрасывает счетчик на 0.

Решение

Этот алерт указывает, что CheckPoint не был выполнен. Вам нужно исследовать логи FE для анализа процесса CheckPoint и решения проблемы:

В **fe.log** узла Leader FE найдите записи, такие как `begin to generate new image: image.xxxx`. Если найдено, это означает, что система начала генерировать новый образ. Продолжайте проверять логи на наличие записей, таких как `checkpoint finished save image.xxxx`, чтобы подтвердить успешное создание образа. Если вы найдете `Exception when generate new image file`, генерация образа не удалась. Вам следует тщательно обработать метаданные на основе конкретной ошибки. Рекомендуется обратиться в команду поддержки для дальнейшего анализа.

Алерт превышения количества потоков FE

PromSQL

```
selena_fe_thread_pool{job="$job_name", type!="completed_task_count"} > 3000
```

Описание алерта

Алерт запускается, когда количество потоков на FE превышает 3000.

Решение

Лимит количества потоков по умолчанию для узлов FE и BE составляет 4096. Большое количество запросов UNION ALL обычно приводит к превышению количества потоков. Рекомендуется уменьшить concurrency запросов UNION ALL и настроить системную переменную `pipeline_dop`. Если невозможно настроить гранулярность SQL-запросов, вы можете глобально настроить `pipeline_dop`:

```
SET GLOBAL pipeline_dop=8;
```

> **ПРИМЕЧАНИЕ**
>
> В чрезвычайных ситуациях для быстрого восстановления сервисов вы можете увеличить динамический параметр FE `thrift_server_max_worker_threads` (значение по умолчанию: 4096).
>
> ```
> ADMIN SET FRONTEND CONFIG ("thrift_server_max_worker_threads"="8192");
> ```

Алерт высокого использования FE JVM

PromSQL

```
sum(jvm_heap_size_bytes{job="$job_name", type="used"}) * 100 /
sum(jvm_heap_size_bytes{job="$job_name", type="max"}) > 90
```

Описание алерта

Алерт запускается, когда использование JVM на узле FE превышает 90%.

Решение

Этот алерт указывает, что использование JVM слишком высокое. Вы можете использовать команду `jmap` для анализа ситуации. Поскольку детальная информация мониторинга для

этой метрики все еще находится в разработке, прямые инсайты ограничены. Выполните следующие действия и отправьте результаты команде поддержки для анализа:

```
# Обратите внимание, что указание `live` в команде может привести к перезапуску FE.
jmap -histo[:live] $fe_pid > jmap.dump
```

> **ПРИМЕЧАНИЕ**
>
> В чрезвычайных ситуациях для быстрого восстановления сервисов вы можете перезапустить соответствующий узел FE или увеличить размер JVM (Xmx), а затем перезапустить сервис FE.

Алерты доступности сервиса

Алерты исключений загрузки

Алерт сбоя загрузки

PromSQL

```
rate(selena_fe_txn_failed{job="$job_name",instance="$fe_master"}[5m]) * 100 > 5
```

Описание алерта

Алерт запускается, когда количество неудачных транзакций загрузки превышает 5% от общего числа.

Решение

Проверьте логи узла Leader FE, чтобы найти информацию об ошибках загрузки. Найдите ключевое слово `status: ABORTED` для идентификации неудачных задач загрузки.

```
2024-04-09 18:34:02.363+08:00 INFO (thrift-server-pool-8845163|12111749)
[DatabaseTransactionMgr.abortTransaction():1279] transaction:
[TransactionState. txn_id: 7398864, label: 967009-2f20a55e-368d-48cf-833a-
762cf1fe07c5, db id: 10139, table id list: 155532, callback id: 967009,
coordinator: FE: 192.168.2.1, transaction status: ABORTED, error replicas
num: 0, replica ids: , prepare time: 1712658795053, commit time: -1, finish
time: 1712658842360, total cost: 47307ms, reason: [E1008]Reached
timeout=30000ms @192.168.1.1:8060 attachment: RLTaskTxnCommitAttachment
[filteredRows=0, loadedRows=0, unselectedRows=0, receivedBytes=1033110486,
taskExecutionTimeMs=0, taskId=TUniqueId(hi:3395895943098091727,
lo:-8990743770681178171), jobId=967009, progress=KafkaProgress
[partitionIdToOffset=2_1211970882|7_1211893755]]] successfully rollback
```

Алерт задержки потребления Routine Load

PromSQL

```
(sum by (job_name)
(selena_fe_routine_load_max_lag_of_partition{job="$job_name",instance="$fe_ma
> 300000
selena_fe_routine_load_jobs{job="$job_name",host="$fe_mater",state="NEED_SCHE
> 3
selena_fe_routine_load_jobs{job="$job_name",host="$fe_mater",state="PAUSED"}
selena_fe_routine_load_jobs{job="$job_name",host="$fe_mater",state="UNSTABLE"
```

Описание алерта

- Алерт запускается, когда более 300 000 записей задержаны в потреблении.
- Алерт запускается, когда количество ожидающих задач Routine Load превышает 3.
- Алерт запускается, когда есть задачи в статусе `PAUSED`.
- Алерт запускается, когда есть задачи в статусе `UNSTABLE`.

Решение

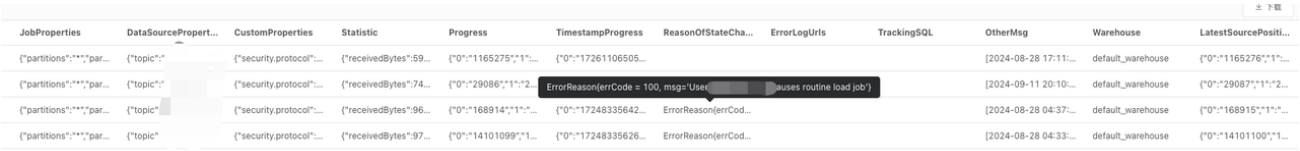
1. Сначала проверьте, является ли статус задачи Routine Load `RUNNING`.

   ```
   SHOW ROUTINE LOAD FROM $db;
   ```

   Обратите внимание на поле `State` в возвращенных данных.

2. Если какая-либо задача Routine Load находится в статусе `PAUSED`, изучите поля `ReasonOfStateChanged`, `ErrorLogUrls` и `TrackingSQL`. Обычно выполнение SQL-запроса в `TrackingSQL` может выявить конкретную ошибку.

   Пример:



JobProperties	DataSourcePropert...	CustomProperties	Statistic	Progress	TimestampProgress	ReasonOfStateCha...	ErrorLogUrls	TrackingSQL	OtherMsg	Warehouse	LatestSourcePositi...
{"partitions":"*","par...	{"topic":"	{"security.protocol":...	{"receivedBytes":59...	{"0":"1165275","1":...	{"0":"17261106505...				[2024-08-28 17:11:...	default_warehouse	{"0":"1165276","1":...
{"partitions":"*","par...	{"topic":"	{"security.protocol":...	{"receivedBytes":74...	{"0":"29086","1":"2...	ErrorReason(errCode = 100, msg='Use auses routine load job')				[2024-09-11 20:10:...	default_warehouse	{"0":"29087","1":"2...
{"partitions":"*","par...	{"topic"	{"security.protocol":...	{"receivedBytes":96...	{"0":"168914","1":"...	{"0":"17248335642...	ErrorReason(errCod...			[2024-08-28 04:37:...	default_warehouse	{"0":"168915","1":"...
{"partitions":"*","par...	{"topic"	{"security.protocol":...	{"receivedBytes":97...	{"0":"14101099","1...	{"0":"17248335626...	ErrorReason(errCod...			[2024-08-28 04:33:...	default_warehouse	{"0":"14101100","1...

3. Если статус задачи Routine Load `RUNNING`, вы можете попытаться увеличить concurrency задачи. Concurrency отдельных задач Routine Load определяется минимальным значением из следующих четырех параметров:

   - `kafka_partition_num`: Количество partitions в Kafka Topic.
   - `desired_concurrent_number`: Установленная concurrency для задачи.
   - `alive_be_num`: Количество активных узлов BE.
   - `max_routine_load_task_concurrent_num`: Параметр конфигурации FE, со значением по умолчанию 5.

В большинстве случаев вам может потребоваться настроить concurrency задачи или количество partitions Kafka Topic (обратитесь в поддержку Kafka при необходимости).

Следующий пример показывает, как установить concurrency для задачи.

```
ALTER ROUTINE LOAD FOR ${routine_load_jobname}
PROPERTIES
(
    "desired_concurrent_number" = "5"
);
```

Алерт лимита транзакций загрузки для одной базы данных

PromSQL

```
sum(selena_fe_txn_running{job="$job_name"}) by(db) > 900
```

Описание алерта

Алерт запускается, когда количество транзакций загрузки для одной базы данных превышает 900 (100 в версиях до v1.5.2).

Решение

Этот алерт обычно запускается большим количеством вновь добавленных задач загрузки. Вы можете временно увеличить лимит на транзакции загрузки для одной базы данных.

```
ADMIN SET FRONTEND CONFIG ("max_running_txn_num_per_db" = "2000");
```

Алерты исключений запросов

Алерт задержки запроса

PromSQL

```
selena_fe_query_latency_ms{job="$job_name", quantile="0.95"} > 5000
```

Описание алерта

Алерт запускается, когда задержка P95 запроса превышает 5 секунд.

Решение

1. Исследуйте, есть ли какие-либо большие запросы.

   Проверьте, потребили ли большие запросы значительные ресурсы машины во время исключения, что привело к тайм-ауту или сбою других запросов.

   - Выполните `show proc '/current_queries';` для просмотра `QueryId` больших запросов. Если вам нужно быстро восстановить сервис, вы можете использовать команду `KILL` для завершения долго выполняющихся запросов.

```
mysql> SHOW PROC '/current_queries';
+--------------------------------------+--------------+------------
+------+-----------+----------------+----------------+-------------
-----+----------+
| QueryId                              | ConnectionId | Database
| User | ScanBytes | ProcessRows    | CPUCostSeconds |
MemoryUsageBytes | ExecTime |
+--------------------------------------+--------------+------------
+------+-----------+----------------+----------------+-------------
-----+----------+
| 7c56495f-ae8b-11ed-8ebf-00163e00accc | 4            | tpcds_100g
| root | 37.88 MB  | 1075769 Rows   | 11.13 Seconds  | 146.70 MB
| 3804     |
| 7d543160-ae8b-11ed-8ebf-00163e00accc | 6            | tpcds_100g
| root | 13.02 GB  | 487873176 Rows | 81.23 Seconds  | 6.37 GB
| 2090     |
+--------------------------------------+--------------+------------
+------+-----------+----------------+----------------+-------------
-----+----------+
2 rows in set (0.01 sec)
```

- Вы также можете перезапустить узлы BE с высоким использованием CPU для решения проблемы.

2. Проверьте, достаточны ли ресурсы машины.

   Проверьте, являются ли нормальными CPU, память, I/O диска и сетевой трафик во время исключения. Если обнаружены аномалии, исследуйте основную причину, изучив вариации пикового трафика и использование ресурсов cluster. Если проблема сохраняется, рассмотрите возможность перезапуска затронутого узла.

ПРИМЕЧАНИЕ

В чрезвычайных ситуациях вы можете решить проблему следующим образом:

- Уменьшение трафика бизнеса и перезапуск затронутого узла BE, если внезапный всплеск трафика вызвал чрезмерное использование ресурсов и сбой запроса.
- Расширение емкости узлов, если высокое использование ресурсов связано с нормальными операциями.

Алерт сбоя запроса

PromSQL

```
sum by (job,instance)(selena_fe_query_err_rate{job="$job_name"}) * 100 > 10

# Этот PromSQL поддерживается с v1.5.2 и далее.
increase(selena_fe_query_internal_err{job="$job_name"})[1m] >10
```

Описание алерта

Алерт запускается, когда частота сбоев запросов превышает 0.1/секунду или 10 неудачных запросов происходят в течение одной минуты.

Решение

Когда этот алерт запускается, проверьте логи для идентификации запросов, которые завершились сбоем.

```
grep 'State=ERR' fe.audit.log
```

Если у вас установлен плагин AuditLoader, вы можете найти соответствующие запросы с помощью следующего запроса.

```
SELECT stmt FROM selena_audit_db__.selena_audit_tbl__ WHERE state='ERR';
```

Обратите внимание, что запросы, которые завершаются сбоем из-за синтаксических ошибок или тайм-аутов, также записываются в `selena_fe_query_err_rate`.

Для сбоев запросов, вызванных проблемами ядра, найдите в `fe.log` ошибку и получите полный стек трассировки и Query Dump, и обратитесь в команду поддержки для устранения неполадок.

Алерт перегрузки запросов

PromSQL

```
abs((sum by (exported_job)
(rate(selena_fe_query_total{process="FE",job="$job_name"}[3m]))-sum by
(exported_job)(rate(selena_fe_query_total{process="FE",job="$job_name"}[3m]
offset 1m)))/sum by (exported_job)
(rate(selena_fe_query_total{process="FE",job="$job_name"}[3m]))) * 100 >
100
abs((sum(selena_fe_connection_total{job="$job_name"})-
sum(selena_fe_connection_total{job="$job_name"} offset
3m))/sum(selena_fe_connection_total{job="$job_name"})) * 100 > 100
```

Описание алерта

Алерт запускается, когда QPS или количество соединений увеличивается на 100% в течение последней минуты.

Решение

Проверьте, являются ли высокочастотные запросы в `fe.audit.log` ожидаемыми. Если есть законные изменения в поведении бизнеса (например, новые сервисы, запущенные в эксплуатацию, или увеличенные объемы данных), мониторьте нагрузку машины и масштабируйте узлы BE по мере необходимости.

Алерт превышения лимита соединений пользователя

PromSQL

```
sum(selena_fe_connection_total{job="$job_name"}) by(user) > 90
```

Описание алерта

Алерт запускается, когда количество соединений пользователя превышает 90. (Лимиты соединений пользователя поддерживаются с версий v1.5.2 и далее.)

Решение

Используйте SQL-команду `SHOW PROCESSLIST` для проверки, соответствует ли количество текущих соединений ожидаемому. Вы можете завершить неожиданные соединения с помощью команды `KILL`. Кроме того, убедитесь, что frontend-сервисы не удерживают соединения открытыми слишком долго, и рассмотрите возможность настройки системной

переменной `wait_timeout` (Единица: Секунды) для ускорения автоматического завершения системой незанятых соединений.

```
SET wait_timeout = 3600;
```

> **ПРИМЕЧАНИЕ**
>
> В чрезвычайных ситуациях вы можете временно увеличить лимит соединений пользователя для восстановления сервиса:
>
> - Для v1.5.2 или позднее:
>
> ```
> ALTER USER 'jack' SET PROPERTIES ("max_user_connections" = "1000");
> ```
>
> - Для v1.5.2 и ранее:
>
> ```
> SET PROPERTY FOR 'jack' 'max_user_connections' = '1000';
> ```

Алерт исключения Schema Change

PromSQL

```
increase(selena_be_engine_requests_total{job="$job_name",type="schema_change"
status="failed"}[1m]) > 1
```

Описание алерта

Алерт запускается, когда более одной задачи Schema Change завершается сбоем в последнюю минуту.

Решение

Выполните следующую инструкцию для проверки, содержит ли поле `Msg` какие-либо сообщения об ошибках:

```
SHOW ALTER COLUMN FROM $db;
```

Если сообщение не найдено, найдите JobId из предыдущего шага в логах Leader FE, чтобы получить контекст.

- Schema Change Out of Memory

  Если Schema Change завершается сбоем из-за недостаточной памяти, найдите в логах **be.WARNING** `failed to process the version`, `failed to process the schema change from tablet` или `Memory of schema change task exceeded limit` для идентификации записей лога, показанных ниже:

  ```
  fail to execute schema change: Memory of schema change task exceed
  limit. DirectSchemaChange Used: 2149621304, Limit: 2147483648. You can
  change the limit by modify BE config
  [memory_limitation_per_thread_for_schema_change]
  ```

  Ошибка лимита памяти обычно вызвана превышением лимита памяти 2 ГБ для одного Schema Change, контролируемого динамическим параметром BE `memory_limitation_per_thread_for_schema_change`. Вы можете изменить этот параметр для решения проблемы.

  ```
  curl -XPOST http://be_host:http_port/api/update_config?
  memory_limitation_per_thread_for_schema_change=8
  ```

- Schema Change Timeout

  За исключением добавления столбцов, которое является легковесной реализацией, большинство Schema Changes включают создание большого количества новых tablets, переписывание исходных данных и реализацию операции через SWAP.

  ```
  Create replicas failed. Error: Error replicas:21539953=99583471,
  21539953=99583467, 21539953=99599851
  ```

  Вы можете решить это следующим образом:

- Увеличение тайм-аута для создания tablets (По умолчанию: 10 секунд).

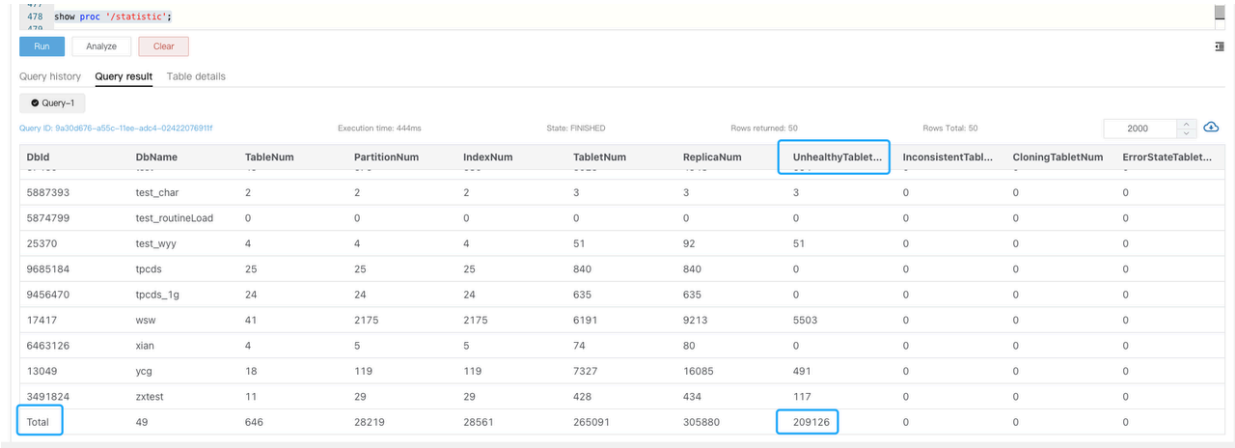
```
ADMIN SET FRONTEND CONFIG ("tablet_create_timeout_second"="60");
```

- Увеличение количества потоков для создания tablets (по умолчанию: 3).

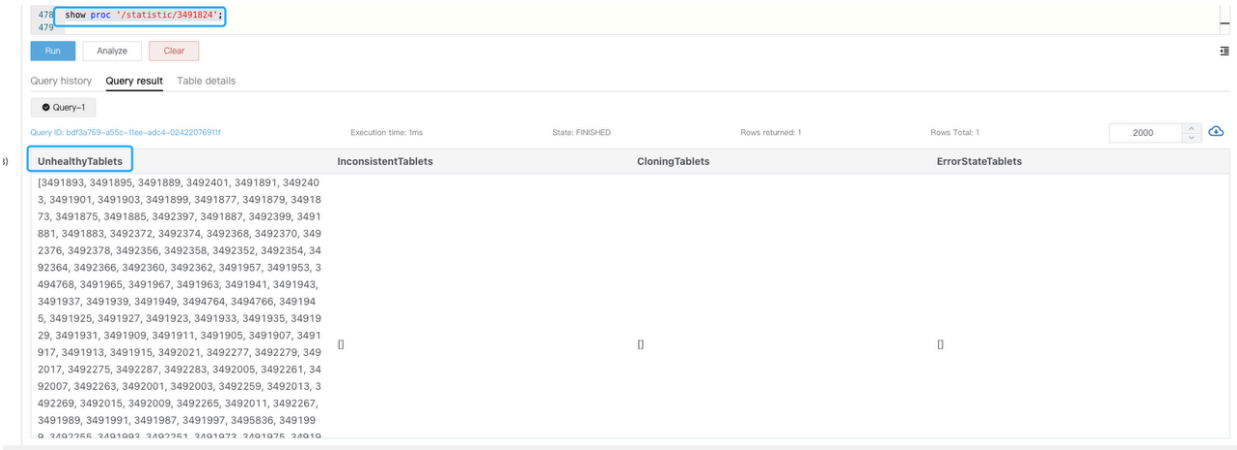
```
curl -XPOST http://be_host:http_port/api/update_config?
alter_tablet_worker_count=6
```

- Ненормальное состояние Tablet

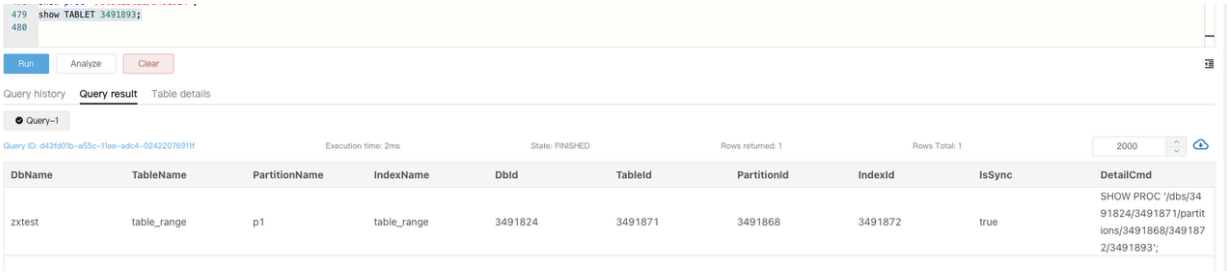
  i. Если tablet находится в ненормальном состоянии, найдите в логах **be.WARNING** `tablet is not normal` и выполните `SHOW PROC '/statistic'` для проверки `UnhealthyTabletNum` на уровне cluster.



  ii. Выполните `SHOW PROC '/statistic/$DbId'` для проверки количества нездоровых tablets в указанной базе данных.



iii. Выполните `SHOW TABLET $tablet_id` для просмотра информации о таблице соответствующего tablet.



iv. Выполните команду, возвращенную в поле `DetailCmd`, для идентификации причины нездоровых tablets.

Обычно нездоровые, а также несогласованные replicas обычно вызваны высокочастотной загрузкой, где прогресс записей в разные replicas не синхронизирован. Вы можете проверить, есть ли у таблицы большое количество записей в реальном времени, и уменьшить количество аномальных replicas, уменьшив частоту загрузки или временно приостановив сервис и повторив задачу после этого.

ПРИМЕЧАНИЕ

В чрезвычайных ситуациях для восстановления сервиса вы можете установить ненормальные replicas как Bad для запуска задачи Clone.

```
ADMIN SET REPLICA STATUS PROPERTIES("tablet_id" = "$tablet_id",
"backend_id" = "$backend_id", "status" = "bad");
```

Перед выполнением этой операции убедитесь, что таблица имеет по крайней мере три полные replicas с только одной ненормальной replica.

Алерт исключения обновления Materialized View

PromSQL

```
increase(selena_fe_mv_refresh_total_failed_jobs[5m]) > 0
```

Описание алерта

Алерт запускается, когда более одного обновления materialized view завершается сбоем в последние пять минут.

Решение

1. Проверьте materialized views, которые завершились сбоем при обновлении.

```
SELECT TABLE_NAME,IS_ACTIVE,INACTIVE_REASON,TASK_NAME FROM
information_schema.materialized_views WHERE LAST_REFRESH_STATE !="
SUCCESS";
```

2. Попытайтесь вручную обновить materialized view.

```
REFRESH MATERIALIZED VIEW $mv_name;
```

3. Если materialized view находится в статусе `INACTIVE`, попытайтесь вручную активировать его.

```
ALTER MATERIALIZED VIEW $mv_name ACTIVE;
```

4. Исследуйте причину сбоя обновления.

```
SELECT * FROM information_schema.task_runs WHERE task_name ='mv-112517'
\G
```

2.2.7 Управление журналами аудита в Selena через AuditLoader

В этом разделе описывается, как управлять журналами аудита Selena в таблице с помощью плагина AuditLoader.

Selena хранит журналы аудита в локальном файле **fe/log/fe.audit.log**, а не во внутренней базе данных. Плагин AuditLoader позволяет управлять журналами аудита непосредственно в вашем cluster. После установки AuditLoader считывает журналы из файла и загружает их в Selena через HTTP PUT. Затем вы можете запрашивать журналы аудита в Selena с помощью SQL-выражений.

Создание таблицы для хранения журналов аудита

Создайте базу данных и таблицу в вашем cluster Selena для хранения журналов аудита. Подробные инструкции см. в разделах CREATE DATABASE и CREATE TABLE.

Поскольку поля журналов аудита различаются в разных версиях Selena, важно следовать приведённым ниже рекомендациям, чтобы избежать проблем совместимости при обновлении:

> **ВНИМАНИЕ**
>
> - Все новые поля должны быть помечены как `NULL`.
> - Поля НЕ должны переименовываться, так как пользователи могут на них полагаться.
> - К типам полей должны применяться только обратно совместимые изменения, например, `VARCHAR(32)` -> `VARCHAR(64)`, чтобы избежать ошибок при вставке.
> - Поля `AuditEvent` разрешаются только по имени. Порядок столбцов в таблице не имеет значения, и пользователь может изменить его в любое время.
> - Поля `AuditEvent`, которые не существуют в таблице, игнорируются, поэтому пользователи могут удалить столбцы, которые им не нужны.

```
CREATE DATABASE selena_audit_db__;

CREATE TABLE selena_audit_db__.selena_audit_tbl__ (
  `queryId` VARCHAR(64) COMMENT "Уникальный идентификатор запроса",
  `timestamp` DATETIME NOT NULL COMMENT "Время начала запроса",
  `queryType` VARCHAR(12) COMMENT "Тип запроса (query, slow_query,
connection)",
  `clientIp` VARCHAR(32) COMMENT "IP-адрес клиента",
  `user` VARCHAR(64) COMMENT "Имя пользователя запроса",
  `authorizedUser` VARCHAR(64) COMMENT "Уникальный идентификатор
пользователя, т.е. user_identity",
  `resourceGroup` VARCHAR(64) COMMENT "Имя группы ресурсов",
  `catalog` VARCHAR(32) COMMENT "Имя каталога",
  `db` VARCHAR(96) COMMENT "База данных, в которой выполняется запрос",
  `state` VARCHAR(8) COMMENT "Состояние запроса (EOF, ERR, OK)",
  `errorCode` VARCHAR(512) COMMENT "Код ошибки",
  `queryTime` BIGINT COMMENT "Время выполнения запроса (миллисекунды)",
  `scanBytes` BIGINT COMMENT "Количество просканированных байт запроса",
  `scanRows` BIGINT COMMENT "Количество просканированных строк запроса",
  `returnRows` BIGINT COMMENT "Количество возвращённых строк запроса",
  `cpuCostNs` BIGINT COMMENT "Время CPU, потреблённое запросом
(наносекунды)",
  `memCostBytes` BIGINT COMMENT "Память, потреблённая запросом (байты)",
  `stmtId` INT COMMENT "Инкрементный идентификатор SQL-выражения",
  `isQuery` TINYINT COMMENT "Является ли SQL запросом (1 или 0)",
  `feIp` VARCHAR(128) COMMENT "IP-адрес FE, выполнившего выражение",
  `stmt` VARCHAR(1048576) COMMENT "Оригинальное SQL-выражение",
  `digest` VARCHAR(32) COMMENT "Отпечаток медленного SQL",
  `planCpuCosts` DOUBLE COMMENT "Использование CPU при планировании запроса
(наносекунды)",
  `planMemCosts` DOUBLE COMMENT "Использование памяти при планировании
запроса (байты)",
  `pendingTimeMs` BIGINT COMMENT "Время ожидания запроса в очереди
(миллисекунды)",
  `candidateMVs` VARCHAR(65533) NULL COMMENT "Список кандидатов
материализованных представлений",
  `hitMvs` VARCHAR(65533) NULL COMMENT "Список совпавших материализованных
представлений",
  `warehouse` VARCHAR(32) NULL COMMENT "Имя хранилища"
) ENGINE = OLAP
DUPLICATE KEY (`queryId`, `timestamp`, `queryType`)
COMMENT "Таблица журналов аудита"
PARTITION BY date_trunc('day', `timestamp`)
PROPERTIES (
  "replication_num" = "1",
```

```
  "partition_live_number" = "30"
);
```

`selena_audit_tbl__` создаётся с динамическими партициями. По умолчанию первая динамическая партиция создаётся через 10 минут после создания таблицы. Затем журналы аудита можно загружать в таблицу. Вы можете проверить партиции в таблице с помощью следующего выражения:

```
SHOW PARTITIONS FROM selena_audit_db__.selena_audit_tbl__;
```

После создания партиции вы можете перейти к следующему шагу.

Загрузка и настройка AuditLoader

1. Загрузите установочный пакет AuditLoader. Пакет совместим со всеми доступными версиями Selena.

2. Распакуйте установочный пакет.

   ```
   unzip auditloader.zip
   ```

   Будут извлечены следующие файлы:

   - **auditloader.jar**: JAR-файл AuditLoader.
   - **plugin.properties**: файл свойств AuditLoader. Вам не нужно изменять этот файл.
   - **plugin.conf**: файл конфигурации AuditLoader. В большинстве случаев вам нужно изменить только поля `user` и `password` в файле.
3. Измените **plugin.conf** для настройки AuditLoader. Вы должны настроить следующие элементы, чтобы AuditLoader работал правильно:

   - `frontend_host_port`: IP-адрес и HTTP-порт FE в формате `<fe_ip>:<fe_http_port>`. Рекомендуется установить значение по умолчанию `127.0.0.1:8030`. Каждый FE в Selena управляет своим журналом аудита независимо, и после установки плагина каждый FE запускает свой фоновый поток для получения и сохранения журналов аудита и записи их через Stream Load. Параметр конфигурации `frontend_host_port` используется для предоставления

IP и порта HTTP-протокола для фоновой задачи Stream Load плагина, и этот параметр не поддерживает несколько значений. IP-часть параметра может использовать IP любого FE в cluster, но это не рекомендуется, потому что если соответствующий FE выйдет из строя, задача записи журнала аудита в фоновом режиме других FE также завершится неудачей из-за сбоя связи. Рекомендуется установить значение по умолчанию `127.0.0.1:8030`, чтобы каждый FE использовал свой собственный HTTP-порт для связи, тем самым избегая влияния на связь в случае исключения других FE (все задачи записи в конечном итоге будут перенаправлены на узел FE Leader для выполнения).

- `database`: имя базы данных, которую вы создали для хранения журналов аудита.
- `table`: имя таблицы, которую вы создали для хранения журналов аудита.
- `user`: ваше имя пользователя cluster. Вы ДОЛЖНЫ иметь привилегию для загрузки данных (LOAD_PRIV) в таблицу.
- `password`: ваш пароль пользователя.
- `secret_key`: ключ (строка, не должна быть длиннее 16 байт), используемый для шифрования пароля. Если этот параметр не установлен, это означает, что пароль в **plugin.conf** не будет зашифрован, и вам нужно указать только открытый текст пароля в `password`. Если этот параметр указан, это означает, что пароль зашифрован этим ключом, и вам нужно указать зашифрованную строку в `password`. Зашифрованный пароль можно сгенерировать в Selena с помощью функции `AES_ENCRYPT`: `SELECT TO_BASE64(AES_ENCRYPT('password','secret_key'));`.
- `filter`: условия фильтра для загрузки журнала аудита. Этот параметр основан на параметре WHERE в Stream Load, т.е. `-H "where: <condition>"`, по умолчанию пустая строка. Пример: `filter=isQuery=1 and clientIp like '127.0.0.1%' and user='root'`.

4. Снова заархивируйте файлы в пакет.

```
zip -q -m -r auditloader.zip auditloader.jar plugin.conf
plugin.properties
```

5. Распространите пакет на все машины, на которых размещены узлы FE. Убедитесь, что все пакеты хранятся по идентичному пути. В противном случае установка завершится неудачей. Не забудьте скопировать абсолютный путь к пакету после распространения.

> **ПРИМЕЧАНИЕ**
>
> Вы также можете распространить **auditloader.zip** на HTTP-сервис, доступный всем FE (например, `httpd` или `nginx`), и установить его через сеть. Обратите внимание, что в обоих случаях **auditloader.zip** должен сохраняться в пути после выполнения установки, и исходные файлы не должны удаляться после установки.

Установка AuditLoader

Выполните следующее выражение вместе с путём, который вы скопировали, чтобы установить AuditLoader в качестве плагина в Selena:

```
INSTALL PLUGIN FROM "<absolute_path_to_package>";
```

Пример установки из локального пакета:

```
INSTALL PLUGIN FROM "<absolute_path_to_package>";
```

Если вы хотите установить плагин через сетевой путь, вам нужно указать md5 пакета в свойствах выражения INSTALL.

Пример:

```
INSTALL PLUGIN FROM "http://xx.xx.xxx.xxx/extra/auditloader.zip"
PROPERTIES("md5sum" = "3975F7B880C9490FE95F42E2B2A28E2D");
```

Подробные инструкции см. в разделе INSTALL PLUGIN.

Проверка установки и запрос журналов аудита

1. Вы можете проверить, была ли установка успешной, с помощью SHOW PLUGINS.

   В следующем примере `Status` плагина `AuditLoader` равен `INSTALLED`, что означает успешную установку.

```
mysql> SHOW PLUGINS\G
*************************** 1. row ***************************
    Name: __builtin_AuditLogBuilder
    Type: AUDIT
Description: builtin audit logger
    Version: 0.12.0
JavaVersion: 1.8.31
ClassName: com.selena.qe.AuditLogBuilder
    SoName: NULL
    Sources: Builtin
    Status: INSTALLED
Properties: {}
*************************** 2. row ***************************
    Name: AuditLoader
    Type: AUDIT
Description: Available for versions 3.3.11+. Load audit log to selena,
and user can view the statistic of queries
    Version: 5.0.0
JavaVersion: 11
ClassName: com.selena.plugin.audit.AuditLoaderPlugin
    SoName: NULL
    Sources: /x/xx/xxx/xxxxx/auditloader.zip
    Status: INSTALLED
Properties: {}
2 rows in set (0.01 sec)
```

2. Выполните несколько случайных SQL для генерации журналов аудита и подождите 60 секунд (или время, которое вы указали в элементе `max_batch_interval_sec` при настройке AuditLoader), чтобы AuditLoader загрузил журналы аудита в Selena.

3. Проверьте журналы аудита, запросив таблицу.

```
SELECT * FROM selena_audit_db__.selena_audit_tbl__;
```

Следующий пример показывает, когда журналы аудита успешно загружены в таблицу:

```
mysql> SELECT * FROM selena_audit_db__.selena_audit_tbl__\G
*************************** 1. row ***************************
       queryId: 01975a33-4129-7520-97a2-05e641cec6c9
     timestamp: 2025-06-10 14:16:37
     queryType: query
      clientIp: xxx.xx.xxx.xx:65283
          user: root
authorizedUser: 'root'@'%'
 resourceGroup: default_wg
       catalog: default_catalog
            db:
         state: EOF
     errorCode:
     queryTime: 3
     scanBytes: 0
      scanRows: 0
    returnRows: 1
     cpuCostNs: 33711
  memCostBytes: 4200
        stmtId: 102
       isQuery: 1
          feIp: xxx.xx.xxx.xx
          stmt: SELECT * FROM selena_audit_db__.selena_audit_tbl__
        digest:
  planCpuCosts: 908
  planMemCosts: 0
 pendingTimeMs: -1
  candidateMvs: null
        hitMVs: null
……————
```

Устранение неполадок

Если журналы аудита не загружаются в таблицу после создания динамической партиции и установки плагина, вы можете проверить, правильно ли настроен **plugin.conf**. Чтобы изменить его, вы должны сначала удалить плагин:

```
UNINSTALL PLUGIN AuditLoader;
```

Журналы AuditLoader выводятся в **fe.log**, вы можете найти их, выполнив поиск по ключевому слову `audit` в **fe.log**. После того как все настройки будут установлены

правильно, вы можете следовать описанным выше шагам, чтобы снова установить AuditLoader.

2.2.8 Включение доступа по FQDN

В этом разделе описывается, как включить доступ к cluster с использованием полного доменного имени (FQDN). FQDN — это **полное доменное имя** для конкретного объекта, к которому можно получить доступ через Интернет. FQDN состоит из двух частей: имени хоста и доменного имени.

До версии 2.4 Selena поддерживала доступ к FE и BE только по IP-адресу. Даже если для добавления узла в cluster использовалось FQDN, оно в конечном итоге преобразовывалось в IP-адрес. Это создавало большие неудобства для администраторов баз данных, поскольку изменение IP-адресов определённых узлов в cluster Selena могло привести к сбоям доступа к узлам. В версии 2.4 Selena отвязала каждый узел от его IP-адреса. Теперь вы можете управлять узлами в Selena исключительно с помощью их FQDN.

Предварительные требования

Чтобы включить доступ по FQDN для cluster Selena, убедитесь, что выполнены следующие требования:

- Каждая машина в cluster должна иметь имя хоста.
- В файле **/etc/hosts** на каждой машине необходимо указать соответствующие IP-адреса и FQDN других машин в cluster.
- IP-адреса в файле **/etc/hosts** должны быть уникальными.

Настройка нового cluster с доступом по FQDN

По умолчанию узлы FE в новом cluster запускаются с доступом по IP-адресу. Чтобы запустить новый cluster с доступом по FQDN, необходимо запустить узлы FE, выполнив следующие команды **при первом запуске cluster**:

```
./bin/start_fe.sh --host_type FQDN --daemon
```

Свойство `--host_type` указывает метод доступа, используемый для запуска узла. Допустимые значения включают `FQDN` и `IP`. Вам нужно указать это свойство только ОДИН РАЗ при первом запуске узла.

Каждый узел BE идентифицирует себя с помощью `BE Address`, определённого в метаданных FE. Поэтому вам НЕ нужно указывать `--host_type` при запуске узлов BE. Если `BE Address` определяет узел BE с FQDN, узел BE идентифицирует себя с этим FQDN.

Включение доступа по FQDN в существующем cluster

Чтобы включить доступ по FQDN в существующем cluster, который ранее был запущен с IP-адресами, сначала необходимо **обновить** Selena до версии 2.4.0 или более поздней.

Включение доступа по FQDN для узлов FE

Необходимо включить доступ по FQDN для всех non-Leader Follower узлов FE перед включением его для Leader FE узла.

> **ВНИМАНИЕ**
>
> Убедитесь, что в cluster есть как минимум три Follower FE узла, прежде чем включать доступ по FQDN для узлов FE.

Включение доступа по FQDN для non-Leader Follower FE узлов

1. Перейдите в каталог развёртывания узла FE и выполните следующую команду для остановки узла FE:

```
./bin/stop_fe.sh
```

2. Выполните следующий оператор через MySQL-клиент, чтобы проверить статус `Alive` остановленного узла FE. Подождите, пока статус `Alive` не станет `false`.

```
SHOW PROC '/frontends'\G
```

3. Выполните следующий оператор для замены IP-адреса на FQDN.

```
ALTER SYSTEM MODIFY FRONTEND HOST "<fe_ip>" TO "<fe_hostname>";
```

4. Выполните следующую команду для запуска узла FE с доступом по FQDN.

```
./bin/start_fe.sh --host_type FQDN --daemon
```

   Свойство `--host_type` указывает метод доступа, используемый для запуска узла. Допустимые значения включают `FQDN` и `IP`. Вам нужно указать это свойство только ОДИН РАЗ при перезапуске узла после его модификации.

5. Проверьте статус `Alive` узла FE. Подождите, пока статус `Alive` не станет `true`.

```
SHOW PROC '/frontends'\G
```

6. Повторите вышеуказанные шаги для включения доступа по FQDN для других non-Leader Follower FE узлов один за другим, когда статус `Alive` текущего узла FE станет `true`.

Включение доступа по FQDN для Leader FE узла

После того как все non-Leader FE узлы были успешно изменены и перезапущены, вы можете включить доступ по FQDN для Leader FE узла.

> **ПРИМЕЧАНИЕ**
>
> До тех пор пока Leader FE узел не будет включён с доступом по FQDN, FQDN, используемые для добавления узлов в cluster, по-прежнему преобразуются в соответствующие IP-адреса. После того как для cluster будет избран Leader FE узел с включённым доступом по FQDN, FQDN не будут преобразовываться в IP-адреса.

1. Перейдите в каталог развёртывания Leader FE узла и выполните следующую команду для остановки Leader FE узла.

```
./bin/stop_fe.sh
```

2. Выполните следующий оператор через MySQL-клиент, чтобы проверить, был ли избран новый Leader FE узел для cluster.

```
SHOW PROC '/frontends'\G
```

   Любой узел FE со статусом `Alive` и `isMaster` равным `true` является работающим Leader FE.

3. Выполните следующий оператор для замены IP-адреса на FQDN.

```
ALTER SYSTEM MODIFY FRONTEND HOST "<fe_ip>" TO "<fe_hostname>";
```

4. Выполните следующую команду для запуска узла FE с доступом по FQDN.

```
./bin/start_fe.sh --host_type FQDN --daemon
```

   Свойство `--host_type` указывает метод доступа, используемый для запуска узла. Допустимые значения включают `FQDN` и `IP`. Вам нужно указать это свойство только ОДИН РАЗ при перезапуске узла после его модификации.

5. Проверьте статус `Alive` узла FE.

```
SHOW PROC '/frontends'\G
```

Если статус `Alive` становится `true`, узел FE успешно изменён и добавлен в cluster как Follower FE узел.

Включение доступа по FQDN для узлов BE

Выполните следующий оператор через MySQL-клиент для замены IP-адреса на FQDN, чтобы включить доступ по FQDN для узла BE.

```
ALTER SYSTEM MODIFY BACKEND HOST "<be_ip>" TO "<be_hostname>";
```

> **ПРИМЕЧАНИЕ**

> Вам НЕ нужно перезапускать узел BE после включения доступа по FQDN.

Откат

Чтобы откатить cluster Selena с включённым доступом по FQDN до более ранней версии, не поддерживающей доступ по FQDN, сначала необходимо включить доступ по IP-адресу для всех узлов в cluster. Вы можете обратиться к разделу Включение доступа по FQDN в существующем cluster для общего руководства, за исключением того, что вам нужно изменить SQL-команды на следующие:

- Включение доступа по IP-адресу для узла FE:

```
ALTER SYSTEM MODIFY FRONTEND HOST "<fe_hostname>" TO "<fe_ip>";
```

- Включение доступа по IP-адресу для узла BE:

```
ALTER SYSTEM MODIFY BACKEND HOST "<be_hostname>" TO "<be_ip>";
```

Изменения вступят в силу после успешного перезапуска вашего cluster.

FAQ

В: При включении доступа по FQDN для узла FE возникает ошибка: "required 1 replica. But none were active with this master". Что делать?

О: Убедитесь, что в cluster есть как минимум три Follower FE узла, прежде чем включать доступ по FQDN для узлов FE.

В: Могу ли я добавить новый узел по IP-адресу в cluster с включённым доступом по FQDN?

О: Да.

2.2.9 Настройка часового пояса

В этой теме описывается, как настроить часовой пояс и как настройки часового пояса влияют на работу системы.

Настройка часового пояса на уровне сессии или глобального часового пояса

Вы можете настроить часовой пояс на уровне сессии или глобальный часовой пояс для вашего cluster Selena с помощью параметра `time_zone`.

- Для настройки часового пояса на уровне сессии выполните команду `SET time_zone = 'xxx';`. Вы можете настроить разные часовые пояса для разных сессий. Настройка часового пояса становится недействительной при отключении от FE.
- Для настройки глобального часового пояса выполните команду `SET global time_zone = 'xxx';`. Настройка часового пояса сохраняется в FE и остаётся действительной даже после отключения от FE.

> **Примечание**
>
> Перед загрузкой данных в Selena измените глобальный часовой пояс вашего cluster Selena на то же значение, что и параметр `system_time_zone`. В противном случае после загрузки данных данные типа DATE будут некорректными. Параметр `system_time_zone` относится к часовому поясу машин, на которых размещены FE. При запуске машин часовой пояс машин записывается как значение этого параметра. Вы не можете вручную настроить этот параметр.

Формат часового пояса

Значение параметра `time_zone` не чувствительно к регистру. Значение параметра может быть в одном из следующих форматов.

Формат	Пример
Смещение UTC	`SET time_zone = '+10:00';` `SET global time_zone = '-6:00';`
Название часового пояса	`SET time_zone = 'Asia/Shanghai';` `SET global time_zone = 'America/Los_Angeles';`

Для получения дополнительной информации о формате часового пояса см. Список часовых поясов базы данных tz.

> **Примечание**
>
> Сокращения часовых поясов не поддерживаются, за исключением CST. Если вы установите значение `time_zone` в `CST`, Selena преобразует `CST` в `Asia/Shanghai`.

Часовой пояс по умолчанию

Значение по умолчанию параметра `time_zone` - `Asia/Shanghai`.

Просмотр настроек часового пояса

Чтобы просмотреть настройку часового пояса, выполните следующую команду.

```
SHOW VARIABLES LIKE '%time_zone%';
```

Влияние настроек часового пояса

- Настройки часового пояса влияют на значения времени, возвращаемые операторами SHOW LOAD и SHOW BACKENDS. Однако настройки не влияют на значение, указанное в предложении `LESS THAN`, когда столбцы партиционирования, указанные в операторе CREATE TABLE, имеют тип DATE или DATETIME. Настройки также не влияют на данные типов DATE и DATETIME.
- Настройки часового пояса влияют на отображение и хранение следующих функций:
  - **from_unixtime**: возвращает дату и время вашего указанного часового пояса на основе указанной временной метки UTC. Например, если глобальный часовой пояс вашего cluster Selena - `Asia/Shanghai`, `select FROM_UNIXTIME(0);` вернёт `1970-01-01 08:00:00`.
  - **unix_timestamp**: возвращает временную метку UTC на основе даты и времени вашего указанного часового пояса. Например, если глобальный часовой пояс вашего cluster Selena - `Asia/Shanghai`, `select UNIX_TIMESTAMP('1970-01-01 08:00:00');` вернёт `0`.
  - **curtime**: возвращает текущее время вашего указанного часового пояса. Например, если текущее время указанного часового пояса - 16:34:05, `select CURTIME();` вернёт `16:34:05`.
  - **now**: возвращает текущую дату и время вашего указанного часового пояса. Например, если текущая дата и время указанного часового пояса - 2021-02-11 16:34:13, `select NOW();` вернёт `2021-02-11 16:34:13`.
  - **convert_tz**: преобразует дату и время из одного часового пояса в другой. Например, `select CONVERT_TZ('2021-08-01 11:11:11', 'Asia/Shanghai', 'America/Los_Angeles');` вернёт `2021-07-31 20:11:11`.

2.2.10 Мониторинг и управление большими запросами

В этом разделе описывается, как мониторить и управлять большими запросами в вашем cluster Selena.

Большие запросы включают запросы, которые сканируют слишком много строк или занимают слишком много ресурсов CPU и памяти. Они могут легко исчерпать ресурсы cluster и вызвать перегрузку системы, если не наложить на них ограничения. Для решения этой проблемы Selena предоставляет ряд мер для мониторинга и управления большими запросами, предотвращая монополизацию ресурсов cluster запросами.

Общая идея обработки больших запросов в Selena следующая:

1. Настроить автоматические меры предосторожности против больших запросов с помощью групп ресурсов и очередей запросов.
2. Мониторить большие запросы в реальном времени и завершать те, которые обходят меры предосторожности.
3. Анализировать журналы аудита и журналы больших запросов для изучения паттернов больших запросов и тонкой настройки механизмов предосторожности, настроенных ранее.

Эта функция поддерживается с версии v1.5.2.

Настройка мер предосторожности против больших запросов

Selena предоставляет два инструмента предосторожности для работы с большими запросами — группы ресурсов и очереди запросов. Вы можете использовать группы ресурсов для остановки выполнения больших запросов. Очереди запросов, с другой стороны, могут помочь вам ставить входящие запросы в очередь, когда достигнут порог параллелизма или лимит ресурсов, предотвращая перегрузку системы.

Фильтрация больших запросов через группы ресурсов

Группы ресурсов могут автоматически идентифицировать и завершать большие запросы. При создании группы ресурсов вы можете указать верхний предел времени CPU, использования памяти или количества просканированных строк, на которые запрос имеет право. Среди всех запросов, попадающих в группу ресурсов, любые запросы, требующие больше ресурсов, отклоняются и возвращаются с ошибкой. Дополнительную информацию и инструкции по группам ресурсов см. в разделе Изоляция ресурсов.

Перед созданием групп ресурсов необходимо выполнить следующее выражение для включения Pipeline Engine, от которого зависит функция группы ресурсов:

```
SET GLOBAL enable_pipeline_engine = true;
```

Следующий пример создаёт группу ресурсов `bigQuery`, которая ограничивает верхний предел времени CPU до `100` секунд, верхний предел количества просканированных строк до `100000` и верхний предел использования памяти до `1073741824` байт (1 ГБ):

```
CREATE RESOURCE GROUP bigQuery
TO
    (db='sr_hub')
WITH (
    'cpu_weight' = '10',
    'mem_limit' = '20%',
    'big_query_cpu_second_limit' = '100',
    'big_query_scan_rows_limit' = '100000',
    'big_query_mem_limit' = '1073741824'
);
```

Если требуемые ресурсы запроса превышают любой из лимитов, запрос не будет выполнен и вернётся с ошибкой. Следующий пример показывает сообщение об ошибке, возвращаемое, когда запрос требует слишком много просканированных строк:

```
ERROR 1064 (HY000): exceed big query scan_rows limit: current is 4 but
limit is 1
```

Если вы впервые настраиваете группы ресурсов, рекомендуется установить относительно более высокие лимиты, чтобы они не препятствовали обычным запросам. Вы можете тонко настроить эти лимиты после того, как лучше узнаете паттерны больших запросов.

Смягчение перегрузки системы через очереди запросов

Очереди запросов предназначены для смягчения ухудшения перегрузки системы, когда занятость ресурсов cluster превышает заранее определённые пороги. Вы можете установить пороги для максимального параллелизма, использования памяти и использования CPU. Selena автоматически ставит входящие запросы в очередь, когда достигнут любой из этих порогов. Ожидающие запросы либо ждут в очереди выполнения,

либо отменяются, когда достигнут заранее определённый порог ресурсов. Дополнительную информацию см. в разделе Очереди запросов.

Выполните следующие выражения для включения очередей запросов для SELECT-запросов:

```
SET GLOBAL enable_query_queue_select = true;
```

После включения функции очереди запросов вы можете определить правила для запуска очередей запросов.

- Укажите порог параллелизма для запуска очереди запросов.

  Следующий пример устанавливает порог параллелизма на `100`:

  ```
  SET GLOBAL query_queue_concurrency_limit = 100;
  ```

- Укажите порог коэффициента использования памяти для запуска очереди запросов.

  Следующий пример устанавливает порог коэффициента использования памяти на `0.9`:

  ```
  SET GLOBAL query_queue_mem_used_pct_limit = 0.9;
  ```

- Укажите порог коэффициента использования CPU для запуска очереди запросов.

  Следующий пример устанавливает порог промилле использования CPU (использование CPU * 1000) на `800`:

  ```
  SET GLOBAL query_queue_cpu_used_permille_limit = 800;
  ```

Вы также можете решить, как обрабатывать эти запросы в очереди, настроив максимальную длину очереди и тайм-аут для каждого ожидающего запроса в очереди.

- Укажите максимальную длину очереди запросов. При достижении этого порога входящие запросы отклоняются.

Следующий пример устанавливает длину очереди запросов на `100`:

```
SET GLOBAL query_queue_max_queued_queries = 100;
```

- Укажите максимальный тайм-аут ожидающего запроса в очереди. При достижении этого порога соответствующий запрос отклоняется.

  Следующий пример устанавливает максимальный тайм-аут на `480` секунд:

  ```
  SET GLOBAL query_queue_pending_timeout_second = 480;
  ```

Вы можете проверить, находится ли запрос в ожидании, с помощью SHOW PROCESSLIST.

```
mysql> SHOW PROCESSLIST;
+------+------+---------------------+-------+---------+--------------------
-+------+-------+-------------------+-----------+
| Id   | User | Host                | Db    | Command | ConnectionStartTime
| Time | State | Info              | IsPending |
+------+------+---------------------+-------+---------+--------------------
-+------+-------+-------------------+-----------+
|    2 | root | xxx.xx.xxx.xx:xxxxx |       | Query   | 2022-11-24 18:08:29
|    0 | OK    | SHOW PROCESSLIST  | false     |
+------+------+---------------------+-------+---------+--------------------
-+------+-------+-------------------+-----------+
```

Если `IsPending` равно `true`, соответствующий запрос ожидает в очереди запросов.

Мониторинг больших запросов в реальном времени

Начиная с версии v1.5.2, Selena поддерживает просмотр запросов, которые в данный момент обрабатываются в cluster, и ресурсов, которые они занимают. Это позволяет мониторить cluster на случай, если какие-либо большие запросы обходят меры предосторожности и вызывают неожиданную перегрузку системы.

Мониторинг через клиент MySQL

1. Вы можете просмотреть запросы, которые в данный момент обрабатываются (`current_queries`), с помощью SHOW PROC.

```
SHOW PROC '/current_queries';
```

Selena возвращает ID запроса (`QueryId`), ID соединения (`ConnectionId`) и потребление ресурсов каждого запроса, включая размер просканированных данных (`ScanBytes`), количество обработанных строк (`ProcessRows`), время CPU (`CPUCostSeconds`), использование памяти (`MemoryUsageBytes`) и время выполнения (`ExecTime`).

```
mysql> SHOW PROC '/current_queries';
+--------------------------------------+--------------+------------+---
---+-----------+----------------+----------------+------------------+--
--------+
| QueryId                              | ConnectionId | Database   |
User | ScanBytes | ProcessRows    | CPUCostSeconds | MemoryUsageBytes |
ExecTime |
+--------------------------------------+--------------+------------+---
---+-----------+----------------+----------------+------------------+--
--------+
| 7c56495f-ae8b-11ed-8ebf-00163e00accc | 4            | tpcds_100g |
root | 37.88 MB  | 1075769 Rows   | 11.13 Seconds  | 146.70 MB        |
3804     |
| 7d543160-ae8b-11ed-8ebf-00163e00accc | 6            | tpcds_100g |
root | 13.02 GB  | 487873176 Rows | 81.23 Seconds  | 6.37 GB          |
2090     |
+--------------------------------------+--------------+------------+---
---+-----------+----------------+----------------+------------------+--
--------+
2 rows in set (0.01 sec)
```

2. Вы можете дополнительно изучить потребление ресурсов запроса на каждом узле BE, указав ID запроса.

```
SHOW PROC '/current_queries/<QueryId>/hosts';
```

Selena возвращает размер просканированных данных запроса (`ScanBytes`), количество просканированных строк (`ScanRows`), время CPU (`CPUCostSeconds`) и использование памяти (`MemUsageBytes`) на каждом узле BE.

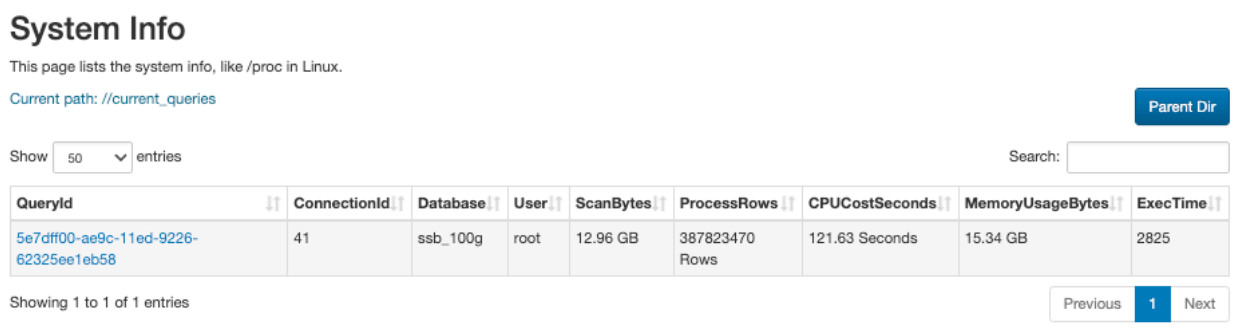
```
mysql> show proc '/current_queries/7c56495f-ae8b-11ed-8ebf-
00163e00accc/hosts';
+--------------------+-----------+-------------+----------------+------
---------+
| Host               | ScanBytes | ScanRows    | CpuCostSeconds |
MemUsageBytes |
+--------------------+-----------+-------------+----------------+------
---------+
| 172.26.34.185:8060 | 11.61 MB  | 356252 Rows | 52.93 Seconds  | 51.14
MB      |
| 172.26.34.186:8060 | 14.66 MB  | 362646 Rows | 52.89 Seconds  | 50.44
MB      |
| 172.26.34.187:8060 | 11.60 MB  | 356871 Rows | 52.91 Seconds  | 48.95
MB      |
+--------------------+-----------+-------------+----------------+------
---------+
3 rows in set (0.00 sec)
```

Мониторинг через консоль FE

Помимо клиента MySQL, вы можете использовать консоль FE для визуализированного интерактивного мониторинга.

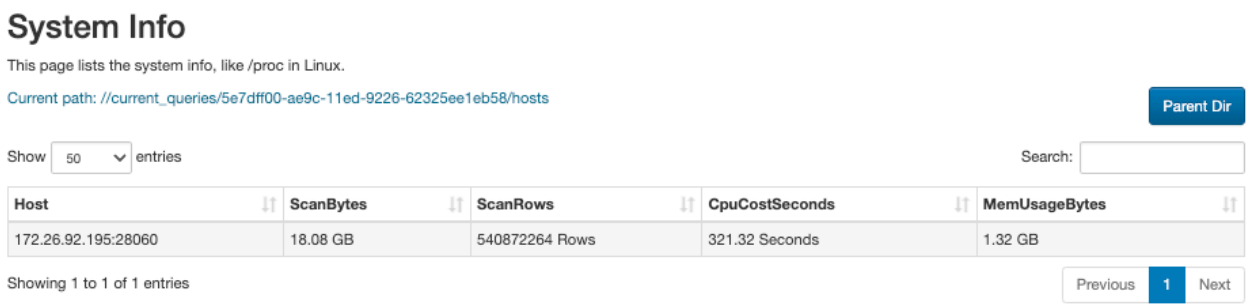
1. Перейдите в консоль FE в браузере, используя следующий URL:

```
http://<fe_IP>:<fe_http_port>/system?path=//current_queries
```



   На странице **System Info** вы можете просмотреть запросы, которые в данный момент обрабатываются, и их потребление ресурсов.

2. Нажмите на **QueryID** запроса.



На появившейся странице вы можете просмотреть подробную информацию о потреблении ресурсов по узлам.

Ручное завершение больших запросов

Если какие-либо большие запросы обходят установленные вами меры предосторожности и угрожают доступности системы, вы можете завершить их вручную, используя соответствующий ID соединения в выражении KILL:

```
KILL QUERY <ConnectionId>;
```

Анализ журналов больших запросов

Начиная с версии v1.5.2, Selena поддерживает журналы больших запросов, которые хранятся в файле **fe/log/fe.big_query.log**. По сравнению с журналами аудита Selena, журналы больших запросов выводят три дополнительных поля:

- `bigQueryLogCPUSecondThreshold`
- `bigQueryLogScanBytesThreshold`
- `bigQueryLogScanRowsThreshold`

Эти три поля соответствуют порогам потребления ресурсов, которые вы определили, чтобы определить, является ли запрос большим.

Чтобы включить журналы больших запросов, выполните следующее выражение:

```
SET GLOBAL enable_big_query_log = true;
```

После включения журналов больших запросов вы можете определить правила для их запуска.

- Укажите порог времени CPU для запуска журналов больших запросов.

  Следующий пример устанавливает порог времени CPU на `600` секунд:

  ```
  SET GLOBAL big_query_log_cpu_second_threshold = 600;
  ```

- Укажите порог размера сканируемых данных для запуска журналов больших запросов.

  Следующий пример устанавливает порог размера сканируемых данных на `10737418240` байт (10 ГБ):

  ```
  SET GLOBAL big_query_log_scan_bytes_threshold = 10737418240;
  ```

- Укажите порог количества просканированных строк для запуска журналов больших запросов.

  Следующий пример устанавливает порог количества просканированных строк на `1500000000`:

  ```
  SET GLOBAL big_query_log_scan_rows_threshold = 1500000000;
  ```

Тонкая настройка мер предосторожности

Из статистики, полученной при мониторинге в реальном времени и из журналов больших запросов, вы можете изучить паттерны пропущенных больших запросов (или обычных запросов, ошибочно диагностированных как большие) в вашем cluster, а затем оптимизировать настройки для групп ресурсов и очереди запросов.

Если значительная часть больших запросов соответствует определённому паттерну SQL и вы хотите навсегда запретить этот паттерн SQL, вы можете добавить этот паттерн в чёрный список SQL. Selena отклоняет все запросы, соответствующие любым паттернам, указанным в чёрном списке SQL, и возвращает ошибку. Дополнительную информацию см. в разделе Управление чёрным списком SQL.

Чтобы включить чёрный список SQL, выполните следующее выражение:

```
ADMIN SET FRONTEND CONFIG ("enable_sql_blacklist" = "true");
```

Затем вы можете добавить регулярное выражение, представляющее паттерн SQL, в чёрный список SQL с помощью ADD SQLBLACKLIST.

Следующий пример добавляет `COUNT(DISTINCT)` в чёрный список SQL:

```
ADD SQLBLACKLIST "SELECT COUNT(DISTINCT .+) FROM .+";
```

2.2.11 Compaction для cluster с общими данными

В этом разделе описывается, как управлять compaction в cluster Selena с общими данными.

Обзор

Каждая операция загрузки данных в Selena генерирует новую версию файлов данных. Compaction объединяет файлы данных из разных версий в более крупные файлы, уменьшая количество мелких файлов и повышая эффективность запросов.

Показатель Compaction Score

Обзор

Compaction Score отражает состояние слияния файлов данных в партиции. Более высокий показатель указывает на меньший прогресс слияния, что означает, что партиция имеет больше несмерженных версий файлов данных. FE поддерживает информацию о Compaction Score для каждой партиции, включая Max Compaction Score (наивысший показатель среди всех tablet в партиции).

Если Max Compaction Score партиции ниже параметра FE `lake_compaction_score_selector_min_score` (по умолчанию: 10), compaction для этой партиции считается завершённым. Max Compaction Score, превышающий 100, указывает на нездоровое состояние compaction. Когда показатель превышает параметр FE `lake_ingest_slowdown_threshold` (по умолчанию: 100), система замедляет фиксацию транзакций загрузки данных для партиции. Если он превышает `lake_compaction_score_upper_bound` (по умолчанию: 2000), система отклоняет транзакции импорта для партиции.

Правила расчёта

Обычно каждый файл данных добавляет 1 к Compaction Score. Например, если партиция имеет один tablet и 10 файлов данных, сгенерированных при первой операции загрузки, Max Compaction Score партиции равен 10. Все файлы данных, сгенерированные транзакцией в пределах tablet, группируются как Rowset.

При расчёте показателя Rowset tablet группируются по размеру, и группа с наибольшим количеством файлов определяет Compaction Score tablet.

Например, tablet проходит 7 операций загрузки, генерируя Rowset с размерами: 100 МБ, 100 МБ, 100 МБ, 10 МБ, 10 МБ, 10 МБ и 10 МБ. При расчёте система объединит три Rowset по 100 МБ в одну группу и четыре Rowset по 10 МБ в другую. Compaction Score рассчитывается на основе группы с большим количеством файлов. В этом случае вторая группа имеет больший compaction score. Compaction приоритизирует группу с более высоким показателем, поэтому после первого compaction распределение Rowset будет: 100 МБ, 100 МБ, 100 МБ и 40 МБ.

Рабочий процесс Compaction

Для cluster с общими данными Selena представляет новый механизм compaction, управляемый FE:

1. Расчёт показателя: Leader FE рассчитывает и сохраняет Compaction Score для партиций на основе результатов публикации транзакций.
2. Выбор кандидатов: FE выбирает партиции с наивысшими Max Compaction Score в качестве кандидатов на compaction.
3. Генерация задач: FE инициирует транзакции compaction для выбранных партиций, генерирует подзадачи уровня tablet и отправляет их на Compute Node (CN) до достижения лимита, установленного параметром FE `lake_compaction_max_tasks`.
4. Выполнение подзадач: CN выполняют подзадачи compaction в фоновом режиме. Количество одновременных подзадач на CN контролируется параметром CN `compact_threads`.
5. Сбор результатов: FE агрегирует результаты подзадач и фиксирует транзакцию compaction.
6. Публикация: FE публикует успешно зафиксированную транзакцию compaction.

Управление compaction

Просмотр показателей compaction

- Вы можете просмотреть показатели compaction партиций в определённой таблице с помощью выражения SHOW PROC. Обычно вам нужно обращать внимание только на поле `MaxCS`. Если `MaxCS` ниже 10, compaction считается завершённым. Если `MaxCS` выше 100, Compaction Score относительно высок. Если `MaxCS` превышает 500, Compaction Score очень высок и может потребоваться ручное вмешательство.

```
SHOW PARTITIONS FROM <table_name>
SHOW PROC '/dbs/<database_name>/<table_name>/partitions'
```

Пример:

```
mysql> SHOW PROC '/dbs/load_benchmark/store_sales/partitions';
+-------------+---------------+----------------+----------------+------
-------+--------+--------------+-------+-----------------------------
+---------+----------+-----------+----------+------------+-------+-----
--+-------+
| PartitionId | PartitionName | CompactVersion | VisibleVersion |
NextVersion | State  | PartitionKey | Range | DistributionKey
| Buckets | DataSize | RowCount  | CacheTTL | AsyncWrite | AvgCS |
P50CS | MaxCS |
+-------------+---------------+----------------+----------------+------
-------+--------+--------------+-------+-----------------------------
+---------+----------+-----------+----------+------------+-------+-----
--+-------+
| 38028       | store_sales   | 913            | 921            | 923
| NORMAL |               |       | ss_item_sk, ss_ticket_number | 64
| 15.6GB   | 273857126 | 2592000  | false      | 10.00 | 10.00 | 10.00
|
+-------------+---------------+----------------+----------------+------
-------+--------+--------------+-------+-----------------------------
+---------+----------+-----------+----------+------------+-------+-----
--+-------+
1 row in set (0.20 sec)
```

- Вы также можете просмотреть показатели compaction партиций, запросив системное представление `information_schema.partitions_meta`.

Пример:

```
mysql> SELECT * FROM information_schema.partitions_meta ORDER BY Max_CS
LIMIT 10;
+--------------+---------------------------+-------------------------
--+--------------+-----------------+-----------------+-----------------
-----+--------------+---------------+-----------------+----------------
-------------------------+---------+-----------------+----------------
+---------------------+-----------------------------+--------------+---
------+-----------+------------+------------------+----------+--------
+--------+-----------------------------------------------------------------
------+
| DB_NAME      | TABLE_NAME                 | PARTITION_NAME
| PARTITION_ID | COMPACT_VERSION | VISIBLE_VERSION |
VISIBLE_VERSION_TIME | NEXT_VERSION | PARTITION_KEY | PARTITION_VALUE |
DISTRIBUTION_KEY                        | BUCKETS | REPLICATION_NUM |
STORAGE_MEDIUM | COOLDOWN_TIME       | LAST_CONSISTENCY_CHECK_TIME |
IS_IN_MEMORY | IS_TEMP | DATA_SIZE | ROW_COUNT  | ENABLE_DATACACHE |
AVG_CS   | P50_CS | MAX_CS | STORAGE_PATH
|
+--------------+---------------------------+-------------------------
--+--------------+-----------------+-----------------+-----------------
-----+--------------+---------------+-----------------+----------------
-------------------------+---------+-----------------+----------------
+---------------------+-----------------------------+--------------+---
------+-----------+------------+------------------+----------+--------
+--------+-----------------------------------------------------------------
------+
| tpcds_1t     | call_center                | call_center
|        11905 |               0 |               2 | 2024-03-17
08:30:47  |            3 |               |                 |
cc_call_center_sk                       |       1 |               1 |
HDD            | 9999-12-31 23:59:59 | NULL                        |
0 |       0 | 12.3KB    |         42 |                0 |        0 |
0 |      0 | s3://XXX/536a3c77-52c3-485a-8217-
781734a970b1/db10328/11906/11905 |
...
```

Просмотр задач compaction

По мере загрузки новых данных в систему FE постоянно планирует задачи compaction для выполнения на различных узлах CN. Сначала вы можете просмотреть общий статус задач compaction на FE, а затем просмотреть детали выполнения каждой задачи на CN.

Просмотр общего статуса задач compaction

Вы можете просмотреть общий статус задач compaction с помощью выражения SHOW PROC.

```
SHOW PROC '/compactions';
```

Пример:

```
mysql> SHOW PROC '/compactions';
+---------------------+-------+---------------------+---------------------+--
-----------------------------------------------------------------------------
| Partition           | TxnID | StartTime           | CommitTime          | F
|
+---------------------+-------+---------------------+---------------------+--
-----------------------------------------------------------------------------
| ssb.lineorder.10081 | 15    | 2026-01-10 03:29:07 | 2026-01-10 03:29:11 | 2
{"sub_task_count":12,"read_local_sec":0,"read_local_mb":218,"read_remote_sec"
|
...
```

Возвращаются следующие поля:

- `Partition`: Партиция, к которой принадлежит задача compaction.
- `TxnID`: Идентификатор транзакции, назначенный задаче compaction.
- `StartTime`: Время начала задачи compaction. `NULL` означает, что задача ещё не инициирована.
- `CommitTime`: Время фиксации данных задачи compaction. `NULL` означает, что данные ещё не зафиксированы.
- `FinishTime`: Время публикации данных задачи compaction. `NULL` означает, что данные ещё не опубликованы.
- `Error`: Сообщение об ошибке (если есть) задачи compaction.
- `Profile`: (поддерживается с v1.5.2) Профиль задачи compaction после завершения.
  - `sub_task_count`: Количество подзадач (эквивалентно tablet) в партиции.
  - `read_local_sec`: Общее время чтения данных из локального кэша всеми подзадачами. Единица: секунды.
  - `read_local_mb`: Общий размер данных, прочитанных из локального кэша всеми подзадачами. Единица: МБ.

- `read_remote_sec`: Общее время чтения данных из удалённого хранилища всеми подзадачами. Единица: секунды.
- `read_remote_mb`: Общий размер данных, прочитанных из удалённого хранилища всеми подзадачами. Единица: МБ.
- `read_segment_count`: Общее количество файлов, прочитанных всеми подзадачами.
- `write_segment_count`: Общее количество новых файлов, сгенерированных всеми подзадачами.
- `write_segment_mb`: Общий размер новых файлов, сгенерированных всеми подзадачами. Единица: МБ.
- `write_remote_sec`: Общее время записи данных в удалённое хранилище всеми подзадачами. Единица: секунды.
- `in_queue_sec`: Общее время нахождения всех подзадач в очереди. Единица: секунды.

Просмотр деталей выполнения задач compaction

Каждая задача compaction делится на несколько подзадач, каждая из которых соответствует tablet. Вы можете просмотреть детали выполнения каждой подзадачи, запросив системное представление `information_schema.be_cloud_native_compactions`.

Пример:

```
mysql> SELECT * FROM information_schema.be_cloud_native_compactions;
+-------+--------+-----------+---------+---------+------+--------------------
-------------------------------------------------------------------------------
| BE_ID | TXN_ID | TABLET_ID | VERSION | SKIPPED | RUNS | START_TIME
|
+-------+--------+-----------+---------+---------+------+--------------------
-------------------------------------------------------------------------------
| 10001 |  51047 |     43034 |      12 |       0 |    1 | 2024-09-24 19:15:15
{"read_local_sec":0,"read_local_mb":31,"read_remote_sec":0,"read_remote_mb":0
|
...
```

Возвращаются следующие поля:

- `BE_ID`: Идентификатор CN.
- `TXN_ID`: Идентификатор транзакции, к которой принадлежит подзадача.

- `TABLET_ID`: Идентификатор tablet, к которому принадлежит подзадача.
- `VERSION`: Версия tablet.
- `RUNS`: Количество выполнений подзадачи.
- `START_TIME`: Время начала подзадачи.
- `FINISH_TIME`: Время завершения подзадачи.
- `PROGRESS`: Прогресс compaction tablet в процентах.
- `STATUS`: Статус подзадачи. Сообщения об ошибках будут возвращены в этом поле, если есть ошибка.
- `PROFILE`: (поддерживается с v1.5.2) Профиль выполнения подзадачи.
  - `read_local_sec`: Время чтения данных из локального кэша подзадачей. Единица: секунды.
  - `read_local_mb`: Размер данных, прочитанных из локального кэша подзадачей. Единица: МБ.
  - `read_remote_sec`: Время чтения данных из удалённого хранилища подзадачей. Единица: секунды.
  - `read_remote_mb`: Размер данных, прочитанных из удалённого хранилища подзадачей. Единица: МБ.
  - `read_local_count`: Количество чтений данных из локального кэша подзадачей.
  - `read_remote_count`: Количество чтений данных из удалённого хранилища подзадачей.
  - `in_queue_sec`: Время нахождения подзадачи в очереди. Единица: секунды.

Настройка задач compaction

Вы можете настроить задачи compaction с помощью следующих параметров FE и CN (BE).

Параметры FE

Вы можете динамически настроить следующий параметр FE.

```
ADMIN SET FRONTEND CONFIG ("lake_compaction_max_tasks" = "-1");
```

lake_compaction_max_tasks

- По умолчанию: -1
- Тип: Int
- Единица: -
- Изменяемый: Да

- Описание: Максимальное количество одновременных задач Compaction, разрешённых в cluster с общими данными. Установка этого элемента в `-1` означает расчёт количества одновременных задач адаптивным способом, то есть количество активных узлов CN, умноженное на 16. Установка этого значения в `0` отключит compaction.
- Введён в: v1.5.2

```
ADMIN SET FRONTEND CONFIG ("lake_compaction_disable_tables" =
"11111;22222");
```

lake_compaction_disable_tables

- По умолчанию: ""
- Тип: String
- Единица: -
- Изменяемый: Да
- Описание: Отключение compaction для определённых таблиц. Это не повлияет на уже начавшийся compaction. Значение этого элемента — идентификатор таблицы. Несколько значений разделяются `;`.
- Введён в: v1.5.2

Параметры CN

Вы можете динамически настроить следующий параметр CN.

```
UPDATE information_schema.be_configs SET VALUE = 8
WHERE name = "compact_threads";
```

compact_threads

- По умолчанию: 4
- Тип: Int
- Единица: -
- Изменяемый: Да
- Описание: Максимальное количество потоков, используемых для одновременных задач compaction. Эта конфигурация стала динамической начиная с v1.5.2.
- Введён в: v1.5.2

> **ПРИМЕЧАНИЕ**

> В продакшене рекомендуется устанавливать `compact_threads` на 25% от количества ядер CPU BE/CN.

max_cumulative_compaction_num_singleton_deltas

- По умолчанию: 500
- Тип: Int
- Единица: -
- Изменяемый: Да
- Описание: Максимальное количество сегментов, которые могут быть объединены в одном Cumulative Compaction. Вы можете уменьшить это значение, если во время compaction происходит OOM.
- Введён в: -

> **ПРИМЕЧАНИЕ**
>
> В продакшене рекомендуется устанавливать `max_cumulative_compaction_num_singleton_deltas` на `100` для ускорения задач compaction и снижения потребления ресурсов.

lake_pk_compaction_max_input_rowsets

- По умолчанию: 500
- Тип: Int
- Единица: -
- Изменяемый: Да
- Описание: Максимальное количество входных rowset, разрешённых в задаче compaction таблицы с первичным ключом в cluster с общими данными. Значение по умолчанию этого параметра было изменено с `5` на `1000` с v1.5.2. После включения политики Sized-tiered Compaction для таблиц с первичным ключом (путём установки `enable_pk_size_tiered_compaction_strategy` в `true`) Selena не нужно ограничивать количество rowset для каждого compaction, чтобы уменьшить амплификацию записи. Поэтому значение по умолчанию этого параметра было увеличено.
- Введён в: v1.5.2

> **ПРИМЕЧАНИЕ**
>
> В продакшене рекомендуется устанавливать `max_cumulative_compaction_num_singleton_deltas` на `100` для ускорения задач

> Compaction и снижения потребления ресурсов.

Ручной запуск задач compaction

```
-- Запуск compaction для всей таблицы.
ALTER TABLE <table_name> COMPACT;
-- Запуск compaction для определённой партиции.
ALTER TABLE <table_name> COMPACT <partition_name>;
-- Запуск compaction для нескольких партиций.
ALTER TABLE <table_name> COMPACT (<partition_name>, <partition_name>, ...);
```

Отмена задач compaction

Вы можете вручную отменить задачу compaction, используя идентификатор транзакции задачи.

```
CANCEL COMPACTION WHERE TXN_ID = <TXN_ID>;
```

> **ПРИМЕЧАНИЕ**
>
> - Выражение CANCEL COMPACTION должно быть отправлено с узла Leader FE.
> - Выражение CANCEL COMPACTION применяется только к транзакциям, которые не были зафиксированы, то есть `CommitTime` равен NULL в результате `SHOW PROC '/compactions'`.
> - CANCEL COMPACTION — это асинхронный процесс. Вы можете проверить, была ли задача отменена, выполнив `SHOW PROC '/compactions'`.

Лучшие практики

Поскольку Compaction имеет решающее значение для производительности запросов, рекомендуется регулярно отслеживать статус слияния данных таблиц и партиций. Вот некоторые лучшие практики и рекомендации:

- Старайтесь увеличивать интервал между загрузками (избегайте сценариев с интервалами менее 10 секунд) и увеличивать размер пакета на загрузку (избегайте размеров пакетов менее 100 строк данных).
- Настройте количество параллельных рабочих потоков compaction на CN для ускорения выполнения задач. В продакшене рекомендуется устанавливать `compact_threads` на 25% от количества ядер CPU BE/CN.

- Отслеживайте статус задач Compaction с помощью `show proc '/compactions'` и `select * from information_schema.be_cloud_native_compactions;`.
- Отслеживайте Compaction Score и настройте оповещения на его основе. Встроенный шаблон мониторинга Grafana Selena включает эту метрику.
- Обращайте внимание на потребление ресурсов во время compaction, особенно на использование памяти. Шаблон мониторинга Grafana также включает эту метрику.

Устранение неполадок

Медленные запросы

Чтобы определить медленные запросы, вызванные несвоевременным Compaction, вы можете проверить в SQL Profile значение `SegmentsReadCount`, делённое на `TabletCount` в пределах одного Fragment. Если это большое значение, например, десятки или более, несвоевременный Compaction может быть причиной медленного запроса.

Высокий Max Compaction Score в cluster

1. Проверьте, находятся ли параметры, связанные с Compaction, в разумных диапазонах, используя `ADMIN SHOW FRONTEND CONFIG LIKE "%lake_compaction%"` и `SELECT * FROM information_schema.be_configs WHERE name = "compact_threads"`.
2. Проверьте, не застрял ли Compaction, используя `SHOW PROC '/compactions'`:
   - Если `CommitTime` остаётся NULL, проверьте системное представление `information_schema.be_cloud_native_compactions` для выяснения причины застревания Compaction.
   - Если `FinishTime` остаётся NULL, найдите причину сбоя публикации в журнале Leader FE, используя `TxnID`.
3. Проверьте, работает ли compaction медленно, используя `SHOW PROC '/compactions'`:
   - Если `sub_task_count` слишком большой (проверьте размер каждого tablet в этой партиции с помощью `SHOW PARTITIONS`), таблица может быть создана неправильно.
   - Если `read_remote_mb` слишком большой (более 30% от общего объёма прочитанных данных), проверьте размер диска сервера, а также квоту кэша через `SHOW BACKENDS` для поля `DataCacheMetrics`.
   - Если `write_remote_sec` слишком большой (более 90% от общего времени Compaction), запись в удалённое хранилище может быть слишком медленной. Это можно проверить, проверив метрики мониторинга, специфичные для общих данных, с ключевыми словами `single upload latency` и `multi upload latency`.

- Если `in_queue_sec` слишком большой (среднее время ожидания на tablet превышает 60 секунд), настройки параметров могут быть неразумными или другие запущенные Compaction слишком медленные.

2.2.12 Плавное завершение работы

Начиная с версии v1.5.2, Selena поддерживает плавное завершение работы (Graceful Exit).

Обзор

Плавное завершение работы — это механизм, предназначенный для поддержки **обновлений и перезапусков без прерывания работы** узлов FE, BE и CN в Selena. Его основная цель — минимизировать влияние на выполняющиеся запросы и задачи загрузки данных во время операций обслуживания, таких как перезапуск узла, последовательное обновление или масштабирование cluster.

Плавное завершение работы гарантирует, что:

- Узел **прекращает принимать новые задачи** после начала завершения работы;
- Существующим запросам и заданиям загрузки **разрешено завершиться** в контролируемом временном окне;
- Системные компоненты (FE/BE/CN) **координируют изменения статуса**, чтобы cluster корректно перенаправлял трафик.

Механизмы плавного завершения работы различаются для узлов FE и BE/CN, как описано ниже.

Механизм плавного завершения работы FE

Сигнал триггера

Плавное завершение работы FE инициируется командой:

```
stop_fe.sh -g
```

Это отправляет сигнал `SIGUSR1`, тогда как завершение по умолчанию (без опции `-g`) отправляет сигнал `SIGTERM`.

Осведомлённость балансировщика нагрузки

При получении сигнала:

- FE немедленно возвращает **HTTP 500** на конечной точке `/api/health`.
- Балансировщики нагрузки обнаруживают ухудшенное состояние в течение ~15 секунд и прекращают направлять новые соединения к этому FE.

Логика опустошения соединений и завершения работы

Follower FE

- Обрабатывает запросы только на чтение.
- Если у узла FE нет активных сессий, соединение закрывается немедленно.
- Если выполняется SQL, узел FE ожидает завершения выполнения перед закрытием сессии.

Leader FE

- Обработка запросов на чтение идентична обработке Follower.
- Обработка запросов на запись требует:
  - Закрытия BDBJE.
  - Разрешения завершения выбора нового Leader.
  - Перенаправления последующих записей на вновь избранного Leader.

Контроль тайм-аута

Если запрос выполняется слишком долго, FE принудительно завершает работу через **60 секунд** (настраивается с помощью опции `--timeout`).

Механизм плавного завершения работы BE/CN

Сигнал триггера

Плавное завершение работы BE инициируется командой:

```
stop_be.sh -g
```

Плавное завершение работы CN инициируется командой:

```
stop_cn.sh -g
```

Это отправляет сигнал `SIGTERM`, тогда как завершение по умолчанию (без опции `-g`) отправляет сигнал `SIGKILL`.

Переход состояния

После получения сигнала:

- Узел BE/CN помечает себя как **завершающийся**.
- Он отклоняет **новые фрагменты запросов**, возвращая `INTERNAL_ERROR`.
- Он продолжает обрабатывать существующие фрагменты.

Цикл ожидания для выполняющихся запросов

Поведение, при котором BE/CN ожидает завершения существующих фрагментов, контролируется параметром конфигурации BE/CN `loop_count_wait_fragments_finish` (по умолчанию: 2). Фактическая продолжительность ожидания равна `loop_count_wait_fragments_finish × 10 секунд` (то есть 20 секунд по умолчанию). Если фрагменты остаются после тайм-аута, BE/CN продолжает нормальное завершение работы (закрытие потоков, сети и других процессов).

Улучшенная осведомлённость FE

Начиная с версии v1.5.2, FE больше не помечает BE/CN как `DEAD` на основе сбоев heartbeat. Он корректно распознаёт состояние BE/CN "завершение работы", что позволяет значительно увеличить временные окна плавного завершения для завершения фрагментов.

Конфигурации

Конфигурации FE

`stop_fe.sh -g --timeout`

- Описание: Максимальное время ожидания перед принудительным завершением FE.
- По умолчанию: 60 (секунд)
- Как применить: Укажите в команде скрипта, например, `--timeout 120`.

Минимальное время обнаружения LB

- Описание: Балансировщику нагрузки требуется минимум 15 секунд для обнаружения ухудшенного состояния.
- По умолчанию: 15 (секунд)
- Как применить: Фиксированное значение

Конфигурации BE/CN

`loop_count_wait_fragments_finish`

- Описание: Продолжительность ожидания BE/CN для существующих фрагментов. Умножьте значение на 10 секунд.
- По умолчанию: 2
- Как применить: Измените в файле конфигурации BE/CN или обновите динамически.

`graceful_exit_wait_for_frontend_heartbeat`

- Описание: Ожидает ли BE/CN подтверждения **SHUTDOWN** от FE через heartbeat. Начиная с версии v1.5.2.
- По умолчанию: false
- Как применить: Измените в файле конфигурации BE/CN или обновите динамически.

`stop_be.sh -g --timeout`, `stop_cn.sh -g --timeout`

- Описание: Максимальное время ожидания перед принудительным завершением BE/CN. Установите значение больше, чем `loop_count_wait_fragments_finish` * 10, чтобы предотвратить завершение до достижения времени ожидания BE/CN.
- По умолчанию: false
- Как применить: Укажите в команде скрипта, например, `--timeout 30`.

Глобальные переключатели

Плавное завершение работы **включено по умолчанию** начиная с версии v1.5.2. Чтобы временно отключить его, установите параметр конфигурации BE/CN `loop_count_wait_fragments_finish` в `0`.

Ожидаемое поведение во время плавного завершения работы

Рабочие нагрузки запросов

Тип запроса	Ожидаемое поведение
Короткий (менее 20 секунд)	BE/CN ожидает достаточно долго, поэтому запросы завершаются нормально.
Средний (от 20 до 60 секунд)	Запросы, завершённые в окне ожидания BE/CN, возвращаются успешно; иначе запросы отменяются и требуют ручного повтора.
Длительный (более 60 секунд)	Запросы, вероятно, будут прерваны FE/BE/CN из-за тайм-аута и требуют ручного повтора.

Задачи загрузки данных

- **Задачи загрузки через Flink или Kafka Connectors** автоматически повторяются без видимого пользователю прерывания.
- **Задачи Stream Load (без фреймворка), Broker Load и Routine Load** могут завершиться с ошибкой при разрыве соединения. Требуется ручной повтор.
- **Фоновые задачи** автоматически перепланируются и выполняются механизмом повтора FE.

Операции обновления и перезапуска

Плавное завершение работы обеспечивает:

- Отсутствие простоя всего cluster;
- Безопасное последовательное обновление путём опустошения узлов по одному.

Ограничения и различия версий

Различия поведения версий

Версия	Поведение
v1.5.2	Плавное завершение работы BE имеет недостатки: FE может преждевременно пометить BE/CN как `DEAD`, что приводит к отмене запросов. Эффективное ожидание ограничено (15 секунд по умолчанию).
v1.5.2+	Полностью поддерживает увеличенную продолжительность ожидания; FE корректно определяет состояние BE/CN "завершение работы". Рекомендуется для production.

Операционные ограничения

- В экстремальных случаях (например, зависание BE/CN) плавное завершение работы может не сработать. Завершение процесса требует `kill -9`, с риском частичной

персистентности данных (восстановимой через snapshot).

Использование

Предварительные требования

Версия Selena:

- **v1.5.2+**: Базовая поддержка плавного завершения работы.
- **v1.5.2+**: Улучшенное управление статусом, более длительные окна ожидания (до нескольких минут).

Конфигурация:

- Убедитесь, что `loop_count_wait_fragments_finish` установлен в положительное целое число.
- Установите `graceful_exit_wait_for_frontend_heartbeat` в `true`, чтобы FE мог обнаружить состояние BE "EXITING".

Выполнение плавного завершения работы FE

```
./bin/stop_fe.sh -g --timeout 60
```

Параметры:

- `--timeout`: Максимальное время ожидания перед принудительным завершением узла FE.

Поведение:

- Система сначала отправляет сигнал `SIGUSR1`.
- После тайм-аута переходит к `SIGKILL`.

Проверка состояния FE

Вы можете проверить состояние FE через следующий API:

```
http://<fe_host>:8030/api/health
```

Балансировщик нагрузки удаляет узел после получения последовательных ответов не-200.

Выполнение плавного завершения работы BE/CN

- **Для v1.5.2:**
  - BE:

```
./be/bin/stop_be.sh -g
```

  - CN:

```
./be/bin/stop_cn.sh -g
```

- **Для v1.5.2+:**
  - BE:

```
./bin/stop_be.sh -g --timeout 600
```

  - CN:

```
./bin/stop_cn.sh -g --timeout 600
```

BE/CN завершает работу немедленно, если фрагментов не осталось.

Проверка статуса BE/CN

Выполните на FE:

```
SHOW BACKENDS;
```

`StatusCode`:

- `SHUTDOWN`: Плавное завершение работы BE/CN в процессе.
- `DISCONNECTED`: Узел BE/CN полностью завершил работу.

Рабочий процесс последовательного обновления

Процедура

1. Выполните плавное завершение работы на узле `A`.
2. Убедитесь, что узел `A` отображается как `DISCONNECTED` на стороне FE.
3. Обновите и перезапустите узел `A`.
4. Повторите вышеуказанное для оставшихся узлов.

Мониторинг плавного завершения работы

Проверьте логи FE `fe.log`, логи BE `be.log` или логи CN `cn.log`, чтобы убедиться, были ли задачи во время завершения работы.

Устранение неполадок

BE/CN завершается по тайм-ауту

Если задачи не завершаются в течение периода плавного завершения работы, BE/CN запускает принудительное завершение (`SIGKILL`). Проверьте, вызвано ли это чрезмерно длительной продолжительностью задачи или неправильными конфигурациями (например, слишком малым значением `--timeout`).

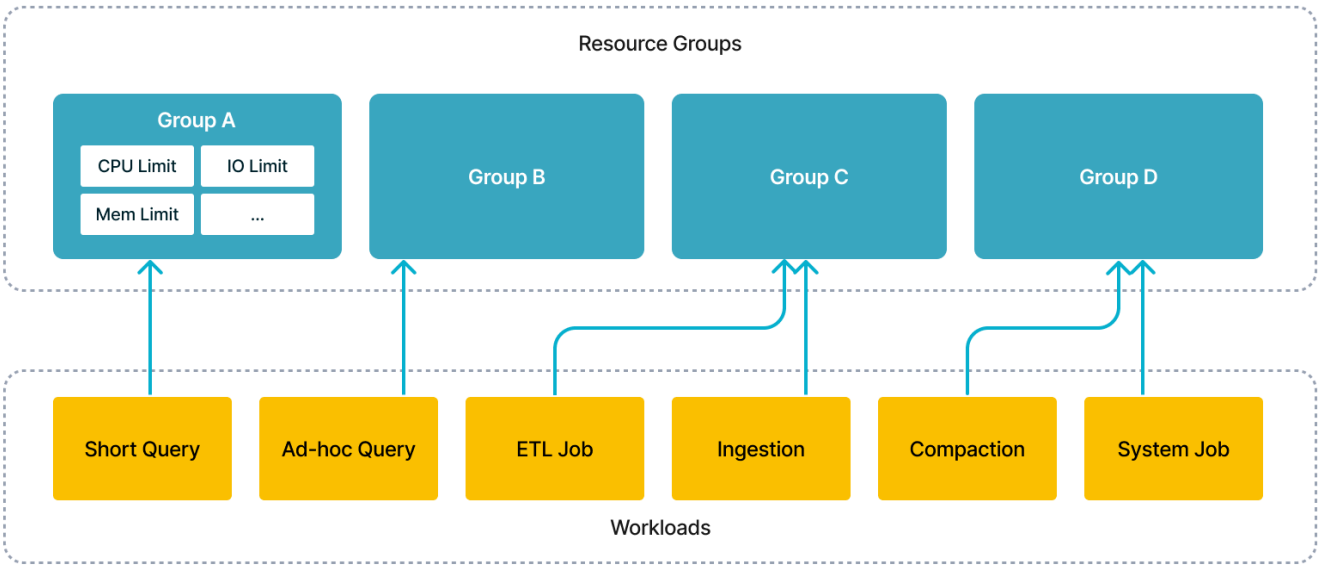
Статус узла не SHUTDOWN

Если статус узла не `SHUTDOWN`, проверьте, установлен ли `loop_count_wait_fragments_finish` в положительное целое число, или отправил ли BE/CN heartbeat перед завершением работы (если нет, установите `graceful_exit_wait_for_frontend_heartbeat` в `true`).

2.2.13 Управление ресурсами

Группа ресурсов

В этом разделе описывается функция группы ресурсов в Selena.



С помощью этой функции вы можете одновременно запускать несколько рабочих нагрузок в одном cluster, включая короткие запросы, ad-hoc запросы, ETL-задания, чтобы сэкономить дополнительные затраты на развёртывание нескольких cluster. С технической точки зрения, движок выполнения планирует параллельные рабочие нагрузки в соответствии со спецификациями пользователей и изолирует помехи между ними.

Дорожная карта группы ресурсов:

- Начиная с версии v1.5.2, Selena поддерживает ограничение потребления ресурсов для запросов и реализацию изоляции и эффективного использования ресурсов между арендаторами в одном cluster.
- В Selena v1.5.2 вы можете дополнительно ограничить потребление ресурсов для больших запросов и предотвратить исчерпание ресурсов cluster чрезмерно большими запросами, чтобы гарантировать стабильность системы.
- Selena v1.5.2 поддерживает ограничение потребления вычислительных ресурсов для загрузки данных (INSERT).
- Начиная с версии v1.5.2, Selena поддерживает наложение жёстких ограничений на ресурсы CPU.

	Внутренняя таблица	Внешняя таблица	Ограничение больших запросов	INSERT INTO	Broker Load	Routine Load, Stream Load, Schema Change	Жёсткий лимит CPU
2.2	√	×	×	×	×	×	x
2.3	√	√	√	×	×	×	x
2.5	√	√	√	√	×	×	x

	Внутренняя таблица	Внешняя таблица	Ограничение больших запросов	INSERT INTO	Broker Load	Routine Load, Stream Load, Schema Change	Жёсткий лимит CPU
3.1 & 3.2	√	√	√	√	√	×	x
3.3.5 и позже	√	√	√	√	√	×	√

Термины

В этом разделе описываются термины, которые необходимо понять перед использованием функции группы ресурсов.

Группа ресурсов

Каждая группа ресурсов — это набор вычислительных ресурсов с определённого BE. Вы можете разделить каждый BE вашего cluster на несколько групп ресурсов. Когда запрос назначается группе ресурсов, Selena выделяет ресурсы CPU и памяти для группы ресурсов на основе квот ресурсов, которые вы указали для группы ресурсов.

Вы можете указать квоты ресурсов CPU и памяти для группы ресурсов на BE, используя следующие параметры:

Параметр	Описание	Диапазон значений	По умолчанию
cpu_weight	Вес планирования CPU этой группы ресурсов на узле BE.	(0, `avg_be_cpu_cores`] (действует при значении больше 0)	0
exclusive_cpu_cores	Параметр жёсткой изоляции CPU для этой группы ресурсов.	(0, `min_be_cpu_cores - 1`] (действует при значении больше 0)	0
mem_limit	Процент памяти, доступной для запросов этой группы ресурсов на текущем узле BE.	(0, 1] (обязательно)	-
mem_pool	Группирует группы ресурсов для совместного использования лимита памяти.	Строка	default_mem_pool
spill_mem_limit_threshold	Порог использования памяти, который вызывает сброс на диск.	(0, 1]	1.0
concurrency_limit	Максимальное количество параллельных запросов в этой группе ресурсов.	Целое число (действует при значении больше 0)	0

Параметр	Описание	Диапазон значений	По умолчанию
big_query_cpu_second_limit	Максимальное время CPU (в секундах) для задач больших запросов на каждом узле BE.	Целое число (действует при значении больше 0)	0
big_query_scan_rows_limit	Максимальное количество строк, которые задачи больших запросов могут сканировать на каждом узле BE.	Целое число (действует при значении больше 0)	0
big_query_mem_limit	Максимальная память, которую задачи больших запросов могут использовать на каждом узле BE.	Целое число (действует при значении больше 0)	0

Параметры ресурсов CPU

`cpu_weight`

Этот параметр указывает вес планирования CPU группы ресурсов на одном узле BE, определяя относительную долю времени CPU, выделенную задачам из этой группы. До версии v1.5.2 это называлось `cpu_core_limit`.

Диапазон значений: (0, `avg_be_cpu_cores`], где `avg_be_cpu_cores` — среднее количество ядер CPU на всех узлах BE. Параметр действует только при значении больше 0. Должен быть установлен либо cpu_weight, либо exclusive_cpu_cores больше 0, но не оба одновременно.

> **ПРИМЕЧАНИЕ**
>
> Например, предположим, что три группы ресурсов rg1, rg2 и rg3 имеют значения cpu_weight 2, 6 и 8 соответственно. На полностью загруженном узле BE эти группы получат 12.5%, 37.5% и 50% времени CPU. Если узел не полностью загружен и rg1 и rg2 под нагрузкой, а rg3 простаивает, rg1 и rg2 получат 25% и 75% времени CPU соответственно.

`exclusive_cpu_cores`

Этот параметр определяет жёсткий лимит CPU для группы ресурсов. Он имеет два значения:

- **Эксклюзивность**: Резервирует `exclusive_cpu_cores` ядер CPU исключительно для этой группы ресурсов, делая их недоступными для других групп, даже когда они простаивают.

- **Квота**: Ограничивает группу ресурсов использованием только этих зарезервированных ядер CPU, предотвращая использование доступных ресурсов CPU из других групп.

Диапазон значений: (0, `min_be_cpu_cores - 1`], где `min_be_cpu_cores` — минимальное количество ядер CPU на всех узлах BE. Действует только при значении больше 0. Только один из `cpu_weight` или `exclusive_cpu_cores` может быть установлен больше 0.

- Группы ресурсов с `exclusive_cpu_cores` больше 0 называются Эксклюзивными группами ресурсов, а ядра CPU, выделенные им, называются Эксклюзивными ядрами. Другие группы называются Общими группами ресурсов и работают на Общих ядрах.
- Общее количество `exclusive_cpu_cores` по всем группам ресурсов не может превышать `min_be_cpu_cores - 1`. Верхний предел установлен, чтобы оставить как минимум одно Общее ядро доступным.

Связь между `exclusive_cpu_cores` и `cpu_weight`:

Только один из `cpu_weight` или `exclusive_cpu_cores` может быть активен одновременно. Эксклюзивные группы ресурсов работают на своих собственных зарезервированных Эксклюзивных ядрах без необходимости делить время CPU через `cpu_weight`.

Вы можете настроить, могут ли Общие группы ресурсов заимствовать Эксклюзивные ядра у Эксклюзивных групп ресурсов, используя конфигурацию BE `enable_resource_group_cpu_borrowing`. При значении `true` (по умолчанию) Общие группы могут заимствовать ресурсы CPU, когда Эксклюзивные группы простаивают.

Чтобы изменить эту конфигурацию динамически, используйте следующую команду:

```
UPDATE information_schema.be_configs SET VALUE = "false" WHERE NAME =
"enable_resource_group_cpu_borrowing";
```

Параметры ресурсов памяти

`mem_limit`

Указывает процент памяти (пул запросов), доступной для группы ресурсов на текущем узле BE. Диапазон значений: (0,1].

`mem_pool`

Начиная с версии v2.0.0, указывает идентификатор общего пула памяти. Группы ресурсов с одинаковым идентификатором mem_pool используют общий пул памяти, коллективно ограниченный `mem_limit`. Если не указано, группа ресурсов назначается в `default_mem_pool`, и её использование памяти ограничивается только её собственным `mem_limit`.

Все группы ресурсов, использующие один и тот же `mem_pool`, должны быть настроены с одинаковым `mem_limit`.

Чтобы ограничить две группы ресурсов совокупным потреблением 50% памяти, их можно определить следующим образом:

```
CREATE RESOURCE GROUP rg1
TO (db='db1')
WITH (
    'mem_limit' = '50%',
    'mem_pool' = 'shared_pool'
);

CREATE RESOURCE GROUP rg2
TO (db='db1')
WITH (
    'mem_limit' = '50%',
    'mem_pool' = 'shared_pool'
);
```

`spill_mem_limit_threshold`

Определяет порог использования памяти, который вызывает сброс на диск. Диапазон значений: (0,1], по умолчанию 1 (неактивно). Введён в версии v1.5.2.

- Когда автоматический сброс включён (`spill_mode` установлен в `auto`), но группы ресурсов отключены, система сбросит промежуточные результаты на диск, когда использование памяти запроса превысит 80% от `query_mem_limit`.
- Когда группы ресурсов включены, сброс произойдёт, если:
  - Общее использование памяти всех запросов в группе превышает `текущий лимит памяти BE * mem_limit * spill_mem_limit_threshold`, или
  - Использование памяти текущего запроса превышает 80% от `query_mem_limit`.

Параметры параллелизма запросов

`concurrency_limit`

Определяет максимальное количество параллельных запросов в группе ресурсов для предотвращения перегрузки системы. Действует только при значении больше 0, по умолчанию 0.

Параметры ресурсов больших запросов

Вы можете настроить лимиты ресурсов специально для больших запросов, используя следующие параметры:

`big_query_cpu_second_limit`

Указывает максимальное время CPU (в секундах), которое задачи больших запросов могут использовать на каждом узле BE, суммируя фактическое время CPU, использованное параллельными задачами. Действует только при значении больше 0, по умолчанию 0.

`big_query_scan_rows_limit`

Устанавливает лимит на количество строк, которые задачи больших запросов могут сканировать на каждом узле BE. Действует только при значении больше 0, по умолчанию 0.

`big_query_mem_limit`

Определяет максимальную память, которую задачи больших запросов могут использовать на каждом узле BE, в байтах. Действует только при значении больше 0, по умолчанию 0.

> **ПРИМЕЧАНИЕ**
>
> Когда запрос, выполняющийся в группе ресурсов, превышает указанный выше лимит больших запросов, запрос будет прерван с ошибкой. Вы также можете просматривать сообщения об ошибках в столбце `ErrorCode` узла FE **fe.audit.log**.

Тип (Устарел с версии v1.5.2)

До версии v1.5.2 Selena позволяла устанавливать `type` группы ресурсов в `short_query`. Однако параметр `type` устарел и заменён на `exclusive_cpu_cores`. Для любых существующих групп ресурсов этого типа система автоматически преобразует их в

Эксклюзивную группу ресурсов, где значение `exclusive_cpu_cores` равно `cpu_weight` после обновления до версии v1.5.2.

Системные группы ресурсов

В каждом экземпляре Selena есть две системные группы ресурсов: `default_wg` и `default_mv_wg`. Вы можете изменить конфигурацию системных групп ресурсов с помощью команды ALTER RESOURCE GROUP, но не можете определять для них классификаторы или удалять системные группы ресурсов.

default_wg

`default_wg` будет назначена обычным запросам, которые находятся под управлением групп ресурсов, но не соответствуют ни одному классификатору. Лимиты ресурсов по умолчанию для `default_wg` следующие:

- `cpu_weight`: Количество ядер CPU BE.
- `mem_limit`: 100%.
- `concurrency_limit`: 0.
- `big_query_cpu_second_limit`: 0.
- `big_query_scan_rows_limit`: 0.
- `big_query_mem_limit`: 0.
- `spill_mem_limit_threshold`: 1.

default_mv_wg

`default_mv_wg` будет назначена задачам обновления асинхронных материализованных представлений, если группа ресурсов не была выделена соответствующему материализованному представлению в свойстве `resource_group` при создании материализованного представления. Лимиты ресурсов по умолчанию для `default_mv_wg` следующие:

- `cpu_weight`: 1.
- `mem_limit`: 80%.
- `concurrency_limit`: 0.
- `spill_mem_limit_threshold`: 80%.

Классификатор

Каждый классификатор содержит одно или несколько условий, которые могут быть сопоставлены со свойствами запросов. Selena определяет классификатор, который лучше

всего соответствует каждому запросу на основе условий соответствия, и назначает ресурсы для выполнения запроса.

Классификаторы поддерживают следующие условия:

- `user`: имя пользователя.
- `role`: роль пользователя.
- `query_type`: тип запроса. Поддерживаются `SELECT` и `INSERT` (с версии v1.5.2). Когда задачи INSERT INTO или BROKER LOAD попадают в группу ресурсов с `query_type` равным `insert`, узел BE резервирует указанные ресурсы CPU для задач.
- `source_ip`: блок CIDR, из которого инициирован запрос.
- `db`: база данных, к которой обращается запрос. Может быть указана строками, разделёнными запятыми `,`.
- `plan_cpu_cost_range`: Диапазон предполагаемой стоимости CPU запроса. Формат: `(DOUBLE, DOUBLE]`. По умолчанию NULL, что означает отсутствие такого ограничения. Столбец `PlanCpuCost` в `fe.audit.log` представляет оценку системы стоимости CPU для запроса. Этот параметр поддерживается с версии v1.5.2.
- `plan_mem_cost_range`: Диапазон предполагаемой стоимости памяти запроса. Формат: `(DOUBLE, DOUBLE]`. По умолчанию NULL, что означает отсутствие такого ограничения. Столбец `PlanMemCost` в `fe.audit.log` представляет оценку системы стоимости памяти для запроса. Этот параметр поддерживается с версии v1.5.2.

Классификатор соответствует запросу только тогда, когда одно или все условия классификатора соответствуют информации о запросе. Если несколько классификаторов соответствуют запросу, Selena вычисляет степень соответствия между запросом и каждым классификатором и определяет классификатор с наивысшей степенью соответствия.

> **ПРИМЕЧАНИЕ**
>
> Вы можете просмотреть группу ресурсов, к которой принадлежит запрос, в столбце `ResourceGroup` узла FE **fe.audit.log** или выполнив `EXPLAIN VERBOSE <query>`, как описано в Просмотр группы ресурсов запроса.

Selena вычисляет степень соответствия между запросом и классификатором, используя следующие правила:

- Если классификатор имеет то же значение `user`, что и запрос, степень соответствия классификатора увеличивается на 1.

- Если классификатор имеет то же значение `role`, что и запрос, степень соответствия классификатора увеличивается на 1.
- Если классификатор имеет то же значение `query_type`, что и запрос, степень соответствия классификатора увеличивается на 1 плюс число, полученное из следующего расчёта: 1/Количество полей `query_type` в классификаторе.
- Если классификатор имеет то же значение `source_ip`, что и запрос, степень соответствия классификатора увеличивается на 1 плюс число, полученное из следующего расчёта: (32 - `cidr_prefix`)/64.
- Если классификатор имеет то же значение `db`, что и запрос, степень соответствия классификатора увеличивается на 10.
- Если стоимость CPU запроса попадает в `plan_cpu_cost_range`, степень соответствия классификатора увеличивается на 1.
- Если стоимость памяти запроса попадает в `plan_mem_cost_range`, степень соответствия классификатора увеличивается на 1.

Если несколько классификаторов соответствуют запросу, классификатор с большим количеством условий имеет более высокую степень соответствия.

```
-- Классификатор B имеет больше условий, чем классификатор A. Поэтому
классификатор B имеет более высокую степень соответствия, чем классификатор
A.


classifier A (user='Alice')


classifier B (user='Alice', source_ip = '192.168.1.0/24')
```

Если несколько соответствующих классификаторов имеют одинаковое количество условий, классификатор, условия которого описаны более точно, имеет более высокую степень соответствия.

```
-- Блок CIDR, указанный в классификаторе B, меньше по диапазону, чем
классификатор A. Поэтому классификатор B имеет более высокую степень
соответствия, чем классификатор A.
classifier A (user='Alice', source_ip = '192.168.1.0/16')
classifier B (user='Alice', source_ip = '192.168.1.0/24')

-- Классификатор C имеет меньше типов запросов, указанных в нём, чем
классификатор D. Поэтому классификатор C имеет более высокую степень
соответствия, чем классификатор D.
classifier C (user='Alice', query_type in ('select'))
classifier D (user='Alice', query_type in ('insert','select'))
```

Если несколько классификаторов имеют одинаковую степень соответствия, один из классификаторов будет выбран случайным образом.

```
-- Если запрос одновременно обращается к db1 и db2, и классификаторы E и F
имеют
-- наивысшую степень соответствия среди попавших классификаторов, один из E
и F будет выбран случайным образом.
classifier E (db='db1')
classifier F (db='db2')
```

Изоляция вычислительных ресурсов

Вы можете изолировать вычислительные ресурсы между запросами, настроив группы ресурсов и классификаторы.

Включение групп ресурсов

Чтобы использовать группу ресурсов, вы должны включить Pipeline Engine для вашего cluster Selena:

```
-- Включить Pipeline Engine в текущей сессии.
SET enable_pipeline_engine = true;
-- Включить Pipeline Engine глобально.
SET GLOBAL enable_pipeline_engine = true;
```

> **ПРИМЕЧАНИЕ**

> Начиная с версии v1.5.2, группа ресурсов включена по умолчанию, и переменная сессии `enable_resource_group` устарела.

Создание групп ресурсов и классификаторов

Выполните следующее выражение, чтобы создать группу ресурсов, связать группу ресурсов с классификатором и выделить вычислительные ресурсы группе ресурсов:

```
CREATE RESOURCE GROUP <group_name>
TO (
    user='string',
    role='string',
    query_type in ('select'),
    source_ip='cidr'
) --Создать классификатор. Если вы создаёте более одного классификатора,
разделите классификаторы запятыми (`,`).
WITH (
    "{ cpu_weight | exclusive_cpu_cores }" = "INT",
    "mem_limit" = "m%",
    "concurrency_limit" = "INT",
    "type" = "str" --Тип группы ресурсов. Установите значение normal.
);
```

Пример:

```
CREATE RESOURCE GROUP rg1
TO
    (user='rg1_user1', role='rg1_role1', query_type in ('select'),
source_ip='192.168.x.x/24'),
    (user='rg1_user2', query_type in ('select'),
source_ip='192.168.x.x/24'),
    (user='rg1_user3', source_ip='192.168.x.x/24'),
    (user='rg1_user4'),
    (db='db1')
WITH (
    'exclusive_cpu_cores' = '10',
    'mem_limit' = '20%',
    'big_query_cpu_second_limit' = '100',
    'big_query_scan_rows_limit' = '100000',
    'big_query_mem_limit' = '1073741824'
);
```

Указание группы ресурсов (опционально)

Вы можете указать группу ресурсов для текущей сессии напрямую, включая `default_wg` и `default_mv_wg`.

```
SET resource_group = 'group_name';
```

Просмотр групп ресурсов и классификаторов

Выполните следующее выражение, чтобы запросить все группы ресурсов и классификаторы:

```
SHOW RESOURCE GROUPS ALL;
```

Выполните следующее выражение, чтобы запросить группы ресурсов и классификаторы вошедшего пользователя:

```
SHOW RESOURCE GROUPS;
```

Выполните следующее выражение, чтобы запросить указанную группу ресурсов и её классификаторы:

```
SHOW RESOURCE GROUP group_name;
```

Пример:

```
mysql> SHOW RESOURCE GROUPS ALL;
+---------------+-------+------------+---------------------+-----------+---
-------------------------+---------------------------+---------------------
+-------------------+---------------------------+-------------------------
--------------+
| name          | id    | cpu_weight | exclusive_cpu_cores | mem_limit |
big_query_cpu_second_limit | big_query_scan_rows_limit |
big_query_mem_limit | concurrency_limit | spill_mem_limit_threshold |
classifiers                            |
+---------------+-------+------------+---------------------+-----------+---
-------------------------+---------------------------+---------------------
+-------------------+---------------------------+-------------------------
--------------+
| default_mv_wg | 3     | 1          | 0                   | 80.0%     | 0
| 0                         | 0                   | null              | 80%
| (id=0, weight=0.0)                     |
| default_wg    | 2     | 1          | 0                   | 100.0%    | 0
| 0                         | 0                   | null              |
100%                      | (id=0, weight=0.0)                      |
| rge1          | 15015 | 0          | 6                   | 90.0%     | 0
| 0                         | 0                   | null              |
100%                      | (id=15016, weight=1.0, user=rg1_user)  |
| rgs1          | 15017 | 8          | 0                   | 90.0%     | 0
| 0                         | 0                   | null              |
100%                      | (id=15018, weight=1.0, user=rgs1_user) |
| rgs2          | 15019 | 8          | 0                   | 90.0%     | 0
| 0                         | 0                   | null              |
100%                      | (id=15020, weight=1.0, user=rgs2_user) |
+---------------+-------+------------+---------------------+-----------+---
-------------------------+---------------------------+---------------------
+-------------------+---------------------------+-------------------------
--------------+
```

> **ПРИМЕЧАНИЕ**
>
> В приведённом выше примере `weight` указывает степень соответствия.

Чтобы запросить все поля группы ресурсов, включая устаревшие поля.

Добавив ключевое слово `VERBOSE` к трём командам, упомянутым выше, вы можете просмотреть все поля группы ресурсов, включая устаревшие, такие как `type` и `max_cpu_cores`.

```
SHOW VERBOSE RESOURCE GROUPS ALL;
SHOW VERBOSE RESOURCE GROUPS;
SHOW VERBOSE RESOURCE GROUP group_name;
```

Управление группами ресурсов и классификаторами

Вы можете изменить квоты ресурсов для каждой группы ресурсов. Вы также можете добавлять или удалять классификаторы из групп ресурсов.

Выполните следующее выражение, чтобы изменить квоты ресурсов для существующей группы ресурсов:

```
ALTER RESOURCE GROUP group_name WITH (
    'cpu_core_limit' = 'INT',
    'mem_limit' = 'm%'
);
```

Выполните следующее выражение, чтобы удалить группу ресурсов:

```
DROP RESOURCE GROUP group_name;
```

Выполните следующее выражение, чтобы добавить классификатор в группу ресурсов:

```
ALTER RESOURCE GROUP <group_name> ADD (user='string', role='string',
query_type in ('select'), source_ip='cidr');
```

Выполните следующее выражение, чтобы удалить классификатор из группы ресурсов:

```
ALTER RESOURCE GROUP <group_name> DROP (CLASSIFIER_ID_1, CLASSIFIER_ID_2,
...);
```

Выполните следующее выражение, чтобы удалить все классификаторы группы ресурсов:

```
ALTER RESOURCE GROUP <group_name> DROP ALL;
```

Наблюдение за группами ресурсов

Просмотр группы ресурсов запроса

Для запросов, которые ещё не были выполнены, вы можете просмотреть группу ресурсов, которой соответствует запрос, в поле `RESOURCE GROUP`, возвращаемом `EXPLAIN VERBOSE <query>`.

Пока запрос выполняется, вы можете проверить, в какую группу ресурсов попал запрос, из поля `ResourceGroup`, возвращаемого `SHOW PROC '/current_queries'` и `SHOW PROC '/global_current_queries'`.

После завершения запроса вы можете просмотреть группу ресурсов, которой соответствовал запрос, проверив поле `ResourceGroup` в файле **fe.audit.log** на узле FE.

- Если запрос не находится под управлением групп ресурсов, значение столбца — пустая строка `""`.
- Если запрос находится под управлением групп ресурсов, но не соответствует ни одному классификатору, значение столбца — пустая строка `""`. Но этот запрос назначен группе ресурсов по умолчанию `default_wg`.

Мониторинг групп ресурсов

Вы можете настроить мониторинг и оповещения для ваших групп ресурсов.

Метрики FE и BE, связанные с группами ресурсов, следующие. Все метрики ниже имеют метку `name`, указывающую на соответствующую группу ресурсов.

Метрики FE

Следующие метрики FE предоставляют статистику только в пределах текущего узла FE:

Метрика	Единица	Тип	Описание
selena_fe_query_resource_group	Количество	Мгновенное	Количество исторически выполненных запросов в этой группе ресурсов (включая выполняющиеся в данный момент).
selena_fe_query_resource_group_latency	мс	Мгновенное	Процентиль задержки запросов для этой группы ресурсов. Метка `type` указывает конкретные процентили, включая `mean`, `75_quantile`, `95_quantile`, `98_quantile`, `99_quantile`, `999_quantile`.

Метрика	Единица	Тип	Описание
selena_fe_query_resource_group_err	Количество	Мгновенное	Количество запросов в этой группе ресурсов, которые завершились с ошибкой.
selena_fe_resource_group_query_queue_total	Количество	Мгновенное	Общее количество исторически поставленных в очередь запросов в этой группе ресурсов (включая выполняющиеся в данный момент). Эта метрика поддерживается с версии v1.5.2. Действительна только при включённых очередях запросов.
selena_fe_resource_group_query_queue_pending	Количество	Мгновенное	Количество запросов, в данный момент находящихся в очереди этой группы ресурсов. Эта метрика поддерживается с версии v1.5.2. Действительна только при включённых очередях запросов.
selena_fe_resource_group_query_queue_timeout	Количество	Мгновенное	Количество запросов в этой группе ресурсов, которые истекли по тайм-ауту в очереди. Эта метрика поддерживается с версии v1.5.2. Действительна только при включённых очередях запросов.

Метрики BE

Метрика	Единица	Тип	Описание
resource_group_running_queries	Количество	Мгновенное	Количество запросов, в данный момент выполняющихся в этой группе ресурсов.
resource_group_total_queries	Количество	Мгновенное	Количество исторически выполненных запросов в этой группе ресурсов (включая выполняющиеся в данный момент).
resource_group_bigquery_count	Количество	Мгновенное	Количество запросов в этой группе ресурсов, которые превысили лимит больших запросов.
resource_group_concurrency_overflow_count	Количество	Мгновенное	Количество запросов в этой группе ресурсов, которые превысили лимит `concurrency_limit`.
resource_group_mem_limit_bytes	Байты	Мгновенное	Лимит памяти для этой группы ресурсов.

Метрика	Единица	Тип	Описание
resource_group_mem_inuse_bytes	Байты	Мгновенное	Память, в данный момент используемая этой группой ресурсов.
resource_group_cpu_limit_ratio	Процент	Мгновенное	Соотношение `cpu_core_limit` этой группы ресурсов к общему `cpu_core_limit` всех групп ресурсов.
resource_group_inuse_cpu_cores	Количество	Среднее	Расчётное количество ядер CPU, используемых этой группой ресурсов. Это значение является приблизительной оценкой. Оно представляет среднее значение, рассчитанное на основе статистики двух последовательных сборов метрик. Эта метрика поддерживается с версии v1.5.2.
resource_group_cpu_use_ratio	Процент	Среднее	**Устарело** Соотношение временных квантов потока Pipeline, используемых этой группой ресурсов, к общим временным квантам потока Pipeline, используемым всеми группами ресурсов. Представляет среднее значение, рассчитанное на основе статистики двух последовательных сборов метрик.
resource_group_connector_scan_use_ratio	Процент	Среднее	**Устарело** Соотношение временных квантов потока Scan внешней таблицы, используемых этой группой ресурсов, к общим временным квантам потока Pipeline, используемым всеми группами ресурсов. Представляет среднее значение, рассчитанное на основе статистики двух последовательных сборов метрик.
resource_group_scan_use_ratio	Процент	Среднее	**Устарело** Соотношение временных квантов потока Scan внутренней таблицы, используемых этой группой ресурсов, к общим временным квантам потока Pipeline, используемым всеми группами ресурсов. Представляет среднее значение, рассчитанное на основе статистики двух последовательных сборов метрик.

Просмотр информации об использовании группы ресурсов

Начиная с версии v1.5.2, Selena поддерживает SQL-выражение SHOW USAGE RESOURCE GROUPS, которое используется для отображения информации об использовании для каждой группы ресурсов на всех BE. Описания каждого поля следующие:

- `Name`: Имя группы ресурсов.
- `Id`: ID группы ресурсов.
- `Backend`: IP или FQDN BE.
- `BEInUseCpuCores`: Количество ядер CPU, в данный момент используемых этой группой ресурсов на этом BE. Это значение является приблизительной оценкой.

- `BEInUseMemBytes`: Количество байт памяти, в данный момент используемых этой группой ресурсов на этом BE.
- `BERunningQueries`: Количество запросов из этой группы ресурсов, всё ещё выполняющихся на этом BE.

Обратите внимание:

- BE периодически отправляют эту информацию об использовании ресурсов в Leader FE с интервалом, указанным в `report_resource_usage_interval_ms`, который по умолчанию установлен на 1 секунду.
- Результаты покажут только строки, где по крайней мере один из `BEInUseCpuCores`/`BEInUseMemBytes`/`BERunningQueries` является положительным числом. Другими словами, информация отображается только тогда, когда группа ресурсов активно использует какие-либо ресурсы на BE.

Пример:

```
MySQL [(none)]> SHOW USAGE RESOURCE GROUPS;
+------------+----+-----------+-----------------+-----------------+--------
----------+
| Name       | Id | Backend   | BEInUseCpuCores | BEInUseMemBytes |
BERunningQueries |
+------------+----+-----------+-----------------+-----------------+--------
----------+
| default_wg | 0  | 127.0.0.1 | 0.100           | 1               | 5
|
+------------+----+-----------+-----------------+-----------------+--------
----------+
| default_wg | 0  | 127.0.0.2 | 0.200           | 2               | 6
|
+------------+----+-----------+-----------------+-----------------+--------
----------+
| wg1        | 0  | 127.0.0.1 | 0.300           | 3               | 7
|
+------------+----+-----------+-----------------+-----------------+--------
----------+
| wg2        | 0  | 127.0.0.1 | 0.400           | 4               | 8
|
+------------+----+-----------+-----------------+-----------------+--------
----------+
```

Просмотр информации о потоках для Эксклюзивных и Общих групп ресурсов

Выполнение запросов в основном включает три пула потоков: `pip_exec`, `pip_scan` и `pip_con_scan`.

- Эксклюзивные группы ресурсов работают в своих выделенных пулах потоков и привязаны к Эксклюзивным ядрам CPU, выделенным им.
- Общие группы ресурсов работают в общих пулах потоков и привязаны к оставшимся Общим ядрам CPU.

Потоки в этих трёх пулах следуют соглашению об именовании `{ pip_exec | pip_scan | pip_con_scan }_{ com | <resource_group_id> }`, где `com` относится к общему пулу потоков, а `<resource_group_id>` относится к ID Эксклюзивной группы ресурсов.

Вы можете просмотреть информацию о CPU, привязанном к каждому потоку BE, через системное представление `information_schema.be_threads`. Поля `BE_ID`, `NAME` и `BOUND_CPUS` представляют ID BE, имя потока и количество ядер CPU, привязанных к этому потоку, соответственно.

```
select * from information_schema.be_threads where name like '%pip_exec%';
select * from information_schema.be_threads where name like '%pip_scan%';
select * from information_schema.be_threads where name like
'%pip_con_scan%';
```

Пример:

```
select BE_ID, NAME, FINISHED_TASKS, BOUND_CPUS from
information_schema.be_threads where name like '%pip_exec_com%' and be_id =
10223;
+-------+--------------+----------------+------------+
| BE_ID | NAME         | FINISHED_TASKS | BOUND_CPUS |
+-------+--------------+----------------+------------+
| 10223 | pip_exec_com | 2091295        | 10         |
| 10223 | pip_exec_com | 2088025        | 10         |
| 10223 | pip_exec_com | 1637603        | 6          |
| 10223 | pip_exec_com | 1641260        | 6          |
| 10223 | pip_exec_com | 1634197        | 6          |
| 10223 | pip_exec_com | 1633804        | 6          |
| 10223 | pip_exec_com | 1638184        | 6          |
| 10223 | pip_exec_com | 1636374        | 6          |
| 10223 | pip_exec_com | 2095951        | 10         |
| 10223 | pip_exec_com | 2095248        | 10         |
| 10223 | pip_exec_com | 2098745        | 10         |
| 10223 | pip_exec_com | 2085338        | 10         |
| 10223 | pip_exec_com | 2101221        | 10         |
| 10223 | pip_exec_com | 2093901        | 10         |
| 10223 | pip_exec_com | 2092364        | 10         |
| 10223 | pip_exec_com | 2091366        | 10         |
+-------+--------------+----------------+------------+
```

Очереди запросов

В этом разделе описывается, как управлять очередями запросов в Selena.

Начиная с версии v1.5.2, Selena поддерживает очереди запросов. При включённых очередях запросов Selena автоматически ставит входящие запросы в очередь, когда достигнут порог параллелизма или лимит ресурсов, тем самым предотвращая ухудшение перегрузки. Ожидающие запросы ждут в очереди, пока не станет доступно достаточно вычислительных ресурсов для начала выполнения. Начиная с версии v1.5.2, Selena поддерживает настройку очередей запросов на уровне группы ресурсов.

Вы можете установить пороги использования CPU, использования памяти и параллелизма запросов для активации очередей запросов.

Дорожная карта:

Версия	Глобальная очередь запросов	Очередь на уровне группы ресурсов	Коллективное управление параллелизмом	Динамическая корректировка параллелизма
v1.5.2	□	×	×	×
v1.5.2	□	□	□	□

Включение очередей запросов

Очереди запросов отключены по умолчанию. Вы можете включить глобальные очереди или очереди на уровне группы ресурсов для загрузки INSERT, SELECT-запросов и запросов статистики, установив соответствующие глобальные переменные сессии.

Включение глобальных очередей запросов

- Включить очереди запросов для задач загрузки:

```
SET GLOBAL enable_query_queue_load = true;
```

- Включить очереди запросов для SELECT-запросов:

```
SET GLOBAL enable_query_queue_select = true;
```

- Включить очереди запросов для запросов статистики:

```
SET GLOBAL enable_query_queue_statistic = true;
```

Включение очередей запросов на уровне группы ресурсов

Начиная с версии v1.5.2, Selena поддерживает настройку очередей запросов на уровне группы ресурсов.

Чтобы включить очереди запросов на уровне группы ресурсов, вам также нужно установить `enable_group_level_query_queue` в дополнение к глобальным переменным сессии, упомянутым выше.

```
SET GLOBAL enable_group_level_query_queue = true;
```

Указание порогов ресурсов

Указание порогов ресурсов для глобальных очередей запросов

Вы можете установить пороги, активирующие очереди запросов, через следующие глобальные переменные сессии:

Переменная	По умолчанию	Описание
query_queue_concurrency_limit	0	Верхний предел параллельных запросов на BE. Вступает в силу только при значении больше `0`. Значение `0` означает, что лимит не установлен.
query_queue_mem_used_pct_limit	0	Верхний предел процента использования памяти на BE. Вступает в силу только при значении больше `0`. Значение `0` означает, что лимит не установлен. Диапазон: [0, 1]
query_queue_cpu_used_permille_limit	0	Верхний предел промилле использования CPU (использование CPU * 1000) на BE. Вступает в силу только при значении больше `0`. Значение `0` означает, что лимит не установлен. Диапазон: [0, 1000]

> **ПРИМЕЧАНИЕ**
>
> По умолчанию BE отправляет отчёт об использовании ресурсов в FE с интервалом в одну секунду. Вы можете изменить этот интервал, установив элемент конфигурации BE `report_resource_usage_interval_ms`.

Указание порогов ресурсов для очередей на уровне группы ресурсов

Начиная с версии v1.5.2, вы можете установить индивидуальные лимиты параллелизма (`concurrency_limit`) и лимиты ядер CPU (`max_cpu_cores`) при создании группы ресурсов. Когда инициируется запрос, если любое из потреблений ресурсов превышает порог ресурсов на глобальном уровне или уровне группы ресурсов, запрос будет помещён в очередь до тех пор, пока все потребления ресурсов не будут в пределах порога.

Переменная	По умолчанию	Описание
concurrency_limit	0	Лимит параллелизма для группы ресурсов на одном узле BE. Вступает в силу только при значении больше `0`.
max_cpu_cores	0	Лимит ядер CPU для этой группы ресурсов на одном узле BE. Вступает в силу только при значении больше `0`. Диапазон: [0, `avg_be_cpu_cores`], где `avg_be_cpu_cores` представляет среднее количество ядер CPU на всех узлах BE.

Вы можете использовать SHOW USAGE RESOURCE GROUPS для просмотра информации об использовании ресурсов для каждой группы ресурсов на каждом узле BE, как описано в Просмотр информации об использовании группы ресурсов.

Управление параллелизмом запросов

Когда количество выполняющихся запросов (`num_running_queries`) превышает глобальный лимит или `concurrency_limit` группы ресурсов, входящие запросы помещаются в очередь. Способ получения `num_running_queries` отличается между версиями < v1.5.2 и >= v1.5.2.

- В версиях < v1.5.2 `num_running_queries` сообщается BE с интервалом, указанным в `report_resource_usage_interval_ms`. Поэтому может быть некоторая задержка в идентификации изменений в `num_running_queries`. Например, если `num_running_queries`, сообщённое BE в данный момент, не превышает глобальный лимит или `concurrency_limit` группы ресурсов, но входящие запросы поступают и превышают `concurrency_limit` до следующего отчёта, эти входящие запросы будут выполнены без ожидания в очереди.

- В версиях >= v1.5.2 все выполняющиеся запросы коллективно управляются Leader FE. Каждый Follower FE уведомляет Leader FE при инициировании или завершении запроса, позволяя Selena обрабатывать сценарии, когда происходит внезапное увеличение запросов, превышающих `concurrency_limit`.

Настройка очередей запросов

Вы можете установить ёмкость очереди запросов и максимальный тайм-аут запросов в очередях через следующие глобальные переменные сессии:

Переменная	По умолчанию	Описание
query_queue_max_queued_queries	1024	Верхний предел запросов в очереди. При достижении этого порога входящие запросы отклоняются. Вступает в силу только при значении больше `0`.
query_queue_pending_timeout_second	300	Максимальный тайм-аут ожидающего запроса в очереди. При достижении этого порога соответствующий запрос отклоняется. Единица измерения: секунды.

Настройка динамической корректировки параллелизма запросов

Начиная с версии v1.5.2, для запросов, управляемых очередью запросов и выполняемых Pipeline Engine, Selena может динамически корректировать параллелизм запроса `pipeline_dop` для входящих запросов на основе текущего количества выполняющихся запросов `num_running_queries`, количества фрагментов `num_fragments` и параллелизма запроса `pipeline_dop`. Это позволяет динамически контролировать параллелизм запросов, минимизируя накладные расходы на планирование и обеспечивая оптимальное использование ресурсов BE. Дополнительную информацию о фрагментах и параллелизме запросов `pipeline_dop` см. в разделе Управление запросами - Настройка параллелизма запросов.

Для каждого запроса в очереди Selena поддерживает концепцию драйверов, которые представляют параллельные фрагменты запроса на одном BE. Его логическое значение `num_drivers`, представляющее общий параллелизм всех фрагментов этого запроса на одном BE, равно `num_fragments * pipeline_dop`. Когда поступает новый запрос, Selena корректирует параллелизм запроса `pipeline_dop` на основе следующих правил:

- Чем больше количество выполняющихся драйверов `num_drivers` превышает нижний предел параллельных драйверов `query_queue_driver_low_water`, тем ниже корректируется параллелизм запроса `pipeline_dop`.
- Selena ограничивает количество выполняющихся драйверов `num_drivers` ниже верхнего предела параллельных драйверов для запросов `query_queue_driver_high_water`.

Вы можете настроить динамическую корректировку параллелизма запроса `pipeline_dop`, используя следующие глобальные переменные сессии:

Переменная	По умолчанию	Описание
query_queue_driver_high_water	-1	Верхний предел параллельных драйверов для запроса. Вступает в силу только при неотрицательном значении. При значении `0` эквивалентно `avg_be_cpu_cores * 16`, где `avg_be_cpu_cores` представляет среднее количество ядер CPU на всех узлах BE. При значении больше `0` используется это значение напрямую.
query_queue_driver_low_water	-1	Нижний предел параллельных драйверов для запросов. Вступает в силу только при неотрицательном значении. При значении `0` эквивалентно `avg_be_cpu_cores * 8`. При значении больше `0` используется это значение напрямую.

Мониторинг очередей запросов

Вы можете просматривать информацию, связанную с очередями запросов, используя следующие методы.

SHOW PROC

Вы можете проверить количество выполняющихся запросов, использование памяти и CPU на узлах BE с помощью SHOW PROC:

```
mysql> SHOW PROC '/backends'\G
*************************** 1. row ***************************
...
    NumRunningQueries: 0
           MemUsedPct: 0.79 %
           CpuUsedPct: 0.0 %
```

SHOW PROCESSLIST

Вы можете проверить, находится ли запрос в очереди (когда `IsPending` равно `true`), с помощью SHOW PROCESSLIST:

```
mysql> SHOW PROCESSLIST;
+------+------+---------------------+-------+---------+--------------------
-+------+-------+-------------------+-----------+
| Id   | User | Host                | Db    | Command | ConnectionStartTime
| Time | State | Info              | IsPending |
+------+------+---------------------+-------+---------+--------------------
-+------+-------+-------------------+-----------+
|    2 | root | xxx.xx.xxx.xx:xxxxx |       | Query   | 2022-11-24 18:08:29
|    0 | OK    | SHOW PROCESSLIST  | false     |
+------+------+---------------------+-------+---------+--------------------
-+------+-------+-------------------+-----------+
```

Журнал аудита FE

Вы можете проверить файл журнала аудита FE **fe.audit.log**. Поле `PendingTimeMs` указывает время, которое запрос провёл в ожидании в очереди, и его единица измерения — миллисекунды.

Метрики мониторинга

Вы можете получить метрики очередей запросов в Selena, используя функцию Мониторинг и оповещения. Следующие метрики FE получены из статистических данных каждого узла FE.

Метрика	Единица	Тип	Описание
selena_fe_query_queue_pending	Количество	Мгновенное	Текущее количество запросов в очереди.
selena_fe_query_queue_total	Количество	Мгновенное	Общее количество исторически поставленных в очередь запросов (включая выполняющиеся в данный момент).
selena_fe_query_queue_timeout	Количество	Мгновенное	Общее количество запросов, которые истекли по тайм-ауту в очереди.
selena_fe_resource_group_query_queue_total	Количество	Мгновенное	Общее количество исторически поставленных в очередь запросов в этой группе ресурсов (включая выполняющиеся в данный момент). Метка `name` указывает имя группы ресурсов. Эта метрика поддерживается с версии v1.5.2.
selena_fe_resource_group_query_queue_pending	Количество	Мгновенное	Количество запросов, в данный момент находящихся в очереди для этой группы ресурсов. Метка `name` указывает имя группы ресурсов. Эта метрика поддерживается с версии v1.5.2.
selena_fe_resource_group_query_queue_timeout	Количество	Мгновенное	Количество запросов, которые истекли по тайм-ауту в очереди для этой группы ресурсов. Метка `name` указывает имя группы ресурсов. Эта метрика поддерживается с версии v1.5.2.

SHOW RUNNING QUERIES

Начиная с версии v1.5.2, Selena поддерживает SQL-выражение `SHOW RUNNING QUERIES`, которое используется для отображения информации об очереди для каждого запроса. Значения каждого поля следующие:

- `QueryId`: ID запроса.
- `ResourceGroupId`: ID группы ресурсов, в которую попал запрос. Когда нет попадания в определённую пользователем группу ресурсов, отображается "-".
- `StartTime`: Время начала запроса.
- `PendingTimeout`: Время, когда ожидающий запрос истечёт по тайм-ауту в очереди.
- `QueryTimeout`: Время, когда запрос истечёт по тайм-ауту.

- `State`: Состояние очереди запроса, где "PENDING" означает, что он находится в очереди, а "RUNNING" означает, что он выполняется в данный момент.
- `Slots`: Логическое количество ресурсов, запрошенных запросом, в настоящее время фиксировано на `1`.
- `Frontend`: Узел FE, инициировавший запрос.
- `FeStartTime`: Время запуска узла FE, инициировавшего запрос.

Пример:

```
MySQL [(none)]> SHOW RUNNING QUERIES;
+---------------------------------------+-----------------+-----------------
----+---------------------+---------------------+-----------+-------+------
---------------------------+---------------------+
| QueryId                               | ResourceGroupId | StartTime
| PendingTimeout      | QueryTimeout        |   State   | Slots | Frontend
| FeStartTime         |
+---------------------------------------+-----------------+-----------------
----+---------------------+---------------------+-----------+-------+------
---------------------------+---------------------+
| a46f68c6-3b49-11ee-8b43-00163e10863a | -               | 2023-08-15
16:56:37 | 2023-08-15 17:01:37 | 2023-08-15 17:01:37 |  RUNNING  | 1     |
127.00.00.01_9010_1692069711535 | 2023-08-15 16:37:03 |
| a6935989-3b49-11ee-935a-00163e13bca3 | 12003           | 2023-08-15
16:56:40 | 2023-08-15 17:01:40 | 2023-08-15 17:01:40 |  RUNNING  | 1     |
127.00.00.02_9010_1692069658426 | 2023-08-15 16:37:03 |
| a7b5e137-3b49-11ee-8b43-00163e10863a | 12003           | 2023-08-15
16:56:42 | 2023-08-15 17:01:42 | 2023-08-15 17:01:42 |  PENDING  | 1     |
127.00.00.03_9010_1692069711535 | 2023-08-15 16:37:03 |
+---------------------------------------+-----------------+-----------------
----+---------------------+---------------------+-----------+-------+------
---------------------------+---------------------+
```

Управление запросами

Количество пользовательских соединений

`Property` устанавливается на уровне пользователя. Для установки максимального количества соединений между клиентом и FE используйте следующую команду.

```
ALTER USER '<username>' SET PROPERTIES ("key"="value", ...)
```

Свойства пользователя включают ресурсы, назначенные пользователю. Свойства, установленные здесь, относятся к пользователю, а не к `user_identity`. То есть, если два пользователя `jack'@'%` и `jack'@'192.%` созданы оператором `CREATE USER`, то оператор `ALTER USER SET PROPERTIES` будет работать с пользователем `jack`, а не с `jack'@'%` или `jack'@'192.%`.

Пример 1:

```
-- Для пользователя `jack` изменить максимальное количество соединений на 1000
ALTER USER 'jack' SET PROPERTIES ("max_user_connections" = "1000");

-- Проверить лимит соединений для пользователя root
SHOW PROPERTY FOR 'root';
```

Переменные сессии, связанные с запросами

Переменные сессии могут быть установлены как 'key' = 'value', что позволяет ограничить параллелизм, память и другие параметры запросов в текущей сессии. Например:

- parallel_fragment_exec_instance_num

  Параллелизм запроса со значением по умолчанию 1. Указывает количество экземпляров фрагментов на каждом BE. Вы можете установить это значение равным половине количества ядер CPU BE для улучшения производительности запросов.

- query_mem_limit

  Лимит памяти запроса на каждом узле BE, может быть скорректирован, когда запрос сообщает о недостатке памяти.

- load_mem_limit

  Лимит памяти для импорта, может быть скорректирован, когда задача импорта сообщает о недостатке памяти.

Пример 2:

```
set parallel_fragment_exec_instance_num  = 8;
set query_mem_limit  = 137438953472;
```

Квота ёмкости хранилища базы данных

Квота ёмкости хранилища базы данных по умолчанию неограничена. Вы можете изменить значение квоты с помощью `alter database`.

```
ALTER DATABASE db_name SET DATA QUOTA quota;
```

Единицы измерения квоты: B/K/KB/M/MB/G/GB/T/TB/P/PB

Пример 3:

```
ALTER DATABASE example_db SET DATA QUOTA 10T;
```

Завершение запросов

Чтобы завершить запрос на определённом соединении, используйте следующую команду:

```
kill connection_id;
```

`connection_id` можно увидеть с помощью `show processlist;` или `select connection_id();`.

```
 show processlist;
+------+------------+---------------------+-----------------+--------------
-+---------+------+-------+------+
| Id   | User       | Host                | Cluster         | Db
| Command | Time | State | Info |
+------+------------+---------------------+-----------------+--------------
-+---------+------+-------+------+
|    1 | selenamgr | 172.26.34.147:56208 | default_cluster | selena_monitor
| Sleep   |    8 |       |      |
|  129 | root       | 172.26.92.139:54818 | default_cluster |
| Query   |    0 |       |      |
|  114 | test       | 172.26.34.147:57974 | default_cluster | ssb_100g
| Query   |    3 |       |      |
|    3 | selenamgr | 172.26.34.147:57268 | default_cluster | selena_monitor
| Sleep   |    8 |       |      |
|  100 | root       | 172.26.34.147:58472 | default_cluster | ssb_100
| Sleep   |  637 |       |      |
|  117 | selenamgr | 172.26.34.147:33790 | default_cluster | selena_monitor
| Sleep   |    8 |       |      |
|    6 | selenamgr | 172.26.34.147:57632 | default_cluster | selena_monitor
| Sleep   |    8 |       |      |
|  119 | selenamgr | 172.26.34.147:33804 | default_cluster | selena_monitor
| Sleep   |    8 |       |      |
|  111 | root       | 172.26.92.139:55472 | default_cluster |
| Sleep   | 2758 |       |      |
+------+------------+---------------------+-----------------+--------------
-+---------+------+-------+------+
9 rows in set (0.00 sec)

mysql> select connection_id();
+-----------------+
| CONNECTION_ID() |
+-----------------+
|              98 |
+-----------------+


mysql> kill 114;
Query OK, 0 rows affected (0.02 sec)
```

Управление памятью

В этом разделе кратко описаны классификация памяти и методы управления памятью в Selena.

Классификация памяти

Описание:

Метрика	Название	Описание
process	Общая используемая память BE	
query_pool	Память, используемая для выполнения запросов	Состоит из двух частей: память, используемая на уровне выполнения, и память, используемая на уровне хранения.
load	Память, используемая для загрузки данных	Обычно MemTable
table_meta	Память метаданных	Schema, метаданные Tablet, метаданные RowSet, метаданные Column, ColumnReader, IndexReader
compaction	Compaction многоверсионной памяти	compaction, которая происходит после завершения импорта данных
snapshot	Память снимков	Обычно используется для clone, малое потребление памяти
column_pool	Память пула столбцов	Запрос на освобождение кэша столбцов для ускорения работы со столбцами
page_cache	Собственный PageCache BE	По умолчанию отключён, пользователь может включить его, изменив файл BE

Конфигурация, связанная с памятью

- **Конфигурация BE**

Название	По умолчанию	Описание
vector_chunk_size	4096	Количество строк в chunk
mem_limit	90%	Верхний предел памяти процесса BE. Вы можете установить его как процент ("80%") или физический лимит ("100G"). Жёсткий лимит по умолчанию составляет 90% от размера памяти

Название	По умолчанию	Описание
		сервера, а мягкий лимит — 80%. Вам необходимо настроить этот параметр, если вы хотите развернуть Selena вместе с другими сервисами, интенсивно использующими память, на одном сервере.
disable_storage_page_cache	false	Логическое значение для управления отключением PageCache. Когда PageCache включён, Selena кэширует недавно просканированные данные. PageCache может значительно улучшить производительность запросов при частом повторении похожих запросов. `true` указывает на отключение PageCache. Используйте этот элемент вместе с `storage_page_cache_limit` для ускорения производительности запросов в сценариях с достаточными ресурсами памяти и большим объёмом сканирования данных. Значение по умолчанию этого элемента было изменено с `true` на `false` начиная с Selena v1.5.2.
write_buffer_size	104857600	Лимит ёмкости одного MemTable, при превышении которого выполняется сброс на диск.
load_process_max_memory_limit_bytes	107374182400	Верхний предел ресурсов памяти, которые могут быть заняты всеми процессами загрузки на узле BE. Его значение — меньшее из `mem_limit * load_process_max_memory_limit_percent / 100` и `load_process_max_memory_limit_bytes`. Если этот порог превышен, запускается flush и обратное давление.
load_process_max_memory_limit_percent	30	Максимальный процент ресурсов памяти, которые могут быть заняты всеми процессами загрузки на узле BE. Его значение — меньшее из `mem_limit * load_process_max_memory_limit_percent / 100` и `load_process_max_memory_limit_bytes`. Если этот порог превышен, запускается flush и обратное давление.
default_load_mem_limit	2147483648	Если лимит памяти на принимающей стороне достигнут для одного экземпляра импорта, запускается сброс на диск. Это необходимо изменить с помощью переменной сессии `load_mem_limit` для вступления в силу. Этот параметр является изменяемым, когда включена Event-based Compaction Framework.
max_compaction_concurrency	-1	Максимальный параллелизм compaction (как Base Compaction, так и Cumulative Compaction). Значение -1 указывает, что ограничение на параллелизм не накладывается.
cumulative_compaction_check_interval_seconds	1	Интервал проверки compaction

- **Переменные сессии**

Название	По умолчанию	Описание
query_mem_limit	0	Лимит памяти запроса на каждом узле BE
load_mem_limit	0	Лимит памяти одной задачи импорта. Если значение равно 0, берётся `exec_mem_limit`

Просмотр использования памяти

- **mem_tracker**

```
//Просмотр общей статистики памяти
<http://be_ip:be_http_port/mem_tracker>

// Просмотр детальной статистики памяти
<http://be_ip:be_http_port/mem_tracker?type=query_pool&upper_level=3>
```

- **tcmalloc**

```
<http://be_ip:be_http_port/memz>
```

```
------------------------------------------------
MALLOC:      777276768 (   741.3 MiB) Bytes in use by application
MALLOC: +   8851890176 ( 8441.8 MiB) Bytes in page heap freelist
MALLOC: +    143722232 (   137.1 MiB) Bytes in central cache freelist
MALLOC: +     21869824 (    20.9 MiB) Bytes in transfer cache freelist
MALLOC: +    832509608 (   793.9 MiB) Bytes in thread cache freelists
MALLOC: +     58195968 (    55.5 MiB) Bytes in malloc metadata
MALLOC:   ------------
MALLOC: =  10685464576 (10190.5 MiB) Actual memory used (physical + swap)
MALLOC: +  25231564800 (24062.7 MiB) Bytes released to OS (aka unmapped)
MALLOC:   ------------
MALLOC: =  35917029376 (34253.1 MiB) Virtual address space used
MALLOC:
MALLOC:         112388              Spans in use
MALLOC:            335              Thread heaps in use
MALLOC:           8192              Tcmalloc page size
------------------------------------------------
Call ReleaseFreeMemory() to release freelist memory to the OS (via
madvise()).
Bytes released to the OS take up virtual address space but no physical
memory.
```

Память, запрашиваемая этим методом, является точной. Однако некоторая память в Selena зарезервирована, но не используется. TcMalloc подсчитывает зарезервированную память, а не используемую.

Здесь `Bytes in use by application` относится к памяти, которая в данный момент используется.

- **metrics**

```
curl -XGET http://be_ip:be_http_port/metrics | grep 'mem'
curl -XGET http://be_ip:be_http_port/metrics | grep 'column_pool'
```

Значение metrics обновляется каждые 10 секунд. Возможно отслеживать некоторые статистики памяти со старыми версиями.

Сброс на диск

> *Beta* **BETA FEATURE**
>
> Рекомендации по использованию бета-функций

В этой теме описывается, как сбрасывать промежуточные результаты вычислений крупных операторов на локальные диски и объектное хранилище.

Обзор

Для систем баз данных, которые полагаются на вычисления в памяти для выполнения запросов, таких как Selena, они могут потреблять значительные ресурсы памяти при обработке запросов с операторами агрегации, сортировки и соединения на большом наборе данных. При достижении лимитов памяти эти запросы принудительно завершаются из-за нехватки памяти (OOM).

Однако всё ещё есть случаи, когда вы хотите, чтобы определённые задачи, интенсивно использующие память, были стабильно завершены, и производительность не является вашим главным приоритетом, например, построение материализованного представления или выполнение лёгкого ETL с INSERT INTO SELECT. Эти задачи могут легко исчерпать ваши ресурсы памяти и тем самым заблокировать другие запросы, выполняющиеся в вашем cluster. Обычно для решения этой проблемы вы можете только тонко настроить эти задачи индивидуально и полагаться на вашу стратегию изоляции ресурсов для контроля параллелизма запросов. Это может быть особенно неудобно и, вероятно, приведёт к сбою в некоторых экстремальных сценариях.

Начиная с Selena v1.5.2, Selena поддерживает сброс промежуточных результатов некоторых операторов, интенсивно использующих память, на диски. С помощью этой функции вы можете обменять допустимое снижение производительности на значительное сокращение использования памяти, тем самым улучшая доступность системы.

В настоящее время функция сброса Selena поддерживает следующие операторы:

- Операторы агрегации
- Операторы сортировки
- Операторы Hash join (LEFT JOIN, RIGHT JOIN, FULL JOIN, OUTER JOIN, SEMI JOIN и INNER JOIN)
- Операторы CTE (поддерживаются с версии v1.5.2)

Включение сброса промежуточных результатов

Выполните следующие шаги для включения сброса промежуточных результатов:

1. Укажите локальную директорию для сброса `spill_local_storage_dir`, которая хранит сброшенный промежуточный результат на локальном диске, в файле конфигурации BE **be.conf** или файле конфигурации CN **cn.conf**, и перезапустите cluster, чтобы изменения вступили в силу.

```
spill_local_storage_dir=/<dir_1>[;/<dir_2>]
```

> **ПРИМЕЧАНИЕ**
>
> - Вы можете указать несколько `spill_local_storage_dir`, разделяя их точкой с запятой (`;`).
> - В производственной среде мы настоятельно рекомендуем использовать разные диски для хранения данных и сброса. Когда промежуточные результаты сбрасываются на диск, может произойти значительное увеличение как нагрузки на запись, так и использования диска. Если используется один и тот же диск, этот всплеск может повлиять на другие запросы или задачи, выполняющиеся в cluster.

2. Выполните следующий оператор для включения сброса промежуточных результатов:

```
SET enable_spill = true;
```

3. Настройте режим сброса промежуточных результатов с помощью переменной сессии `spill_mode`:

```
SET spill_mode = { "auto" | "force" };
```

> **ПРИМЕЧАНИЕ**
>
> Каждый раз, когда запрос со сбросом завершается, Selena автоматически очищает сброшенные данные, которые произвёл запрос. Если BE аварийно завершается до очистки данных, Selena очищает их при перезапуске BE.

Переменная	По умолчанию	Описание
enable_spill	false	Включить ли сброс промежуточных результатов. Если установлено в `true`, Selena сбрасывает промежуточные результаты на диск для уменьшения использования памяти при обработке операторов агрегации, сортировки или соединения в запросах.
spill_mode	auto	Режим выполнения сброса промежуточных результатов. Допустимые значения: • `auto`: Сброс автоматически запускается при достижении порога использования памяти. • `force`: Selena принудительно выполняет сброс для всех соответствующих операторов, независимо от использования памяти. Эта переменная вступает в силу только когда переменная `enable_spill` установлена в `true`.

[Предварительный просмотр] Сброс промежуточных результатов в объектное хранилище

Начиная с версии v1.5.2, Selena поддерживает сброс промежуточных результатов в объектное хранилище.

> **ПОДСКАЗКА**
>
> Перед включением сброса в объектное хранилище необходимо создать том хранения для определения объектного хранилища, которое вы хотите использовать. Подробные инструкции по созданию тома хранения см. в CREATE STORAGE VOLUME.

После включения сброса на предыдущем шаге вы можете дополнительно установить эти системные переменные, чтобы разрешить сброс промежуточных результатов в объектное хранилище:

```
SET enable_spill_to_remote_storage = true;

-- Замените <storage_volume_name> на имя тома хранения, который вы хотите использовать.
SET spill_storage_volume = '<storage_volume_name>';
```

После включения сброса в объектное хранилище промежуточные результаты запросов, вызвавших сброс, сначала будут храниться на локальных дисках узлов BE или CN, а затем в объектном хранилище, если достигнут лимит ёмкости локальных дисков.

Обратите внимание, что если том хранения, указанный для `spill_storage_volume`, не существует, сброс в объектное хранилище не будет включён.

Ограничения

- Не все проблемы OOM могут быть решены с помощью сброса. Например, Selena не может освободить память, используемую для вычисления выражений.
- Обычно запросы со сбросом указывают на десятикратное увеличение задержки запроса. Мы рекомендуем увеличить тайм-аут запроса для этих запросов, установив переменную сессии `query_timeout`.
- Наблюдается значительное снижение производительности при сбросе в объектное хранилище по сравнению со сбросом на локальные диски.
- `spill_local_storage_dir` каждого узла BE или CN используется совместно всеми запросами, выполняющимися на узле. В настоящее время Selena не поддерживает установку лимита размера сброшенных данных на локальные диски индивидуально для каждого запроса. Поэтому параллельные запросы со сбросом могут влиять друг на друга.

Балансировка нагрузки

При развёртывании нескольких узлов FE пользователи могут развернуть слой балансировки нагрузки поверх FE для достижения высокой доступности.

Ниже приведены некоторые варианты обеспечения высокой доступности:

Подход на уровне кода

Один из способов — реализовать логику на уровне приложения для выполнения повторных попыток и балансировки нагрузки. Например, если соединение разрывается, оно автоматически повторяется на других соединениях. Этот подход требует, чтобы пользователи настроили несколько адресов узлов FE.

JDBC Connector

JDBC connector поддерживает автоматические повторные попытки:

```
jdbc:mysql:loadbalance://[host1][:port],[host2][:port][,[host3][:port]]...
[/[database]][?
propertyName1=propertyValue1[&propertyName2=propertyValue2]...]
```

ProxySQL

ProxySQL — это прокси-слой MySQL, который поддерживает разделение чтения/записи, маршрутизацию запросов, кэширование SQL, динамическую настройку нагрузки, отказоустойчивость и фильтрацию SQL.

Selena FE отвечает за приём соединений и запросов, и он горизонтально масштабируется и обладает высокой доступностью. Однако FE требует, чтобы пользователи настроили прокси-слой поверх него для достижения автоматической балансировки нагрузки. Ниже приведены шаги настройки:

1. Установка необходимых зависимостей

```
yum install -y gnutls perl-DBD-MySQL perl-DBI perl-devel
```

2. Загрузка установочного пакета

```
wget https://github.com/sysown/proxysql/releases/download/v1.5.2/proxysql-
2.0.14-1-centos7.x86_64.rpm
```

3. Распаковка в текущую директорию

```
rpm2cpio proxysql-2.0.14-1-centos7.x86_64.rpm | cpio -ivdm
```

4. Изменение файла конфигурации

```
vim ./etc/proxysql.cnf
```

Укажите директорию, к которой у пользователя есть права доступа (абсолютный путь):

```
datadir="/var/lib/proxysql"
errorlog="/var/lib/proxysql/proxysql.log"
```

5. Запуск

```
./usr/bin/proxysql -c ./etc/proxysql.cnf --no-monitor
```

6. Вход в систему

```
mysql -u admin -padmin -h 127.0.0.1 -P6032
```

7. Настройка глобального журнала

```
SET mysql-eventslog_filename='proxysql_queries.log';
SET mysql-eventslog_default_log=1;
SET mysql-eventslog_format=2;
LOAD MYSQL VARIABLES TO RUNTIME;
SAVE MYSQL VARIABLES TO DISK;
```

8. Добавление leader-узла

```
insert into mysql_servers(hostgroup_id, hostname, port) values(1,
'172.xx.xx.139', 9030);
```

9. Добавление observer-узлов

```
insert into mysql_servers(hostgroup_id, hostname, port) values(2,
'172.xx.xx.139', 9030);
insert into mysql_servers(hostgroup_id, hostname, port) values(2,
'172.xx.xx.140', 9030);
```

10. Загрузка конфигурации

```
load mysql servers to runtime;
save mysql servers to disk;
```

11. Настройка имени пользователя и пароля

```
insert into mysql_users(username, password, active, default_hostgroup,
backend, frontend) values('root',
'*94BDCEBE19083CE2A1F959FD02F964C7AF4CFC29', 1, 1, 1, 1);
```

12. Загрузка конфигурации

```
load mysql users to runtime;
save mysql users to disk;
```

13. Запись в правила прокси

```
insert into mysql_query_rules(rule_id, active, match_digest,
destination_hostgroup, mirror_hostgroup, apply) values(1, 1, '.', 1, 2, 1);
```

14. Загрузка конфигурации

```
load mysql query rules to runtime;
save mysql query rules to disk;
```

Управление репликами

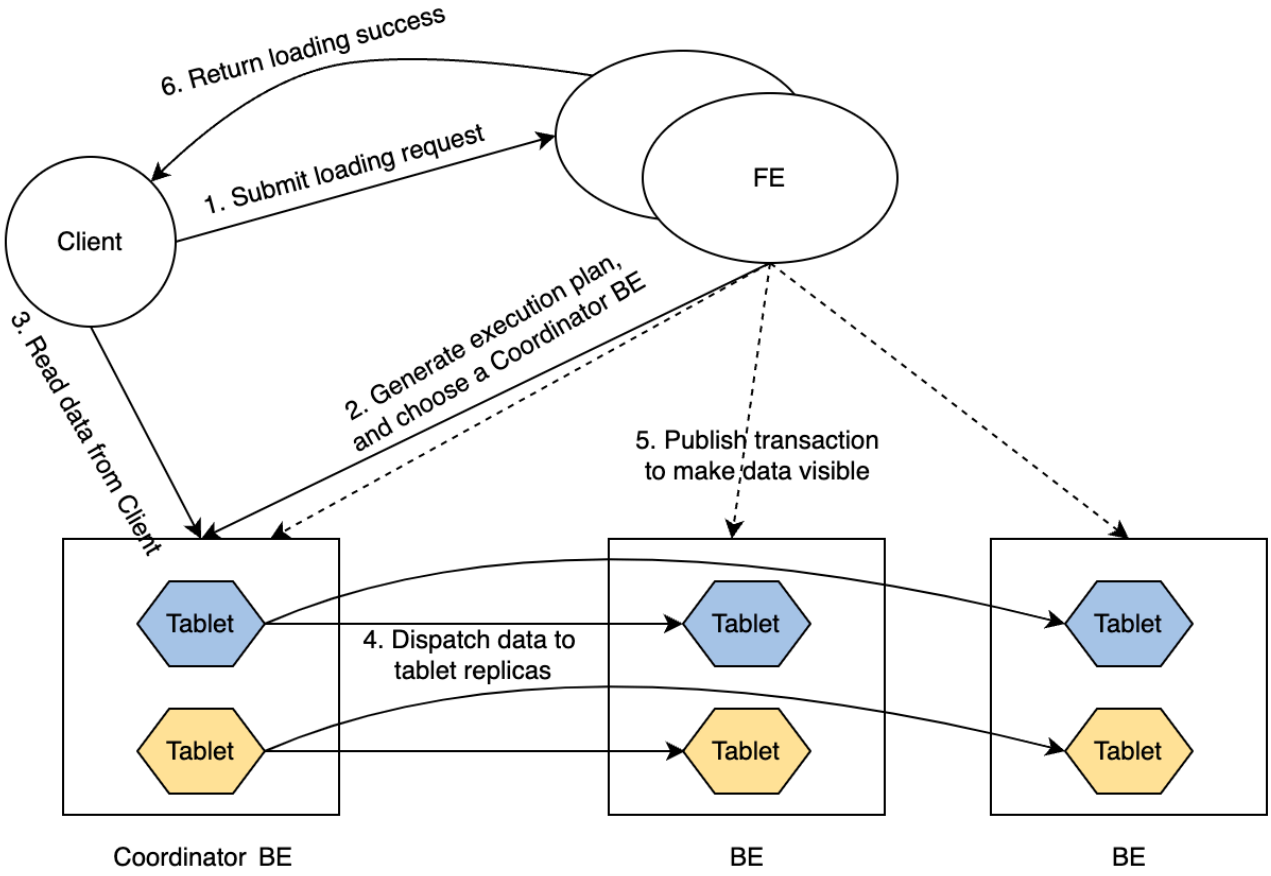
В этом разделе описывается, как управлять репликами данных в вашем cluster Selena.

Обзор

Selena использует стратегию множественных replica для гарантии высокой доступности данных. При создании таблицы необходимо указать количество replica с помощью свойства таблицы `replication_num` (значение по умолчанию: `3`). Когда начинается транзакция загрузки, данные одновременно загружаются в указанное количество replica. Транзакция возвращается с успехом только после того, как данные сохранены в большинстве replica. Подробную информацию см. в разделе Кворум записи. Тем не менее, Selena позволяет указать более низкий кворум записи для таблицы для достижения лучшей производительности загрузки.

Selena хранит несколько replica на разных узлах BE. Например, если вы хотите хранить три replica для таблицы, вы должны развернуть как минимум три узла BE в вашем cluster Selena. Если какая-либо из replica выходит из строя, Selena клонирует здоровую replica, частично или полностью, с другого узла BE для восстановления неисправной replica. Используя технику Multi-Version Concurrency Control (MVCC), Selena ускоряет восстановление replica путём дублирования физических копий этих многоверсионных данных.

Загрузка данных в таблицу с множественными репликами



Процедура транзакции загрузки следующая:

1. Клиент отправляет запрос на загрузку в FE.
2. FE выбирает узел BE в качестве координатора BE для этой транзакции загрузки и генерирует план выполнения для транзакции.
3. Координатор BE читает данные для загрузки от клиента.
4. Координатор BE распределяет данные по всем replica tablet.

> **ПРИМЕЧАНИЕ**
>
> Tablet — это логический фрагмент таблицы. Таблица имеет несколько tablet, и каждый tablet имеет replication_num replica. Количество tablet в таблице определяется свойством `bucket_size` таблицы.

5. После того как данные загружены и сохранены во всех tablet, FE делает загруженные данные видимыми.

6. FE возвращает клиенту успешное завершение загрузки.

Такая процедура гарантирует доступность сервиса даже в экстремальных сценариях.

Кворум записи

Загрузка данных в таблицу с множественными replica может занять много времени. Если вы хотите улучшить производительность загрузки и можете допустить относительно более низкую доступность данных, вы можете установить более низкий кворум записи для таблиц. Кворум записи относится к минимальному количеству replica, которые должны подтвердить операцию записи, прежде чем она будет считаться успешной. Вы можете указать кворум записи, добавив свойство `write_quorum` при CREATE TABLE или добавить это свойство к существующей таблице с помощью ALTER TABLE. Это свойство поддерживается с версии v1.5.2.

`write_quorum` поддерживает следующие значения:

- `MAJORITY`: Значение по умолчанию. Когда большинство replica данных возвращают успех загрузки, Selena возвращает успех задачи загрузки. В противном случае Selena возвращает неудачу задачи загрузки.
- `ONE`: Когда одна из replica данных возвращает успех загрузки, Selena возвращает успех задачи загрузки. В противном случае Selena возвращает неудачу задачи загрузки.
- `ALL`: Когда все replica данных возвращают успех загрузки, Selena возвращает успех задачи загрузки. В противном случае Selena возвращает неудачу задачи загрузки.

Автоматическое восстановление реплик

Replica могут выходить из строя из-за сбоя некоторых узлов BE или неудачи некоторых задач загрузки. Selena автоматически восстанавливает эти неисправные replica.

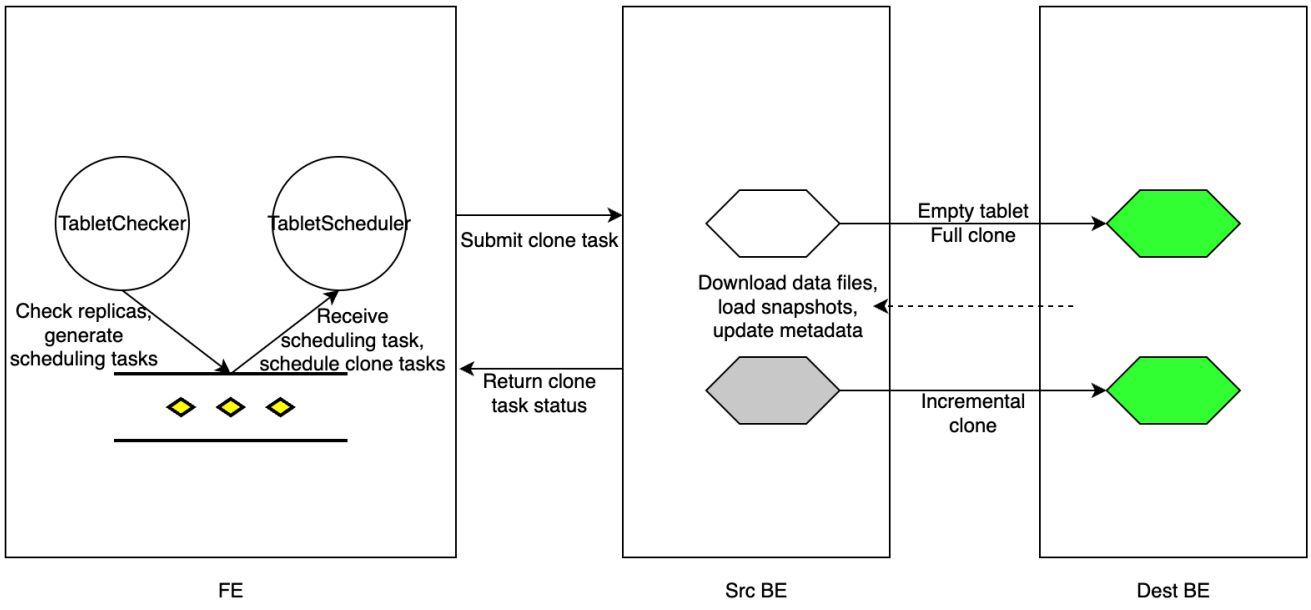
Каждые `tablet_sched_checker_interval_seconds` (по умолчанию 20 секунд) Tablet Checker в FE сканирует все replica tablet всех таблиц в вашем cluster Selena и определяет, является ли replica здоровой, проверяя номер версии текущих видимых данных и состояние здоровья узла BE. Если видимая версия replica отстаёт от версий других replica, Selena выполняет инкрементное клонирование для восстановления неисправной replica. Если узел BE не получает heartbeat или удалён из cluster, или replica слишком

сильно отстаёт, чтобы её можно было восстановить инкрементным клонированием, Selena выполняет полное клонирование для восстановления потерянной replica.

После обнаружения replica tablet, требующих восстановления, FE генерирует задачу планирования tablet и добавляет задачу в очередь задач планирования. Tablet Scheduler в FE получает задачу планирования из очереди, создаёт задачи клонирования для каждой неисправной replica в соответствии с типом клонирования, который им нужен, и назначает задачи исполнительным узлам BE.

Задача клонирования по сути представляет собой копирование данных с исходного узла BE (на котором находится здоровая replica) и загрузку данных на целевой узел BE (на котором находится неисправная replica). Для replica с отстающей версией данных FE назначает задачу инкрементного клонирования исполнителю BE, который хранит неисправную replica, и сообщает исполнительному узлу BE, с какого однорангового узла BE он может найти здоровую replica и клонировать новые данные. Если replica потеряна, FE выбирает выживший узел BE в качестве исполнительного узла BE, создаёт пустую replica на узле BE и назначает задачу полного клонирования узлу BE.

Для каждой задачи клонирования, независимо от её типа, исполнительный узел BE дублирует физические файлы данных со здоровой replica, а затем соответствующим образом обновляет свои метаданные. После завершения задачи клонирования исполнительный узел BE сообщает об успехе задачи в Tablet Scheduler в FE. После удаления избыточных replica tablet FE обновляет свои метаданные, отмечая завершение восстановления replica.



Во время восстановления tablet Selena по-прежнему может выполнять запросы. Selena может загружать данные в таблицу, пока количество здоровых replica удовлетворяет `write_quorum`.

Ручное восстановление реплик

Ручное восстановление replica состоит из двух шагов:

1. Проверка статуса replica.
2. Установка уровня приоритета replica.

Проверка статуса реплик

Следуйте этим шагам, чтобы проверить статус replica tablet для идентификации нездоровых (неисправных) tablet.

1. **Проверьте статус всех tablet в cluster.**

   ```
   SHOW PROC '/statistic';
   ```

   Пример:

```
mysql> SHOW PROC '/statistic';
+----------+-----------------------------+----------+--------------+---
-------+-----------+------------+--------------------+-----------------
------+
| DbId     | DbName                      | TableNum | PartitionNum |
IndexNum | TabletNum | ReplicaNum | UnhealthyTabletNum |
InconsistentTabletNum |
+----------+-----------------------------+----------+--------------+---
-------+-----------+------------+--------------------+-----------------
------+
| 35153636 | default_cluster:DF_Newrisk  | 3        | 3            | 3
| 96        | 288        | 0                  | 0                    |
| 48297972 | default_cluster:PaperData   | 0        | 0            | 0
| 0         | 0          | 0                  | 0                    |
| 5909381  | default_cluster:UM_TEST     | 7        | 7            | 10
| 320       | 960        | 1                  | 0                    |
| Total    | 240                         | 10       | 10           | 13
| 416       | 1248       | 1                  | 0                    |
+----------+-----------------------------+----------+--------------+---
-------+-----------+------------+--------------------+-----------------
------+
```

- `UnhealthyTabletNum`: указывает количество нездоровых tablet в соответствующей базе данных.
- `InconsistentTabletNum`: указывает количество tablet, replica которых несогласованы.

Если значение `UnhealthyTabletNum` или `InconsistentTabletNum` не равно `0` в конкретной базе данных, вы можете проверить нездоровые tablet в базе данных по её `DbId`.

```
SHOW PROC '/statistic/<DbId>'
```

Пример:

```
mysql> SHOW PROC '/statistic/5909381';
+------------------+---------------------+
| UnhealthyTablets | InconsistentTablets |
+------------------+---------------------+
| [40467980]       | []                  |
+------------------+---------------------+
```

ID нездорового tablet возвращается в поле `UnhealthyTablets`.

2. **Проверьте статус tablet в конкретной таблице или partition.**

   Вы можете использовать предложение WHERE в ADMIN SHOW REPLICA STATUS для фильтрации tablet с определённым `STATUS`.

```
ADMIN SHOW REPLICA STATUS FROM <table_name>
[PARTITION (<partition_name_1>[, <partition_name_2>, ...])]
[WHERE STATUS = {'OK'|'DEAD'|'VERSION_ERROR'|'SCHEMA_ERROR'|'MISSING'}]
```

   Пример:

```
mysql> ADMIN SHOW REPLICA STATUS FROM tbl PARTITION (p1, p2) WHERE
STATUS = "OK";
+----------+-----------+-----------+---------+-------------------+-----
---------------+------------------+------------+------------+-------+--
------+--------+
| TabletId | ReplicaId | BackendId | Version | LastFailedVersion |
LastSuccessVersion | CommittedVersion | SchemaHash | VersionNum | IsBad
| State  | Status |
+----------+-----------+-----------+---------+-------------------+-----
---------------+------------------+------------+------------+-------+--
------+--------+
| 29502429 | 29502432  | 10006     | 2       | -1                | 2
| 1                | -1         | 2          | false | NORMAL | OK
|
| 29502429 | 36885996  | 10002     | 2       | -1                | -1
| 1                | -1         | 2          | false | NORMAL | OK
|
| 29502429 | 48100551  | 10007     | 2       | -1                | -1
| 1                | -1         | 2          | false | NORMAL | OK
|
| 29502433 | 29502434  | 10001     | 2       | -1                | 2
| 1                | -1         | 2          | false | NORMAL | OK
|
| 29502433 | 44900737  | 10004     | 2       | -1                | -1
| 1                | -1         | 2          | false | NORMAL | OK
|
| 29502433 | 48369135  | 10006     | 2       | -1                | -1
| 1                | -1         | 2          | false | NORMAL | OK
|
+----------+-----------+-----------+---------+-------------------+-----
---------------+------------------+------------+------------+-------+--
------+--------+
```

Если поле `IsBad` равно `true`, этот tablet повреждён.

Подробную информацию, предоставляемую в поле `Status`, см. в ADMIN SHOW REPLICA STATUS.

Вы можете дополнительно изучить детали tablet в таблице с помощью SHOW TABLET.

```
SHOW TABLET FROM <table_name>
```

Пример:

```
mysql> SHOW TABLET FROM tbl1;
+----------+-----------+-----------+------------+---------+------------
-+-------------------+----------------------+------------------+------
----------------+---------------+----------+----------+--------+-------
------------------+--------------+----------------------+--------------
+----------------------+----------------------+----------------------+
| TabletId | ReplicaId | BackendId | SchemaHash | Version | VersionHash
| LstSuccessVersion | LstSuccessVersionHash | LstFailedVersion |
LstFailedVersionHash | LstFailedTime | DataSize | RowCount | State  |
LstConsistencyCheckTime | CheckVersion |     CheckVersionHash |
VersionCount | PathHash             | MetaUrl              |
CompactionStatus     |
+----------+-----------+-----------+------------+---------+------------
-+-------------------+----------------------+------------------+------
----------------+---------------+----------+----------+--------+-------
------------------+--------------+----------------------+--------------
+----------------------+----------------------+----------------------+
| 29502429 | 29502432  | 10006     | 1421156361 | 2       | 0
| 2                 | 0                    | -1               | 0
| N/A           | 784      | 0        | NORMAL | N/A
| -1           |     -1               | 2            |
-5822326203532286804 | url                  | url                  |
| 29502429 | 36885996  | 10002     | 1421156361 | 2       | 0
| -1                | 0                    | -1               | 0
| N/A           | 784      | 0        | NORMAL | N/A
| -1           |     -1               | 2            |
-1441285706148429853 | url                  | url                  |
| 29502429 | 48100551  | 10007     | 1421156361 | 2       | 0
| -1                | 0                    | -1               | 0
| N/A           | 784      | 0        | NORMAL | N/A
| -1           |     -1               | 2            |
-4784691547051455525 | url                  | url                  |
+----------+-----------+-----------+------------+---------+------------
-+-------------------+----------------------+------------------+------
----------------+---------------+----------+----------+--------+-------
------------------+--------------+----------------------+--------------
+----------------------+----------------------+----------------------+
```

Возвращаемые результаты показывают размер, количество строк, версию и URL tablet.

Поле `State`, возвращаемое SHOW TABLET, указывает состояние задачи tablet, включая `CLONE`, `SCHEMA_CHANGE` и `ROLLUP`.

Вы также можете проверить распределение replica конкретной таблицы или partition, чтобы увидеть, равномерно ли распределены эти replica, используя ADMIN SHOW REPLICA DISTRIBUTION.

```
ADMIN SHOW REPLICA DISTRIBUTION FROM <table_name>
```

Пример:

```
mysql> ADMIN SHOW REPLICA DISTRIBUTION FROM tbl1;
+-----------+------------+-------+---------+
| BackendId | ReplicaNum | Graph | Percent |
+-----------+------------+-------+---------+
| 10000     | 7          |       | 7.29 %  |
| 10001     | 9          |       | 9.38 %  |
| 10002     | 7          |       | 7.29 %  |
| 10003     | 7          |       | 7.29 %  |
| 10004     | 9          |       | 9.38 %  |
| 10005     | 11         | >     | 11.46 % |
| 10006     | 18         | >     | 18.75 % |
| 10007     | 15         | >     | 15.62 % |
| 10008     | 13         | >     | 13.54 % |
+-----------+------------+-------+---------+
```

Возвращаемые результаты показывают количество replica tablet на каждом узле BE и их соответствующие процентные доли.

3. **Проверьте статус replica конкретного tablet.**

С помощью `TabletId` нездоровых tablet, полученных в предыдущих процедурах, вы можете изучить статусы replica.

```
SHOW TABLET <TabletId>
```

Пример:

```
mysql> SHOW TABLET 29502553;
+-------------------------+-----------+---------------+-----------+-----
-----+----------+-------------+----------+--------+--------------------
-------------------------------------------------------+
| DbName                  | TableName | PartitionName | IndexName | DbId
| TableId  | PartitionId | IndexId  | IsSync | DetailCmd
|
+-------------------------+-----------+---------------+-----------+-----
-----+----------+-------------+----------+--------+--------------------
-------------------------------------------------------+
| default_cluster:test    | test      | test          | test      |
29502391 | 29502428 | 29502427    | 29502428 | true   | SHOW PROC
'/dbs/29502391/29502428/partitions/29502427/29502428/29502553'; |
+-------------------------+-----------+---------------+-----------+-----
-----+----------+-------------+----------+--------+--------------------
-------------------------------------------------------+
```

Возвращаемые результаты показывают подробную информацию о базе данных, таблице, partition и индексе (Rollup) tablet.

Вы можете скопировать SQL-выражение из поля `DetailCmd` для дальнейшего изучения статуса replica tablet.

Пример:

```
mysql> SHOW PROC
'/dbs/29502391/29502428/partitions/29502427/29502428/29502553';
+-----------+-----------+---------+-------------+-------------------+--
---------------------+------------------+----------------------+-------
--------+------------+----------+----------+--------+-------+----------
----+----------------------+----------+------------------+
| ReplicaId | BackendId | Version | VersionHash | LstSuccessVersion |
LstSuccessVersionHash | LstFailedVersion | LstFailedVersionHash |
LstFailedTime | SchemaHash | DataSize | RowCount | State  | IsBad |
VersionCount | PathHash             | MetaUrl  | CompactionStatus |
+-----------+-----------+---------+-------------+-------------------+--
---------------------+------------------+----------------------+-------
--------+------------+----------+----------+--------+-------+----------
----+----------------------+----------+------------------+
| 43734060  | 10004     | 2       | 0           | -1                | 0
| -1               | 0                    | N/A           | -1
| 784      | 0        | NORMAL | false | 2            |
-8566523878520798656 | url      | url              |
| 29502555  | 10002     | 2       | 0           | 2                 | 0
| -1               | 0                    | N/A           | -1
| 784      | 0        | NORMAL | false | 2            |
1885826196444191611  | url      | url              |
| 39279319  | 10007     | 2       | 0           | -1                | 0
| -1               | 0                    | N/A           | -1
| 784      | 0        | NORMAL | false | 2            |
1656508631294397870  | url      | url              |
+-----------+-----------+---------+-------------+-------------------+--
---------------------+------------------+----------------------+-------
--------+------------+----------+----------+--------+-------+----------
----+----------------------+----------+------------------+
```

Возвращаемые результаты показывают все replica tablet.

Установка уровня приоритета реплик

Tablet Scheduler автоматически назначает разный уровень приоритета каждому типу задачи клонирования.

Если вы хотите, чтобы tablet из определённой таблицы или определённых partition были восстановлены в первую очередь, вы можете вручную назначить им уровень приоритета `VERY_HIGH` с помощью ADMIN REPAIR TABLE.

```
ADMIN REPAIR TABLE <table_name>
[PARTITION (<partition_name_1>[, <partition_name_2>, ...])]
```

> **ПРИМЕЧАНИЕ**
>
> - Выполнение этого SQL-выражения только отправляет подсказку для изменения уровня приоритета tablet, которые должны быть восстановлены. Это не гарантирует, что эти tablet будут успешно восстановлены.
> - Tablet Scheduler может по-прежнему назначать разные уровни приоритета этим tablet после выполнения этого SQL-выражения.
> - Когда Leader FE узел изменяется или перезапускается, подсказка, отправленная этим SQL-выражением, истекает.

Вы можете отменить эту операцию с помощью ADMIN CANCEL REPAIR TABLE.

```
ADMIN CANCEL REPAIR TABLE <table_name>
[PARTITION (<partition_name_1>[, <partition_name_2>, ...])]
```

Балансировка реплик

Selena автоматически балансирует tablet между узлами BE.

Чтобы переместить tablet с узла с высокой нагрузкой на узел с низкой нагрузкой, Selena сначала создаёт replica tablet на узле с низкой нагрузкой, а затем удаляет соответствующую replica на узле с высокой нагрузкой. Если в cluster используются разные типы носителей данных, Selena категоризирует все узлы BE в соответствии с типами носителей данных. Selena перемещает tablet между узлами BE одного типа носителя данных, когда это возможно. Replica одного tablet хранятся на разных узлах BE.

Нагрузка BE

Selena показывает статистику нагрузки каждого узла BE в cluster с помощью `ClusterLoadStatistics` (CLS). Tablet Scheduler запускает балансировку replica на основе `ClusterLoadStatistics`. Selena оценивает **использование диска** и **количество replica** каждого узла BE и вычисляет `loadScore` соответственно. Чем выше `loadScore` узла BE, тем выше нагрузка на узел. Tablet Scheduler обновляет `ClusterLoadStatistics` каждую минуту.

`capacityCoefficient` и `replicaNumCoefficient` — это весовые коэффициенты для использования диска и количества replica соответственно. Сумма `capacityCoefficient` и `replicaNumCoefficient` равна единице. `capacityCoefficient` динамически регулируется в соответствии с фактическим использованием диска. Когда общее использование диска узла BE ниже 50%, значение `capacityCoefficient` равно 0.5. Когда использование диска выше 75%, значение равно 1. Вы можете настроить этот лимит через элемент конфигурации FE `capacity_used_percent_high_water`. Если использование находится между 50% и 75%, `capacityCoefficient` увеличивается плавно на основе этой формулы:

```
capacityCoefficient= 2 * Использование диска - 0.5
```

`capacityCoefficient` гарантирует, что когда использование диска чрезмерно высоко, `loadScore` этого узла BE становится выше, заставляя систему снизить нагрузку на этот узел BE в кратчайшие сроки.

Политика балансировки

Каждый раз, когда Tablet Scheduler планирует tablet, он выбирает определённое количество здоровых tablet в качестве кандидатов для балансировки через Load Balancer. В следующий раз при планировании tablet Tablet Scheduler балансирует эти здоровые tablet.

Просмотр статуса балансировки системы

Вы можете просмотреть текущий общий статус балансировки системы и детали различных типов балансировки.

- **Просмотр текущего общего статуса балансировки системы:**

  ```
  SHOW PROC '/cluster_balance/balance_stat';
  ```

  Пример:

```
+---------------+--------------------------------+----------+----------
------+----------------+
| StorageMedium | BalanceType                    | Balanced |
PendingTablets | RunningTablets |
+---------------+--------------------------------+----------+----------
------+----------------+
| HDD           | inter-node disk usage          | true     | 0
| 0              |
| HDD           | inter-node tablet distribution | true     | 0
| 0              |
| HDD           | intra-node disk usage          | true     | 0
| 0              |
| HDD           | intra-node tablet distribution | true     | 0
| 0              |
| HDD           | colocation group               | true     | 0
| 0              |
| HDD           | label-aware location           | true     | 0
| 0              |
+---------------+--------------------------------+----------+----------
------+----------------+
```

- `StorageMedium`: Носитель данных.
- `BalanceType`: Тип балансировки.
- `Balanced`: Достигнуто ли сбалансированное состояние.
- `PendingTablets`: Количество tablet со статусом задачи Pending.
- `RunningTablets`: Количество tablet со статусом задачи Running.

- **Просмотр баланса использования диска по узлам:**

```
SHOW PROC '/cluster_balance/cluster_load_stat';
```

Пример:

```
+---------------+-------------------------------------------------------
--------------------+
| StorageMedium | ClusterDiskBalanceStat
|
+---------------+-------------------------------------------------------
--------------------+
| HDD           |
{"balanced":false,"maxBeId":1,"minBeId":2,"maxUsedPercent":0.9,"minUsedPe
|
| SSD           | {"balanced":true}
|
+---------------+-------------------------------------------------------
--------------------+
```

- `StorageMedium`: Носитель данных.
- `ClusterDiskBalanceStat`: Статус балансировки между узлами на основе использования диска. Если не сбалансировано, отображает максимальное и минимальное использование диска и соответствующие BE.

- **Просмотр баланса использования диска внутри узла:**

```
SHOW PROC '/cluster_balance/cluster_load_stat/HDD';
```

Пример:

```
+-------+-----------------+-----------+--------------+--------------+----
-------------------------------------------------------------------------
+
| BeId  | Cluster         | Available | UsedCapacity | Capacity     | Use
BackendDiskBalanceStat
|
+-------+-----------------+-----------+--------------+--------------+----
-------------------------------------------------------------------------
+
| 10004 | default_cluster | true      | 651509602    | 243695955810 | 0.2
{"maxUsedPercent":0.9,"minUsedPercent":0.1,"beId":1,"maxPath":"/disk1","m
|
| 10005 | default_cluster | true      | 651509602    | 243695955810 | 0.2
{"balanced":true}
|
+-------+-----------------+-----------+--------------+--------------+----
-------------------------------------------------------------------------
+
```

- `BeId`: ID узла BE.
- `BackendDiskBalanceStat`: Статус балансировки между дисками внутри узла на основе использования диска. Если не сбалансировано, отображает максимальное и минимальное использование диска и соответствующие пути к дискам.

- **Просмотр сбалансированного распределения по tablet:**

```
SHOW PROC '/dbs/ssb/lineorder/partitions/lineorder';
```

Пример:

```
+---------+-----------+--------+-------------------------+--------------
-------------------------------------------------------------+
| IndexId | IndexName | State  | LastConsistencyCheckTime | TabletBalance
|
+---------+-----------+--------+-------------------------+--------------
-------------------------------------------------------------+
| 11129   | lineorder | NORMAL | NULL                     |
{"maxTabletNum":23,"minTabletNum":21,"maxBeId":10012,"minBeId":10013,"typ
|
| 11230   | lineorder | NORMAL | NULL                     |
{"maxTabletNum":23,"minTabletNum":21,"beId":10012,"maxPath":"/disk1","min
|
| 10432   | lineorder | NORMAL | NULL                     | {"tabletId":1
[10003,10004],"type":"COLOCATION_GROUP","balanced":false}
| 10436   | lineorder | NORMAL | NULL                     | {"tabletId":1
["rack1","rack2"]},"type":"LABEL_AWARE_LOCATION","balanced":false}
+---------+-----------+--------+-------------------------+--------------
-------------------------------------------------------------+
```

- `IndexId`: ID материализованного индекса внутри partition.
- `TabletBalanceStat`: Статус балансировки распределения tablet между узлами или внутри узлов. Если не сбалансировано, отображает детали дисбаланса, включая Colocation Group, Label-aware Location.

- **Просмотр partition с несбалансированным распределением tablet:**

```
SELECT DB_NAME, TABLE_NAME, PARTITION_NAME, TABLET_BALANCED FROM
information_schema.partitions_meta WHERE TABLET_BALANCED = 0;
```

Пример:

```
+--------------+---------------+----------------+-----------------+
| DB_NAME      | TABLE_NAME    | PARTITION_NAME | TABLET_BALANCED |
+--------------+---------------+----------------+-----------------+
| ssb          | lineorder     | lineorder      |               0 |
+--------------+---------------+----------------+-----------------+
```

- `TABLET_BALANCED`: сбалансировано ли распределение tablet.

Проверка задач планирования tablet

Вы можете проверить задачи планирования tablet, которые находятся в ожидании, выполняются и завершены.

- **Проверка ожидающих задач планирования tablet**

```
SHOW PROC '/cluster_balance/pending_tablets';
```

Пример:

```
+----------+--------+-----------------+---------+----------+----------
+-------+---------+--------+----------+---------+--------------------
+---------------------+---------------------+----------+------+--------
-----+---------------+---------------------+------------+--------------
-------+--------+---------------------+-------------------------------+
| TabletId | Type   | Status          | State   | OrigPrio | DynmPrio |
SrcBe | SrcPath | DestBe | DestPath | Timeout | Create              |
LstSched            | LstVisit            | Finished | Rate |
FailedSched | FailedRunning | LstAdjPrio          | VisibleVer |
VisibleVerHash      | CmtVer | CmtVerHash          | ErrMsg
|
+----------+--------+-----------------+---------+----------+----------
+-------+---------+--------+----------+---------+--------------------
+---------------------+---------------------+----------+------+--------
-----+---------------+---------------------+------------+--------------
-------+--------+---------------------+-------------------------------+
| 4203036  | REPAIR | REPLICA_MISSING | PENDING | HIGH     | LOW      |
-1    | -1      | -1     | -1       | 0       | 2019-02-21 15:00:20 |
2019-02-24 11:18:41 | 2019-02-24 11:18:41 | N/A      | N/A  | 2
| 0             | 2019-02-21 15:00:43 | 1          | 0
| 2      | 0                   | unable to find source replica |
+----------+--------+-----------------+---------+----------+----------
+-------+---------+--------+----------+---------+--------------------
+---------------------+---------------------+----------+------+--------
-----+---------------+---------------------+------------+--------------
-------+--------+---------------------+-------------------------------+
```

- `TabletId`: ID tablet, ожидающего планирования. Запланированная задача только для одного tablet.
- `Type`: тип задачи. Допустимые значения: REPAIR и BALANCE.
- `Status`: текущий статус tablet, например REPLICA_MISSING.

- `State`: состояние задачи планирования. Допустимые значения: PENDING, RUNNING, FINISHED, CANCELLED, TIMEOUT и UNEXPECTED.
- `OrigPrio`: исходный приоритет задачи.
- `DynmPrio`: текущий приоритет задачи после динамической корректировки.
- `SrcBe`: ID исходного узла BE.
- `SrcPath`: хеш-значение пути к исходному узлу BE.
- `DestBe`: ID целевого узла BE.
- `DestPath`: хеш-значение пути к целевому узлу BE.
- `Timeout`: тайм-аут задачи при успешном планировании. Единица измерения: секунды.
- `Create`: время создания задачи.
- `LstSched`: время последнего планирования задачи.
- `LstVisit`: время последнего посещения задачи. Посещение задачи здесь означает планирование задачи или отчёт о её выполнении.
- `Finished`: время завершения задачи.
- `Rate`: скорость клонирования данных.
- `FailedSched`: количество неудачных планирований задачи.
- `FailedRunning`: количество неудачных выполнений задачи.
- `LstAdjPrio`: время последней корректировки приоритета задачи.
- `CmtVer`, `CmtVerHash`, `VisibleVer` и `VisibleVerHash`: информация о версии, используемая для выполнения задачи клонирования.
- `ErrMsg`: сообщение об ошибке, возникающее при планировании и выполнении задачи.

- **Проверка выполняющихся задач планирования tablet**

```
SHOW PROC '/cluster_balance/running_tablets';
```

Возвращаемые результаты идентичны результатам для ожидающих задач.

- **Проверка завершённых задач планирования tablet**

```
SHOW PROC '/cluster_balance/history_tablets';
```

Возвращаемые результаты идентичны результатам для ожидающих задач. Если `State` задачи — `FINISHED`, задача успешно завершена. Если нет, пожалуйста, проверьте поле `ErrMsg` для определения причины неудачи задачи.

Управление ресурсами

Поскольку Selena восстанавливает и балансирует tablet путём клонирования tablet с одного узла BE на другой, нагрузка ввода-вывода узла BE может резко возрасти, если узел выполняет такие задачи слишком часто за короткое время. Чтобы избежать этой ситуации, Selena устанавливает лимит параллелизма для задач клонирования для каждого узла BE. Минимальная единица управления ресурсами — это диск, который представляет собой путь хранения данных (`storage_root_path`), указанный вами в файле конфигурации BE. По умолчанию Selena выделяет два слота для каждого диска для обработки задач восстановления tablet. Задача клонирования занимает один слот на исходном узле BE и один на целевом узле BE. Если все слоты на узле BE заняты, Selena прекращает планирование задач на этот узел. Вы можете увеличить количество слотов на узле BE, увеличив значение динамического параметра FE `tablet_sched_slot_num_per_path`.

Selena выделяет два слота специально для задач балансировки tablet, чтобы избежать ситуации, когда узел BE с высокой нагрузкой не может освободить дисковое пространство путём балансировки tablet, потому что задачи восстановления tablet постоянно занимают слоты.

Управление чёрным списком

В некоторых случаях администраторам необходимо запретить определённые шаблоны SQL, чтобы избежать сбоев cluster или неожиданных высоконагруженных запросов. Чёрный список применяется только к операторам SELECT, операторам INSERT (начиная с версии 3.1) и операторам CTAS (начиная с версии 3.4).

Selena позволяет пользователям добавлять, просматривать и удалять чёрные списки SQL.

Синтаксис

Включите чёрный список SQL с помощью `enable_sql_blacklist`. По умолчанию значение False (выключено).

```
admin set frontend config ("enable_sql_blacklist" = "true")
```

Пользователь с правами администратора (ADMIN_PRIV) может управлять чёрными списками, выполняя следующие команды:

```
ADD SQLBLACKLIST "<sql>"
DELETE SQLBLACKLIST <sql_index_number>
SHOW SQLBLACKLIST
```

- Когда `enable_sql_blacklist` имеет значение true, каждый SQL-запрос должен быть проверен по sqlblacklist. Если он совпадает, пользователь будет уведомлён, что SQL находится в чёрном списке. В противном случае SQL будет выполнен нормально. Сообщение может быть следующим, когда SQL находится в чёрном списке:

`ERROR 1064 (HY000): Access denied; sql 'select count (*) from test_all_type_select_2556' is in blacklist`

Добавление в чёрный список

```
ADD SQLBLACKLIST "<sql>"
```

sql - это регулярное выражение для определённого типа SQL.

> **ПОДСКАЗКА**
>
> В настоящее время Selena поддерживает добавление операторов SELECT в чёрный список SQL.

Поскольку сам SQL содержит общие символы `(`, `)`, `*`, `.`, которые могут быть перепутаны с семантикой регулярных выражений, нам нужно их различать с помощью escape-символов. Учитывая, что `(` и `)` используются очень часто в SQL, для них не нужно использовать escape-символы. Другие специальные символы требуют использования escape-символа `\` в качестве префикса. Например:

- Запретить `count(\*)`:

```
ADD SQLBLACKLIST "select count(\\*) from .+"
```

- Запретить `count(distinct)`:

```
ADD SQLBLACKLIST "select count(distinct .+) from .+"
```

- Запретить order by limit `x`, `y`, `1 <= x <=7`, `5 <=y <=7`:

```
ADD SQLBLACKLIST "select id_int from test_all_type_select1 order by id_int
limit [1-7], [5-7]"
```

- Запретить сложный SQL:

```
ADD SQLBLACKLIST "select id_int \\* 4, id_tinyint, id_varchar from
test_all_type_nullable except select id_int, id_tinyint, id_varchar from
test_basic except select (id_int \\* 9 \\- 8) \\/ 2, id_tinyint, id_varchar
from test_all_type_nullable2 except select id_int, id_tinyint, id_varchar
from test_basic_nullable"
```

- Запретить все операторы INSERT INTO:

```
ADD SQLBLACKLIST "(?i)^insert\\s+into\\s+.*";
```

- Запретить все операторы INSERT INTO ... VALUES:

```
ADD SQLBLACKLIST "(?i)^insert\\s+into\\s+.*values\\s*\\(";
```

- Запретить все операторы INSERT INTO ... VALUES, кроме тех, которые направлены на системное представление `_statistics_.column_statistics`:

```
ADD SQLBLACKLIST "(?i)^insert\\s+into\\s+
(?!column_statistics\\b).*values\\s*\\(";
```

Просмотр чёрного списка

```
SHOW SQLBLACKLIST
```

Формат результата: `Index | Forbidden SQL`

Например:

```
mysql> show sqlblacklist;
+-------+-------------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----+
| Index | Forbidden SQL
|
+-------+-------------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----+
| 1     | select count\(\*\) from .+
|
| 2     | select id_int \* 4, id_tinyint, id_varchar from
test_all_type_nullable except select id_int, id_tinyint, id_varchar from
test_basic except select \(id_int \* 9 \- 8\) \/ 2, id_tinyint, id_varchar
from test_all_type_nullable2 except select id_int, id_tinyint, id_varchar
from test_basic_nullable |
| 3     | select id_int from test_all_type_select1 order by id_int limit
[1-7], [5-7]
|
| 4     | select count\(distinct .+\) from .+
|
+-------+-------------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----+
```

SQL, показанный в `Forbidden SQL`, экранирован для всех семантических символов SQL.

Удаление из чёрного списка

```
DELETE SQLBLACKLIST <sql_index_number>
```

`<sql_index_number>` - это список идентификаторов SQL, разделённых запятой (,).

Например, удалим SQL №3 и №4 в приведённом выше чёрном списке:

```
delete sqlblacklist  3, 4;
```

Тогда оставшийся sqlblacklist будет следующим:

```
mysql> show sqlblacklist;
+-------+-----------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----+
| Index | Forbidden SQL
|
+-------+-----------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----+
| 1     | select count\(\*\) from .+
|
| 2     | select id_int \* 4, id_tinyint, id_varchar from
test_all_type_nullable except select id_int, id_tinyint, id_varchar from
test_basic except select \(id_int \* 9 \- 8\) \/ 2, id_tinyint, id_varchar
from test_all_type_nullable2 except select id_int, id_tinyint, id_varchar
from test_basic_nullable |
+-------+-----------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----+
```

Добавление меток на BE

Начиная с версии 3.2.8, Selena поддерживает добавление меток на BE. При создании таблицы или асинхронного материализованного представления вы можете указать метку определённой группы узлов BE. Это гарантирует, что replica данных будут распределены только на узлах BE, связанных с этой меткой. Replica данных будут равномерно распределены между узлами с одинаковой меткой, что повышает высокую доступность данных и изоляцию ресурсов.

Использование

Добавление меток на BE

Предположим, что один cluster Selena включает шесть BE, которые равномерно распределены по трём стойкам. Вы можете добавить метки на BE в зависимости от стоек, где расположены BE.

```
ALTER SYSTEM MODIFY BACKEND "172.xx.xx.46:9050" SET ("labels.location" =
"rack:rack1");
ALTER SYSTEM MODIFY BACKEND "172.xx.xx.47:9050" SET ("labels.location" =
"rack:rack1");
ALTER SYSTEM MODIFY BACKEND "172.xx.xx.48:9050" SET ("labels.location" =
"rack:rack2");
ALTER SYSTEM MODIFY BACKEND "172.xx.xx.49:9050" SET ("labels.location" =
"rack:rack2");
ALTER SYSTEM MODIFY BACKEND "172.xx.xx.50:9050" SET ("labels.location" =
"rack:rack3");
ALTER SYSTEM MODIFY BACKEND "172.xx.xx.51:9050" SET ("labels.location" =
"rack:rack3");
```

После добавления меток вы можете выполнить `SHOW BACKENDS;` и просмотреть метки BE в поле `Location` возвращаемого результата.

Если вам нужно изменить метки BE, вы можете выполнить `ALTER SYSTEM MODIFY BACKEND "172.xx.xx.48:9050" SET ("labels.location" = "rack:xxx");`.

Использование меток для указания распределения данных таблицы на узлах BE

Если вам нужно указать места, куда будут распределены данные таблицы, например, распределение данных таблицы по двум стойкам, rack1 и rack2, вы можете добавить метки к таблице.

После добавления меток все replica одного tablet в таблице распределяются по меткам по принципу Round-Robin. Более того, если несколько replica одного tablet существуют в рамках одной метки, эти replica будут распределены как можно более равномерно между разными BE в этой метке.

> **ПРИМЕЧАНИЕ**
>
> - Если общее количество узлов BE, связанных с метками, меньше количества replica, система в первую очередь обеспечит достаточное количество replica. В этом случае replica могут быть распределены не так, как указано в метке.
> - Метка, которую нужно связать с таблицей, должна уже существовать. В противном случае произойдёт ошибка `Getting analyzing error. Detail message: Cannot find any backend with location: rack:xxx`.

При создании таблицы

Вы можете использовать свойство `"labels.location"` для распределения данных таблицы по rack 1 и rack 2 при создании таблицы:

```
CREATE TABLE example_table (
    order_id bigint NOT NULL,
    dt date NOT NULL,
    user_id INT NOT NULL,
    good_id INT NOT NULL,
    cnt int NOT NULL,
    revenue int NOT NULL
)
PROPERTIES
("labels.location" = "rack:rack1,rack:rack2");
```

Для вновь созданных таблиц значение по умолчанию свойства таблицы `labels.location` равно `*`, что означает, что replica равномерно распределяются по всем меткам. Если распределение данных вновь созданной таблицы не должно учитывать географическое расположение серверов в cluster, вы можете вручную установить свойство таблицы `"labels.location" = ""`.

После создания таблицы

Если вам нужно изменить место распределения данных таблицы после её создания, например, изменить место на rack 1, rack 2 и rack 3, вы можете выполнить следующий оператор:

```
ALTER TABLE example_table
    SET ("labels.location" = "rack:rack1,rack:rack2,rack:rack3");
```

> **ПРИМЕЧАНИЕ**
>
> Если вы обновили Selena до версии 3.2.8 или более поздней, для исторических таблиц, созданных до обновления, данные по умолчанию не распределяются на основе меток. Если вам нужно распределить данные исторической таблицы на основе меток, вы можете выполнить следующий оператор, чтобы добавить метки к исторической таблице:
>
> ```
> ALTER TABLE example_table1
>  SET ("labels.location" = "rack:rack1,rack:rack2");
> ```

Использование меток для указания распределения данных материализованного представления на узлах BE

Если вам нужно указать места, куда будут распределены данные асинхронного материализованного представления, например, распределение данных по двум стойкам, rack1 и rack2, вы можете добавить метки к материализованному представлению.

После добавления меток все replica одного tablet в материализованном представлении распределяются по меткам по принципу Round-Robin. Более того, если несколько replica одного tablet существуют в рамках одной метки, эти replica будут распределены как можно более равномерно между разными BE в этой метке.

> **ПРИМЕЧАНИЕ**
>
> - Если общее количество узлов BE, связанных с метками, меньше количества replica, система в первую очередь обеспечит достаточное количество replica. В этом случае replica могут быть распределены не так, как указано в метке.
> - Метки, которые нужно связать с материализованным представлением, должны уже существовать. В противном случае произойдёт ошибка `Getting analyzing error. Detail message: Cannot find any backend with location: rack:xxx`.

При создании материализованного представления

Если вы хотите распределить данные материализованного представления по rack 1 и rack 2 при его создании, вы можете выполнить следующий оператор:

```
CREATE MATERIALIZED VIEW mv_example_mv
DISTRIBUTED BY RANDOM
PROPERTIES (
"labels.location" = "rack:rack1,rack:rack2")
as
select order_id, dt from example_table;
```

Для вновь созданных материализованных представлений значение по умолчанию свойства `labels.location` равно `*`, что означает, что replica равномерно распределяются по всем меткам. Если распределение данных вновь созданного материализованного представления не должно учитывать географическое расположение серверов в cluster, вы можете вручную установить свойство `"labels.location" = ""`.

После создания материализованного представления

Если вам нужно изменить место распределения данных материализованного представления после его создания, например, изменить место на rack 1, rack 2 и rack 3, вы можете выполнить следующий оператор:

```
ALTER MATERIALIZED VIEW mv_example_mv
    SET ("labels.location" = "rack:rack1,rack:rack2,rack:rack3");
```

ПРИМЕЧАНИЕ

Если вы обновили Selena до версии 3.2.8 или более поздней, для существующих материализованных представлений, созданных до обновления, данные по умолчанию не распределяются на основе меток. Если вам нужно распределить данные существующего материализованного представления на основе меток, вы можете выполнить следующий оператор, чтобы добавить метки к материализованному представлению:

```
ALTER TABLE example_mv1
    SET ("labels.location" = "rack:rack1,rack:rack2");
```

Файловый менеджер

С помощью файлового менеджера вы можете создавать, просматривать и удалять файлы, такие как файлы, используемые для доступа к внешним источникам данных: файлы открытых ключей, файлы закрытых ключей и файлы сертификатов. Вы можете ссылаться на созданные файлы или получать к ним доступ с помощью команд.

Основные понятия

Файл: относится к файлу, который создаётся и сохраняется в Selena. После создания и сохранения файла в Selena система присваивает файлу уникальный идентификатор. Вы можете найти файл по имени базы данных, каталога и имени файла. В базе данных только пользователь с правами администратора может создавать и удалять файлы, а все пользователи, имеющие разрешения на доступ к базе данных, могут использовать файлы, принадлежащие этой базе данных.

Прежде чем начать

- Настройте следующие параметры для каждого FE.
  - `small_file_dir`: путь, по которому хранятся загруженные файлы. Путь по умолчанию - `small_files/`, который находится в рабочей директории FE. Вам нужно указать этот параметр в файле **fe.conf** и затем перезапустить FE, чтобы изменения вступили в силу.
  - `max_small_file_size_bytes`: максимальный размер одного файла. Значение по умолчанию этого параметра - 1 МБ. Если размер файла превышает значение этого параметра, файл не может быть создан. Вы можете указать этот параметр с помощью оператора ADMIN SET CONFIG.
  - `max_small_file_number`: максимальное количество файлов, которые могут быть созданы в cluster. Значение по умолчанию этого параметра - 100. Если количество созданных файлов достигает значения этого параметра, вы не можете продолжать создавать файлы. Вы можете указать этот параметр с помощью оператора ADMIN SET CONFIG.

> Примечание: Увеличение значений этих двух параметров приводит к увеличению использования памяти FE. Поэтому мы рекомендуем не увеличивать значения этих двух параметров без необходимости.

- Настройте следующие параметры для каждого BE.

`small_file_dir`: путь, по которому хранятся загруженные файлы. Путь по умолчанию - `lib/small_files/`, который находится в рабочей директории BE. Вы можете указать этот параметр в файле **be.conf**.

Создание файла

Вы можете выполнить оператор CREATE FILE для создания файла. Для получения дополнительной информации см. CREATE FILE. После создания файла он загружается и сохраняется в Selena.

Просмотр файла

Вы можете выполнить оператор SHOW FILE для просмотра информации о файле, хранящемся в базе данных. Для получения дополнительной информации см. SHOW FILE.

Удаление файла

Вы можете выполнить оператор DROP FILE для удаления файла. Для получения дополнительной информации см. DROP FILE.

Как FE и BE используют файл

- **FE**: Класс SmallFileMgr хранит данные, связанные с файлом, в указанной директории FE. Затем класс SmallFileMgr возвращает локальный путь к файлу, чтобы FE мог использовать файл.
- **BE**: BE вызывает API **/api/get_small_file** (HTTP) для загрузки файла в указанную директорию и записывает информацию о файле. Когда BE запрашивает использование файла, BE проверяет, был ли файл загружен, а затем верифицирует файл. Если файл проходит верификацию, возвращается путь к файлу. Если файл не проходит верификацию, файл удаляется и повторно загружается с FE. При перезапуске BE он предварительно загружает загруженный файл в память.

2.3 Восстановление данных

Selena поддерживает восстановление данных для ошибочно удалённых баз данных/таблиц/partition. После выполнения `drop table` или `drop database` Selena не удаляет данные физически немедленно, а сохраняет их в корзине на определённый период времени (по умолчанию 1 день). Администраторы могут восстановить ошибочно удалённые данные с помощью команды `RECOVER`.

2.3.1 Связанные команды

Синтаксис:

```
-- 1) Восстановить базу данных
RECOVER DATABASE db_name;
-- 2) Восстановить таблицу
RECOVER TABLE [db_name.]table_name;
-- 3) Восстановить partition
RECOVER PARTITION partition_name FROM [db_name.]table_name;
```

2.3.2 Примечания

1. Эта операция может восстановить только удалённую метаинформацию. Время хранения по умолчанию составляет 1 день и может быть настроено параметром `catalog_trash_expire_second` в `fe.conf`.
2. Если после удаления метаинформации была создана новая метаинформация с тем же именем и типом, ранее удалённая метаинформация не может быть восстановлена.

2.3.3 Примеры

1. Восстановить базу данных с именем `example_db`

```
RECOVER DATABASE example_db;
```

2. Восстановить таблицу с именем `example_tbl`

```
RECOVER TABLE example_db.example_tbl;
```

3. Восстановить partition с именем `p1` в таблице `example_tbl`

```
RECOVER PARTITION p1 FROM example_tbl;
```

2.4 Восстановление метаданных

В этом разделе описывается, как восстановить метаданные в ваших cluster Selena, когда узлы FE сталкиваются с различными исключениями.

Как правило, прибегать к восстановлению метаданных необходимо только при возникновении одной из следующих проблем:

- FE не удаётся перезапустить
- FE не может предоставлять сервисы
- Восстановление метаданных на новом узле FE из резервной копии

Проверьте возникшую проблему, следуйте решению, предоставленному в соответствующем разделе, и выполните рекомендуемые действия.

2.4.1 FE не удаётся перезапустить

Узлы FE могут не перезапускаться, если метаданные повреждены или несовместимы с cluster после отката версии.

Несовместимость метаданных после отката

При понижении версии cluster Selena узлы FE могут не перезапускаться, если метаданные несовместимы с теми, что были до понижения версии.

Вы можете определить эту проблему, если столкнулись со следующим исключением при понижении версии cluster:

```
UNKNOWN Operation Type xxx
```

Выполните следующие шаги для восстановления метаданных и запуска FE:

1. Остановите все узлы FE.

2. Сделайте резервную копию директорий метаданных `meta_dir` всех узлов FE.

3. Добавьте конфигурацию `metadata_ignore_unknown_operation_type = true` в конфигурационные файлы **fe.conf** всех узлов FE.

4. Запустите все узлы FE и проверьте целостность ваших данных и метаданных.

5. Если данные и метаданные целы, выполните следующую команду для создания файла образа метаданных:

```
ALTER SYSTEM CREATE IMAGE;
```

6. После того как новый файл образа будет передан в директорию **meta/image** всех узлов FE, вы можете удалить конфигурацию `metadata_ignore_unknown_operation_type= true` из всех конфигурационных файлов FE и перезапустить узлы FE.

Повреждение метаданных

Повреждение метаданных BDBJE и метаданных Selena приводит к сбоям при перезапуске.

Повреждение метаданных BDBJE

Ошибка VLSN

Вы можете определить ошибку VLSN по следующему сообщению:

```
recoveryTracker should overlap or follow on disk last VLSN of 6,684,650
recoveryFirst= 6,684,652 UNEXPECTED_STATE_FATAL: Unexpected internal state,
unable to continue. Environment is invalid and must be closed.
```

Выполните следующие шаги для исправления этой проблемы:

1. Очистите директорию метаданных `meta_dir` узла FE, который выдаёт это исключение.

2. Перезапустите узел FE, используя Leader FE в качестве helper.

```
# Замените <leader_ip> на IP-адрес (priority_networks)
# узла Leader FE, а <leader_edit_log_port> (по умолчанию: 9010) на
# edit_log_port узла Leader FE.
./fe/bin/start_fe.sh --helper <leader_ip>:<leader_edit_log_port> --
daemon
```

ПОДСКАЗКА

- Эта ошибка исправлена в Selena v1.5.2. Вы можете избежать этой проблемы, обновив cluster до версии v1.5.2 или более поздней.
- Это решение не применимо, если более половины узлов FE столкнулись с этой проблемой. Вы должны следовать инструкциям, предоставленным в разделе Крайние меры, чтобы исправить проблему.

RollbackException

Вы можете определить эту проблему по следующему сообщению об ошибке:

```
must rollback 1 total commits(1 of which were durable) to the earliest
point indicated by transaction id=-14752149 time=2022-01-12 14:36:28.21
vlsn=28,069,415 lsn=0x1174/0x16e durable=false in order to rejoin the
replication group. All existing ReplicatedEnvironment handles must be
closed and reinstantiated. Log files were truncated to file 0x4467, offset
0x269, vlsn 28,069,413 HARD_RECOVERY: Rolled back past transaction commit
or abort. Must run recovery by re-opening Environment handles Environment
is invalid and must be closed.
```

Эта проблема возникает, когда Leader FE записывает метаданные BDBJE, но не успевает синхронизировать их с Follower FE до зависания. После перезапуска бывший Leader становится Follower, что приводит к повреждению метаданных.

Для решения этой проблемы достаточно перезапустить этот узел ещё раз, чтобы удалить повреждённые метаданные.

ReplicaWriteException

Вы можете определить эту проблему по ключевому слову `removeReplicaDb` в журнале FE **fe.log**.

```
Caused by: com.sleepycat.je.rep.ReplicaWriteException: (JE 18.3.16) Problem c
transaction 25000090. The current state is:REPLICA. The node transitioned to
Feb 23 01:31:00 UTC 2024 Problem seen replaying entry NameLN_TX/14 vlsn=1,902
isReplicated="1"  txn=-953505106 dbop=REMOVE Originally thrown by HA thread:
10.233.132.23_9010_1684154162022(6)
        at com.sleepycat.je.rep.txn.ReadonlyTxn.disallowReplicaWrite(Readonly
[selena-bdb-je-18.3.16.jar:?]
        at com.sleepycat.je.dbi.DbTree.checkReplicaWrite(DbTree.java:880) ~[s
18.3.16.jar:?]
        at com.sleepycat.je.dbi.DbTree.doCreateDb(DbTree.java:579) ~[selena-b
18.3.16.jar:?]
        at com.sleepycat.je.dbi.DbTree.createInternalDb(DbTree.java:507) ~[se
18.3.16.jar:?]
        at com.sleepycat.je.cleaner.ExtinctionScanner.openDb(ExtinctionScanne
[selena-bdb-je-18.3.16.jar:?]
        at
com.sleepycat.je.cleaner.ExtinctionScanner.prepareForDbExtinction(ExtinctionS
~[selena-bdb-je-18.3.16.jar:?]
        at com.sleepycat.je.dbi.DbTree.doRemoveDb(DbTree.java:1208) ~[selena-
18.3.16.jar:?]
        at com.sleepycat.je.dbi.DbTree.removeReplicaDb(DbTree.java:1261) ~[se
18.3.16.jar:?]
        at com.sleepycat.je.rep.impl.node.Replay.applyNameLN(Replay.java:996)
je-18.3.16.jar:?]
        at com.sleepycat.je.rep.impl.node.Replay.replayEntry(Replay.java:722)
je-18.3.16.jar:?]
        at com.sleepycat.je.rep.impl.node.Replica$ReplayThread.run(Replica.ja
[selena-bdb-je-18.3.16.jar:?]
```

Эта проблема возникает, когда версия BDBJE неисправного узла FE (v18.3.*) не совпадает с версией Leader FE (v7.3.7).

Выполните следующие шаги для исправления этой проблемы:

1. Удалите неисправный узел Follower или Observer.

```
-- Для удаления узла Follower замените <follower_host> на IP-адрес
(priority_networks)
-- узла Follower, а <follower_edit_log_port> (по умолчанию: 9010) на
-- edit_log_port узла Follower.
ALTER SYSTEM DROP FOLLOWER "<follower_host>:<follower_edit_log_port>";

-- Для удаления узла Observer замените <observer_host> на IP-адрес
(priority_networks)
-- узла Observer, а <observer_edit_log_port> (по умолчанию: 9010) на
-- edit_log_port узла Observer.
ALTER SYSTEM DROP OBSERVER "<observer_host>:<observer_edit_log_port>";
```

2. Добавьте неисправный узел обратно в cluster.

```
-- Добавить узел Follower:
ALTER SYSTEM ADD FOLLOWER "<follower_host>:<follower_edit_log_port>";

-- Добавить узел Observer:
ALTER SYSTEM ADD OBSERVER "<observer_host>:<observer_edit_log_port>";
```

3. Очистите директорию метаданных `meta_dir` неисправного узла.

4. Перезапустите неисправный узел, используя Leader FE в качестве helper.

```
# Замените <leader_ip> на IP-адрес (priority_networks)
# узла Leader FE, а <leader_edit_log_port> (по умолчанию: 9010) на
# edit_log_port узла Leader FE.
./fe/bin/start_fe.sh --helper <leader_ip>:<leader_edit_log_port> --
daemon
```

5. После восстановления неисправного узла до здорового состояния вам необходимо обновить пакеты BDBJE в вашем cluster до **selena-bdb-je-18.3.16.jar** (или обновить cluster Selena до версии v1.5.2 или более поздней), следуя порядку: сначала Follower, затем Leader.

InsufficientLogException

Вы можете определить эту проблему по следующему сообщению об ошибке:

```
xxx INSUFFICIENT_LOG: Log files at this node are obsolete. Environment is
invalid and must be closed.
```

Эта проблема возникает, когда узлу Follower требуется полная синхронизация метаданных. Это может произойти в одной из следующих ситуаций:

- Метаданные на узле Follower отстают от метаданных узла Leader, который уже выполнил CheckPoint метаданных. Узел Follower не может выполнить инкрементное обновление метаданных, поэтому требуется полная синхронизация метаданных.
- Бывший узел Leader записывает и выполняет checkpoint своих метаданных, но не успевает синхронизировать их с узлами Follower FE до зависания. После перезапуска он становится узлом Follower. С повреждёнными метаданными, для которых выполнен checkpoint, узел Follower не может выполнить инкрементное удаление своих метаданных, поэтому требуется полная синхронизация метаданных.

Обратите внимание, что это исключение будет выброшено при добавлении нового узла Follower в cluster. В этом случае никаких действий предпринимать не нужно. Если это исключение выбрасывается для существующего узла Follower или узла Leader, достаточно перезапустить узел.

HANDSHAKE_ERROR: Error during the handshake between two nodes

Вы можете определить эту проблему по следующему сообщению об ошибке:

```
2023-11-13 21:51:55,271 WARN (replayer|82) [BDBJournalCursor.wrapDatabaseExce
get DB names for 1 times!Got EnvironmentFailureExce
com.sleepycat.je.EnvironmentFailureException: (JE 18.3.16) Environment must b
com.sleepycat.je.EnvironmentFailureException: Environment invalid because of
18.3.16) 10.26.5.115_9010_1697071897979(1):/data1/meta/bdb A replica with the
10.26.5.115_9010_1697071897979(1) is already active with the Feeder:null HAND
during the handshake between two nodes. Some validity or compatibility check
further communication between the nodes. Environment is invalid and must be c
        at
com.sleepycat.je.EnvironmentFailureException.wrapSelf(EnvironmentFailureExcep
bdb-je-18.3.16.jar:?]
        at com.sleepycat.je.dbi.EnvironmentImpl.checkIfInvalid(EnvironmentImp
bdb-je-18.3.16.jar:?]
        at com.sleepycat.je.dbi.EnvironmentImpl.checkOpen(EnvironmentImpl.jav
je-18.3.16.jar:?]
        at com.sleepycat.je.Environment.checkOpen(Environment.java:2697) ~[se
18.3.16.jar:?]
        at com.sleepycat.je.Environment.getDatabaseNames(Environment.java:245
18.3.16.jar:?]
        at
com.selena.journal.bdbje.BDBEnvironment.getDatabaseNamesWithPrefix(BDBEnviron
fe.jar:?]
        at com.selena.journal.bdbje.BDBJournalCursor.refresh(BDBJournalCursor
fe.jar:?]
        at com.selena.server.GlobalStateMgr$5.runOneCycle(GlobalStateMgr.java
fe.jar:?]
        at com.selena.common.util.Daemon.run(Daemon.java:115) ~[selena-fe.jar
        at com.selena.server.GlobalStateMgr$5.run(GlobalStateMgr.java:2216) ~
Caused by: com.sleepycat.je.EnvironmentFailureException: Environment invalid
exception: (JE 18.3.16) 10.26.5.115_9010_1697071897979(1):/data1/meta/bdb A r
10.26.5.115_9010_1697071897979(1) is already active with the Feeder:null HAND
during the handshake between two nodes. Some validity or compatibility check
further communication between the nodes. Environment is invalid and must be c
by HA thread: UNKNOWN 10.26.5.115_9010_1697071897979(1) Originally thrown by
10.26.5.115_9010_1697071897979(1)
        at
com.sleepycat.je.rep.stream.ReplicaFeederHandshake.negotiateProtocol(ReplicaF
~[selena-bdb-je-18.3.16.jar:?]
        at
com.sleepycat.je.rep.stream.ReplicaFeederHandshake.execute(ReplicaFeederHands
bdb-je-18.3.16.jar:?]
        at com.sleepycat.je.rep.impl.node.Replica.initReplicaLoop(Replica.jav
18.3.16.jar:?]
        at com.sleepycat.je.rep.impl.node.Replica.runReplicaLoopInternal(Repl
bdb-je-18.3.16.jar:?]
```

```
        at com.sleepycat.je.rep.impl.node.Replica.runReplicaLoop(Replica.java
18.3.16.jar:?]
        at com.sleepycat.je.rep.impl.node.RepNode.run(RepNode.java:1869) ~[se
18.3.16.jar:?]
```

Эта проблема возникает, когда бывший узел Leader зависает и снова становится активным, в то время как выжившие узлы Follower пытаются выбрать нового Leader. Узлы Follower попытаются установить новое соединение с бывшим узлом Leader. Однако узел Leader отклонит запрос на соединение, потому что старое соединение всё ещё существует. После отклонения запроса узел Follower установит окружение как недействительное и выбросит это исключение.

Для решения этой проблемы вы можете либо увеличить размер JVM heap, либо использовать алгоритм G1 GC.

DatabaseNotFoundException

Вы можете определить эту проблему по следующему сообщению об ошибке:

```
2024-01-05 12:47:21,087 INFO (main|1) [BDBEnvironment.ensureHelperInLocal():3
skip check local environment because helper node and local node are identical
2024-01-05 12:47:21,339 ERROR (MASTER 172.17.0.1_9112_1704430041062(-1)|1)
[SelenaFE.start():186] SelenaFE start failed
com.sleepycat.je.DatabaseNotFoundException: (JE 18.3.16) _jeRepGroupDB
        at com.sleepycat.je.rep.impl.RepImpl.openGroupDb(RepImpl.java:1974) ~
[selena-bdb-je-18.3.16.jar:?]
        at com.sleepycat.je.rep.impl.RepImpl.getGroupDb(RepImpl.java:1912) ~[
bdb-je-18.3.16.jar:?]
        at
com.sleepycat.je.rep.impl.RepGroupDB.reinitFirstNode(RepGroupDB.java:1439) ~[
bdb-je-18.3.16.jar:?]
        at com.sleepycat.je.rep.impl.node.RepNode.reinitSelfElect(RepNode.jav
~[selena-bdb-je-18.3.16.jar:?]
        at com.sleepycat.je.rep.impl.node.RepNode.startup(RepNode.java:874) ~
[selena-bdb-je-18.3.16.jar:?]
        at com.sleepycat.je.rep.impl.node.RepNode.joinGroup(RepNode.java:2153
[selena-bdb-je-18.3.16.jar:?]
        at com.sleepycat.je.rep.impl.RepImpl.joinGroup(RepImpl.java:618) ~[se
bdb-je-18.3.16.jar:?]
        at
com.sleepycat.je.rep.ReplicatedEnvironment.joinGroup(ReplicatedEnvironment.ja
~[selena-bdb-je-18.3.16.jar:?]
        at com.sleepycat.je.rep.ReplicatedEnvironment.<init>
(ReplicatedEnvironment.java:619) ~[selena-bdb-je-18.3.16.jar:?]
        at com.sleepycat.je.rep.ReplicatedEnvironment.<init>
(ReplicatedEnvironment.java:464) ~[selena-bdb-je-18.3.16.jar:?]
        at com.sleepycat.je.rep.ReplicatedEnvironment.<init>
(ReplicatedEnvironment.java:538) ~[selena-bdb-je-18.3.16.jar:?]
        at com.sleepycat.je.rep.util.DbResetRepGroup.reset(DbResetRepGroup.ja
~[selena-bdb-je-18.3.16.jar:?]
        at
com.selena.journal.bdbje.BDBEnvironment.initConfigs(BDBEnvironment.java:188)
[selena-fe.jar:?]
        at com.selena.journal.bdbje.BDBEnvironment.setup(BDBEnvironment.java:
[selena-fe.jar:?]
        at
com.selena.journal.bdbje.BDBEnvironment.initBDBEnvironment(BDBEnvironment.jav
~[selena-fe.jar:?]
        at com.selena.journal.JournalFactory.create(JournalFactory.java:31) ~
[selena-fe.jar:?]
        at com.selena.server.GlobalStateMgr.initJournal(GlobalStateMgr.java:1
[selena-fe.jar:?]
        at com.selena.server.GlobalStateMgr.initialize(GlobalStateMgr.java:11
[selena-fe.jar:?]
```

```
        at com.selena.SelenaFE.start(SelenaFE.java:129) ~[selena-fe.jar:?]
        at com.selena.SelenaFE.main(SelenaFE.java:83) ~[selena-fe.jar:?]
```

Эта проблема возникает, когда вы добавляете конфигурацию `metadata_failure_recovery = true` в конфигурационный файл FE **fe.conf**.

Для решения этой проблемы необходимо удалить конфигурацию и перезапустить узел.

Повреждение метаданных Selena

Вы можете определить проблему повреждения метаданных Selena по одному из следующих сообщений об ошибке:

```
failed to load journal type xxx
```

Или

```
catch exception when replaying
```

> **WARNING**
>
> Перед тем как приступить к восстановлению метаданных, следуя приведённому ниже решению, настоятельно рекомендуется обратиться за помощью к техническим экспертам сообщества Selena, поскольку это решение может привести к **потере данных**.

Выполните следующие шаги для исправления этой проблемы:

Игнорирование ошибочного Journal ID (предпочтительно)

1. Остановите все узлы FE.
2. Сделайте резервную копию директорий метаданных всех узлов FE.
3. Найдите ошибочный journal ID в журналах. `xxx` в следующем журнале представляет ошибочный journal ID.

```
got interrupt exception or inconsistent exception when replay journal
xxx, will exit
```

4. Добавьте следующую конфигурацию во все файлы **fe.conf** и запустите узлы FE.

```
metadata_journal_skip_bad_journal_ids=xxx
```

5. Если запуск снова завершится неудачей, определите новый неудачный journal ID через шаг 3, добавьте его в **fe.conf** и перезапустите узлы с предыдущими конфигурациями без изменений.

```
metadata_journal_skip_bad_journal_ids=xxx,yyy
```

6. Если система всё ещё не запускается после вышеуказанных шагов, или если слишком много неудачных journal ID, перейдите к режиму восстановления.

Режим восстановления

1. Остановите все узлы FE.

2. Сделайте резервную копию директорий метаданных `meta_dir` всех узлов FE.

3. Добавьте конфигурацию `metadata_enable_recovery_mode = true` в конфигурационные файлы **fe.conf** всех узлов FE. Обратите внимание, что загрузка данных запрещена в этом режиме.

4. Запустите все узлы FE и выполните запросы к таблицам в cluster, чтобы проверить целостность ваших данных.

   Необходимо дождаться завершения восстановления метаданных, если при запросе этих таблиц возвращается следующая ошибка:

```
ERROR 1064 (HY000): capture_consistent_versions error: version already
been compacted.
```

Вы можете выполнить следующую команду с узла Leader FE для просмотра прогресса восстановления метаданных:

```
SHOW PROC '/meta_recovery';
```

Эта команда покажет partition, которые не удалось восстановить. Вы можете следовать возвращённым рекомендациям для восстановления partition. Если ничего не возвращается, это означает успешное восстановление.

5. Если данные и метаданные целы, выполните следующую команду для создания файла образа метаданных:

```
ALTER SYSTEM CREATE IMAGE;
```

6. После того как новый файл образа будет передан в директорию **meta/image** всех узлов FE, вы можете удалить конфигурацию `metadata_enable_recovery_mode = true` из всех конфигурационных файлов FE и перезапустить узлы FE.

2.4.2 FE не может предоставлять сервисы

FE не будет предоставлять сервисы, когда узлы Follower FE не могут выполнить выборы Leader. При возникновении этой проблемы вы можете обнаружить повторяющуюся запись в журнале:

```
wait globalStateMgr to be ready. FE type: INIT. is ready: false
```

Различные исключения могут вызвать эту проблему. Настоятельно рекомендуется устранять проблему поэтапно, следуя разделам ниже. Применение неподходящих решений усугубит проблему и, вероятно, приведёт к потере данных.

1. Большинство узлов Follower не работают

Если большинство узлов Follower не работают, группа FE не будет предоставлять сервисы. Здесь «большинство» означает `1 + (количество узлов Follower/2)`. Обратите внимание, что узел Leader FE сам является Follower, но узлы Observer не являются Follower.

- Вы можете определить роль каждого узла FE из файла **fe/meta/image/ROLE**:

```
cat fe/meta/image/ROLE

#Fri Jan 19 20:03:14 CST 2024
role=FOLLOWER
hostType=IP
name=172.26.92.154_9312_1705568349984
```

- Вы можете просмотреть общее количество узлов Follower из журнала BDBJE:

```
grep "Current group size" fe/meta/bdb/je.info.0

# Пример вывода показывает, что в cluster три узла Follower.
2024-01-24 08:21:44.754 UTC INFO [172.26.92.139_29917_1698226672727]
Current group size: 3
```

Для решения этой проблемы необходимо запустить все узлы Follower в cluster. Если их невозможно перезапустить, обратитесь к разделу Крайние меры.

2. IP-адрес узла изменился

Если `priority_networks` узла не настроен, узел FE случайным образом выберет доступный IP-адрес при перезапуске. Если IP-адрес, записанный в метаданных BDBJE, отличается от того, который используется для запуска узла, FE не будет предоставлять сервисы.

- Вы можете просмотреть IP-адрес, записанный в метаданных BDBJE, из файла **fe/meta/image/ROLE**:

```
cat fe/meta/image/ROLE

#Fri Jan 19 20:03:14 CST 2024
role=FOLLOWER
hostType=IP
name=172.26.92.154_9312_1705568349984
```

  Значение `172.26.92.154` перед первым подчёркиванием — это IP-адрес, записанный в метаданных BDBJE.

- Вы можете просмотреть IP-адрес, используемый для запуска узла, из журнала FE:

```
grep "IP:" fe/log/fe.log

2024-02-06 14:33:58,211 INFO (main|1)
[FrontendOptions.initAddrUseIp():249] Use IP init local addr, IP:
/172.17.0.1
2024-02-06 14:34:27,689 INFO (main|1)
[FrontendOptions.initAddrUseIp():249] Use IP init local addr, IP:
/172.17.0.1
```

Для решения этой проблемы необходимо установить `priority_networks` узла в конфигурационном файле FE **fe.conf** на IP-адрес, записанный в **fe/meta/image/ROLE**, и перезапустить узел.

3. Системные часы между узлами не синхронизированы

Вы можете определить эту проблему по следующему сообщению об ошибке из **fe.out**, **fe.log** или **fe/meta//bdb/je.info.0**:

```
com.sleepycat.je.EnvironmentFailureException: (JE 7.3.7) Environment must
be closed, caused by: com.sleepycat.je.EnvironmentFailureException:
Environment invalid because of previous exception: (JE 7.3.7)
172.26.92.139_29917_1631006307557(2180):xxx Clock delta: 11020 ms. between
Feeder: 172.26.92.154_29917_1641969377236 and this Replica exceeds max
permissible delta: 5000 ms. HANDSHAKE_ERROR: Error during the handshake
between two nodes. Some validity or compatibility check failed, preventing
further communication between the nodes. Environment is invalid and must be
closed. fetchRoot of 0x1278/0x1fcbb8 state=0 expires=never
```

Необходимо синхронизировать системные часы на всех узлах.

4. Недостаточно свободного места на диске

После обновления Selena до версии v1.5.2 или более поздней, или обновления BDBJE до версии v18 или более поздней, узел может не перезапуститься, если свободное пространство на диске, где хранится `meta_dir`, составляет менее 5 ГБ.

Вы можете просмотреть версию BDBJE из пакета **.jar** в директории **fe/lib**.

Для решения этой проблемы вы можете расширить диск или выделить отдельный диск большей ёмкости для метаданных FE.

5. `edit_log_port` изменён

Если `edit_log_port`, записанный в метаданных BDBJE, отличается от того, который настроен в **fe.conf**, FE не будет предоставлять сервисы.

Вы можете просмотреть `edit_log_port`, записанный в метаданных BDBJE, из файла **fe/meta/image/ROLE**:

```
cat fe/meta/image/ROLE

#Fri Jan 19 20:03:14 CST 2024
role=FOLLOWER
hostType=IP
name=172.26.92.154_9312_1705568349984
```

Значение `9312` перед вторым подчёркиванием — это `edit_log_port`, записанный в метаданных BDBJE.

Для решения этой проблемы необходимо установить `edit_log_port` узла в конфигурационном файле FE **fe.conf** на `edit_log_port`, записанный в **fe/meta/image/ROLE**, и перезапустить узел.

6. Недостаточный размер JVM heap

Вы можете просмотреть использование памяти JVM с помощью команды `jstat`:

```
jstat -gcutil pid 1000 1000

  S0     S1     E      O      M     CCS    YGC     YGCT    FGC    FGCT
GCT
  0.00 100.00  27.78  95.45  97.77  94.45     24    0.226     1    0.065
0.291
  0.00 100.00  44.44  95.45  97.77  94.45     24    0.226     1    0.065
0.291
  0.00 100.00  55.56  95.45  97.77  94.45     24    0.226     1    0.065
0.291
  0.00 100.00  72.22  95.45  97.77  94.45     24    0.226     1    0.065
0.291
  0.00 100.00  88.89  95.45  97.77  94.45     24    0.226     1    0.065
0.291
  0.00 100.00   5.26  98.88  97.80  94.45     25    0.231     1    0.065
0.297
  0.00 100.00  21.05  98.88  97.80  94.45     25    0.231     1    0.065
0.297
  0.00 100.00  31.58  98.88  97.80  94.45     25    0.231     1    0.065
0.297
  0.00 100.00  47.37  98.88  97.80  94.45     25    0.231     1    0.065
0.297
  0.00 100.00  63.16  98.88  97.80  94.45     25    0.231     1    0.065
0.297
  0.00 100.00  73.68  98.88  97.80  94.45     25    0.231     1    0.065
0.297
```

Если проценты в поле `O` остаются высокими, это указывает на недостаточный размер JVM heap.

Для решения этой проблемы необходимо увеличить размер JVM heap.

7. Latch timeout. com.sleepycat.je.log.LogbufferPool_FullLatch

Вы можете определить эту проблему по следующему сообщению об ошибке:

```
Environment invalid because of previous exception: xxx Latch timeout.
com.sleepycat.je.log.LogbufferPool_FullLatch xxx'
        at
com.sleepycat.je.EnvironmentFailureException.unexpectedState(EnvironmentFailu
        at com.sleepycat.je.latch.LatchSupport.handleTimeout(LatchSupport.jav
        at com.sleepycat.je.latch.LatchWithStatsImpl.acquireExclusive(LatchWi
        at com.sleepycat.je.log.LogBufferPool.bumpCurrent(LogBufferPool.java:
        at com.sleepycat.je.log.LogManager.flushInternal(LogManager.java:1373
        at com.sleepycat.je.log.LogManager.flushNoSync(LogManager.java:1337)
        at com.sleepycat.je.log.LogFlusher$FlushTask.run(LogFlusher.java:232)
        at java.util.TimerThread.mainLoop(Timer.java:555)
        at java.util.TimerThread.run(Timer.java:505)
```

Эта проблема возникает при чрезмерной нагрузке на локальный диск узла FE.

Для решения этой проблемы вы можете выделить отдельный диск для метаданных FE или заменить диск на более производительный.

8. InsufficientReplicasException

Вы можете определить эту проблему по следующему сообщению об ошибке:

```
com.sleepycat.je.rep.InsufficientReplicasException: (JE 7.3.7) Commit
policy: SIMPLE_MAJORITY required 1 replica. But none were active with this
master.
```

Эта проблема возникает, когда узел Leader FE или узлы Follower FE используют чрезмерные ресурсы памяти, что приводит к Full GC.

Для решения этой проблемы вы можете либо увеличить размер JVM heap, либо использовать алгоритм G1 GC.

9. UnknownMasterException

Вы можете определить эту проблему по следующему сообщению об ошибке:

```
com.sleepycat.je.rep.UnknownMasterException: (JE 18.3.16) Could not determine
at:[/xxx.xxx.xxx.xxx:9010, /xxx.xxx.xxx.xxx:9010]
        at com.sleepycat.je.rep.elections.Learner.findMaster(Learner.java:443
18.3.16.jar:?]
        at
com.sleepycat.je.rep.util.ReplicationGroupAdmin.getMasterSocket(ReplicationGr
[selena-bdb-je-18.3.16.jar:?]
        at
com.sleepycat.je.rep.util.ReplicationGroupAdmin.doMessageExchange(Replication
~[selena-bdb-je-18.3.16.jar:?]
        at
com.sleepycat.je.rep.util.ReplicationGroupAdmin.getGroup(ReplicationGroupAdmi
[selena-bdb-je-18.3.16.jar:?]
        at com.selena.ha.BDBHA.getElectableNodes(BDBHA.java:178) ~[selena-fe.
        at com.selena.common.proc.FrontendsProcNode.getFrontendsInfo(Frontend
[selena-fe.jar:?]
        at com.selena.common.proc.FrontendsProcNode.fetchResult(FrontendsProc
[selena-fe.jar:?]
        at com.selena.sql.ast.ShowProcStmt.getMetaData(ShowProcStmt.java:74)
        at com.selena.qe.ShowExecutor.handleShowProc(ShowExecutor.java:872) ~
        at com.selena.qe.ShowExecutor.execute(ShowExecutor.java:286) ~[selena
        at com.selena.qe.StmtExecutor.handleShow(StmtExecutor.java:1574) ~[se
        at com.selena.qe.StmtExecutor.execute(StmtExecutor.java:688) ~[selena
        at com.selena.qe.ConnectProcessor.handleQuery(ConnectProcessor.java:3
fe.jar:?]
        at com.selena.qe.ConnectProcessor.dispatch(ConnectProcessor.java:530)
        at com.selena.qe.ConnectProcessor.processOnce(ConnectProcessor.java:8
fe.jar:?]
        at com.selena.mysql.nio.ReadListener.lambda$handleEvent$0(ReadListene
fe.jar:?]
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecut
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecu
        at java.lang.Thread.run(Thread.java:829) ~[?:?]
```

При выполнении `SHOW FRONTENDS`, если узел Leader FE не может быть найден, может быть несколько причин:

- Если наблюдается, что более половины узлов FE проходят Full GC, и продолжительность заметно велика.
- Или если журнал содержит ключевое слово `java.lang.OutOfMemoryError: Java heap space`.

Это связано с недостаточным объёмом памяти. Необходимо увеличить выделение памяти JVM.

10. Крайние меры

> **WARNING**
>
> Вы можете попробовать следующее решение как крайнюю меру только тогда, когда ни одно из предыдущих решений не работает.

Это решение предназначено только для экстремальных случаев, таких как:

- Большинство узлов Follower не могут быть перезапущены.
- Узлы Follower не могут выполнить выборы Leader из-за ошибок BDBJE.
- Исключения, отличные от упомянутых в предыдущих разделах.

Выполните следующие шаги для восстановления метаданных:

1. Остановите все узлы FE.

2. Сделайте резервную копию директорий метаданных `meta_dir` всех узлов FE.

3. Выполните следующие команды на **всех серверах, на которых размещены узлы FE**, чтобы определить узел с самыми последними метаданными.

   ```
   # Вам необходимо указать точный пакет .jar, который использует узел, в
   команде,
   # так как пакет варьируется в зависимости от версии Selena.
   java -jar fe/lib/selena-bdb-je-18.3.16.jar DbPrintLog -h meta/bdb/ -vd
   ```

   Пример вывода:

```
<DbPrintLog>
file 0x3b numRepRecords = 24479 firstVLSN = 1,434,126 lastVLSN =
1,458,604
file 0x3c numRepRecords = 22541 firstVLSN = 1,458,605 lastVLSN =
1,481,145
file 0x3d numRepRecords = 25176 firstVLSN = 1,481,146 lastVLSN =
1,506,321
......
file 0x74 numRepRecords = 26903 firstVLSN = 2,927,458 lastVLSN =
2,954,360
file 0x75 numRepRecords = 26496 firstVLSN = 2,954,361 lastVLSN =
2,980,856
file 0x76 numRepRecords = 18727 firstVLSN = 2,980,857 lastVLSN =
2,999,583
... 0 files at end
First file: 0x3b
Last file: 0x76
</DbPrintLog>
```

Узел с наибольшим значением `lastVLSN` имеет самые последние метаданные.

4. Найдите роль (Follower или Observer) узла FE с самыми последними метаданными из файла **fe/meta/image/ROLE**:

```
cat fe/meta/image/ROLE

#Fri Jan 19 20:03:14 CST 2024
role=FOLLOWER
hostType=IP
name=172.26.92.154_9312_1705568349984
```

Если несколько узлов имеют самые последние метаданные, рекомендуется продолжить с узлом Follower. Если несколько узлов Follower имеют самые последние метаданные, вы можете продолжить с любым из них.

5. Выполните соответствующие операции в зависимости от роли узла FE, который вы выбрали на предыдущем шаге.

Продолжить с узлом Follower | **Продолжить с узлом Observer**

Если узел Follower имеет самые последние метаданные, выполните следующие операции:

1. Добавьте следующие конфигурации в **fe.conf**:

```
bdbje_reset_election_group = true
```

2. Перезапустите узел и проверьте целостность ваших данных и метаданных.

3. Проверьте, является ли текущий узел FE узлом Leader FE.

```
SHOW FRONTENDS;
```

   - Если поле `Alive` имеет значение `true`, этот узел FE правильно запущен и добавлен в cluster.
   - Если поле `Role` имеет значение `LEADER`, этот узел FE является узлом Leader FE.

4. Если данные и метаданные целы, и роль узла — Leader, вы можете удалить добавленную ранее конфигурацию и перезапустить узел.

5. Очистите директории метаданных `meta_dir` узлов FE, которые вы хотите добавить обратно в cluster.

6. Запустите новые узлы Follower, используя новый Leader FE в качестве helper.

```
# Замените <leader_ip> на IP-адрес (priority_networks)
# узла Leader FE, а <leader_edit_log_port> (по умолчанию: 9010) на
# edit_log_port узла Leader FE.
./fe/bin/start_fe.sh --helper <leader_ip>:<leader_edit_log_port> --
daemon
```

7. Добавьте узлы Follower обратно в cluster.

```
ALTER SYSTEM ADD FOLLOWER "<new_follower_host>:
<new_follower_edit_log_port>";
```

После того как все узлы добавлены обратно в cluster, метаданные успешно восстановлены.

2.4.3 Восстановление метаданных на новом узле FE из резервной копии

Выполните следующие шаги, если вы хотите запустить новый узел FE с резервной копией метаданных:

1. Скопируйте резервную копию директории метаданных `meta_dir` на новый узел FE.

2. В конфигурационном файле узла FE установите `bdbje_reset_election_group` в значение `true`.

   ```
   bdbje_reset_election_group = true
   ```

3. Запустите узел FE.

   ```
   ./fe/bin/start_fe.sh
   ```

4. Проверьте, является ли текущий узел FE узлом Leader FE.

   ```
   SHOW FRONTENDS;
   ```

   Если поле `Role` имеет значение `LEADER`, этот узел FE является узлом Leader FE. Убедитесь, что его IP-адрес соответствует адресу текущего узла FE.

5. Если данные и метаданные целы, и роль узла — Leader, вы должны удалить конфигурацию `bdbje_reset_election_group` и перезапустить узел.

6. Теперь вы успешно запустили новый узел Leader FE с резервной копией метаданных. Вы можете добавить новые узлы Follower, используя новый Leader FE в качестве helper.

   ```
   # Замените <leader_ip> на IP-адрес (priority_networks)
   # узла Leader FE, а <leader_edit_log_port> (по умолчанию: 9010) на
   # edit_log_port узла Leader FE.
   ./fe/bin/start_fe.sh --helper <leader_ip>:<leader_edit_log_port> --
   daemon
   ```

2.4.4 Конфигурации, связанные с восстановлением метаданных

> **ПОДСКАЗКА**
>
> Вы должны удалить следующие конфигурации после завершения восстановления метаданных.

- bdbje_reset_election_group
- metadata_enable_recovery_mode
- metadata_ignore_unknown_operation_type

2.5 Инструмент миграции данных между cluster

Инструмент миграции данных между cluster Selena предоставляется сообществом Selena. С помощью этого инструмента вы можете легко мигрировать данные из исходного cluster в целевой cluster.

> **ПРИМЕЧАНИЕ**
>
> - Инструмент миграции данных между cluster Selena поддерживает только миграцию данных из cluster shared-nothing в другой cluster shared-nothing или в cluster shared-data.
> - Версия Selena целевого cluster должна быть v1.5.2 или более поздней.

2.5.1 Подготовка

Следующие подготовительные действия должны быть выполнены на целевом cluster для миграции данных.

Открытие портов

Если вы включили брандмауэр, вам необходимо открыть следующие порты:

Компонент	Порт	По умолчанию
FE	query_port	9030
FE	http_port	8030
FE	rpc_port	9020
BE	be_http_port	8040
BE	be_port	9060

Включение режима совместимости для репликации

Selena может вести себя по-разному между старой и новой версиями, что может вызвать проблемы при миграции данных между cluster. Поэтому вы должны включить режим совместимости для целевого cluster перед миграцией данных и отключить его после завершения миграции данных.

1. Вы можете проверить, включён ли режим совместимости для репликации, с помощью следующей команды:

   ```
   ADMIN SHOW FRONTEND CONFIG LIKE
   'enable_legacy_compatibility_for_replication';
   ```

   Если возвращается `true`, это означает, что режим совместимости для репликации включён.

2. Динамическое включение режима совместимости для репликации:

   ```
   ADMIN SET FRONTEND
   CONFIG("enable_legacy_compatibility_for_replication"="true");
   ```

3. Чтобы предотвратить автоматическое отключение режима совместимости во время процесса миграции данных в случае перезапуска cluster, вам также необходимо добавить следующий параметр конфигурации в файл конфигурации FE **fe.conf**:

   ```
   enable_legacy_compatibility_for_replication = true
   ```

После завершения миграции данных вам необходимо удалить конфигурацию `enable_legacy_compatibility_for_replication = true` из файла конфигурации и динамически отключить режим совместимости для репликации с помощью следующей команды:

```
ADMIN SET FRONTEND
CONFIG("enable_legacy_compatibility_for_replication"="false");
```

Настройка миграции данных (опционально)

Вы можете настроить операции миграции данных с помощью следующих параметров FE и BE. В большинстве случаев конфигурация по умолчанию может удовлетворить ваши потребности. Если вы хотите использовать конфигурацию по умолчанию, вы можете пропустить этот шаг.

> **ПРИМЕЧАНИЕ**
>
> Обратите внимание, что увеличение значений следующих параметров конфигурации может ускорить миграцию, но также увеличит нагрузку на исходный cluster.

Параметры FE

Следующие параметры FE являются динамическими параметрами конфигурации. О том, как их изменить, см. Настройка динамических параметров FE.

Параметр	По умолчанию	Единица	Описание
replication_max_parallel_table_count	100	-	Максимальное количество параллельных задач синхронизации данных. Selena создаёт одну задачу синхронизации для каждой таблицы.
replication_max_parallel_replica_count	10240	-	Максимальное количество tablet replica, разрешённых для параллельной синхронизации.
replication_max_parallel_data_size_mb	1048576	МБ	Максимальный размер данных, разрешённых для параллельной синхронизации.
replication_transaction_timeout_sec	86400	Секунды	Продолжительность таймаута для задач синхронизации.

Параметры BE

Следующий параметр BE является динамическим параметром конфигурации. О том, как его изменить, см. Настройка динамических параметров BE.

Параметр	По умолчанию	Единица	Описание
replication_threads	0	-	Количество потоков для выполнения задач синхронизации. `0` означает установку количества потоков равным 4-кратному количеству ядер CPU на машине, где находится BE.

2.5.2 Шаг 1: Установка инструмента

Рекомендуется устанавливать инструмент миграции на сервере, где находится целевой cluster.

1. Запустите терминал и загрузите бинарный пакет инструмента.

```
wget https://releases.selena.io/selena/selena-cluster-sync.tar.gz
```

2. Распакуйте пакет.

```
tar -xvzf selena-cluster-sync.tar.gz
```

2.5.3 Шаг 2: Настройка инструмента

Конфигурация, связанная с миграцией

Перейдите в распакованную папку и измените файл конфигурации **conf/sync.properties**.

```
cd selena-cluster-sync
vi conf/sync.properties
```

Содержимое файла следующее:

```
# Если true, все таблицы будут синхронизированы только один раз, и
программа автоматически завершится после завершения.
one_time_run_mode=false

source_fe_host=
source_fe_query_port=9030
source_cluster_user=root
source_cluster_password=
source_cluster_password_secret_key=
source_cluster_token=

target_fe_host=
target_fe_query_port=9030
target_cluster_user=root
target_cluster_password=
target_cluster_password_secret_key=

jdbc_connect_timeout_ms=30000
jdbc_socket_timeout_ms=60000

# Список имён баз данных или таблиц через запятую, например <db_name> или
<db_name.table_name>
# пример: db1,db2.tbl2,db3
# Порядок применения: 1. include 2. exclude
include_data_list=
exclude_data_list=

# Если нет особых требований, оставьте значения по умолчанию для следующих
конфигураций.
target_cluster_storage_volume=
target_cluster_replication_num=-1
target_cluster_max_disk_used_percent=80
# Для соответствия исходному cluster используйте null.
target_cluster_enable_persistent_index=

max_replication_data_size_per_job_in_gb=1024

meta_job_interval_seconds=180
meta_job_threads=4
ddl_job_interval_seconds=10
ddl_job_batch_size=10

# конфигурация таблиц
ddl_job_allow_drop_target_only=false
ddl_job_allow_drop_schema_change_table=true
```

```
ddl_job_allow_drop_inconsistent_partition=true
ddl_job_allow_drop_inconsistent_time_partition = true
ddl_job_allow_drop_partition_target_only=true
# конфигурация индексов
enable_bitmap_index_sync=false
ddl_job_allow_drop_inconsistent_bitmap_index=true
ddl_job_allow_drop_bitmap_index_target_only=true
# конфигурация MV
enable_materialized_view_sync=false
ddl_job_allow_drop_inconsistent_materialized_view=true
ddl_job_allow_drop_materialized_view_target_only=false
# конфигурация View
enable_view_sync=false
ddl_job_allow_drop_inconsistent_view=true
ddl_job_allow_drop_view_target_only=false

replication_job_interval_seconds=10
replication_job_batch_size=10
report_interval_seconds=300

enable_table_property_sync=false
```

Описание параметров следующее:

Параметр	Описание
one_time_run_mode	Включить ли режим однократной синхронизации. При включении режима однократной синхронизации инструмент миграции выполняет только полную синхронизацию вместо инкрементной синхронизации.
source_fe_host	IP-адрес или FQDN (полное доменное имя) FE исходного cluster.
source_fe_query_port	Порт запросов (`query_port`) FE исходного cluster.
source_cluster_user	Имя пользователя для входа в исходный cluster. Этому пользователю должна быть предоставлена привилегия OPERATE на уровне SYSTEM.
source_cluster_password	Пароль пользователя для входа в исходный cluster.
source_cluster_password_secret_key	Секретный ключ для шифрования пароля пользователя для входа в исходный cluster. Значение по умолчанию — пустая строка, что означает, что пароль не зашифрован. Если вы хотите зашифровать `source_cluster_password`, вы можете получить зашифрованную строку `source_cluster_password` с помощью SQL-выражения `SELECT TO_BASE64(AES_ENCRYPT('<source_cluster_password>','<source_cluster_password_ secret_key>'))`.

Параметр	Описание
source_cluster_token	Токен исходного cluster. О том, как получить токен cluster, см. Получение токена cluster ниже.
target_fe_host	IP-адрес или FQDN (полное доменное имя) FE целевого cluster.
target_fe_query_port	Порт запросов (`query_port`) FE целевого cluster.
target_cluster_user	Имя пользователя для входа в целевой cluster. Этому пользователю должна быть предоставлена привилегия OPERATE на уровне SYSTEM.
target_cluster_password	Пароль пользователя для входа в целевой cluster.
target_cluster_password_secret_key	Секретный ключ для шифрования пароля пользователя для входа в целевой cluster. Значение по умолчанию — пустая строка, что означает, что пароль не зашифрован. Если вы хотите зашифровать `target_cluster_password`, вы можете получить зашифрованную строку `target_cluster_password` с помощью SQL-выражения `SELECT TO_BASE64(AES_ENCRYPT('<target_cluster_password>','<target_cluster_password_ secret_key>'))`.
jdbc_connect_timeout_ms	Таймаут JDBC-подключения в миллисекундах для запросов FE. По умолчанию: `30000`.
jdbc_socket_timeout_ms	Таймаут JDBC-сокета в миллисекундах для запросов FE. По умолчанию: `60000`.
include_data_list	Базы данных и таблицы, которые необходимо мигрировать, с несколькими объектами, разделёнными запятыми (`,`). Например: `db1, db2.tbl2, db3`. Этот параметр имеет приоритет над `exclude_data_list`. Если вы хотите мигрировать все базы данных и таблицы в cluster, вам не нужно настраивать этот параметр.
exclude_data_list	Базы данных и таблицы, которые не нужно мигрировать, с несколькими объектами, разделёнными запятыми (`,`). Например: `db1, db2.tbl2, db3`. `include_data_list` имеет приоритет над этим параметром. Если вы хотите мигрировать все базы данных и таблицы в cluster, вам не нужно настраивать этот параметр.
target_cluster_storage_volume	Storage volume, используемый для хранения таблиц в целевом cluster, когда целевой cluster является cluster shared-data. Если вы хотите использовать storage volume по умолчанию, вам не нужно указывать этот параметр.
target_cluster_replication_num	Количество replica, указываемое при создании таблиц в целевом cluster. Если вы хотите использовать то же количество replica, что и в исходном cluster, вам не нужно указывать этот параметр.
target_cluster_max_disk_used_percent	Порог использования диска в процентах для узлов BE целевого cluster, когда целевой cluster является shared-nothing. Миграция прекращается, когда использование диска любого BE в целевом cluster превышает этот порог. Значение по умолчанию — `80`, что означает 80%.
meta_job_interval_seconds	Интервал в секундах, с которым инструмент миграции получает метаданные из исходного и целевого cluster. Вы можете использовать значение по умолчанию для этого параметра.

Параметр	Описание
meta_job_threads	Количество потоков, используемых инструментом миграции для получения метаданных из исходного и целевого cluster. Вы можете использовать значение по умолчанию для этого параметра.
ddl_job_interval_seconds	Интервал в секундах, с которым инструмент миграции выполняет DDL-операции на целевом cluster. Вы можете использовать значение по умолчанию для этого параметра.
ddl_job_batch_size	Размер пакета для выполнения DDL-операций на целевом cluster. Вы можете использовать значение по умолчанию для этого параметра.
ddl_job_allow_drop_target_only	Разрешить ли инструменту миграции удалять базы данных или таблицы, которые существуют только в целевом cluster, но не в исходном cluster. Значение по умолчанию — `false`, что означает, что они не будут удалены. Вы можете использовать значение по умолчанию для этого параметра.
ddl_job_allow_drop_schema_change_table	Разрешить ли инструменту миграции удалять таблицы с несогласованными схемами между исходным и целевым cluster. Значение по умолчанию — `true`, что означает, что они будут удалены. Вы можете использовать значение по умолчанию для этого параметра. Инструмент миграции автоматически синхронизирует удалённые таблицы во время миграции.
ddl_job_allow_drop_inconsistent_partition	Разрешить ли инструменту миграции удалять partition с несогласованным распределением данных между исходным и целевым cluster. Значение по умолчанию — `true`, что означает, что они будут удалены. Вы можете использовать значение по умолчанию для этого параметра. Инструмент миграции автоматически синхронизирует удалённые partition во время миграции.
ddl_job_allow_drop_partition_target_only	Разрешить ли инструменту миграции удалять partition, которые были удалены в исходном cluster, чтобы сохранить согласованность partition между исходным и целевым cluster. Значение по умолчанию — `true`, что означает, что они будут удалены. Вы можете использовать значение по умолчанию для этого параметра.
replication_job_interval_seconds	Интервал в секундах, с которым инструмент миграции запускает задачи синхронизации данных. Вы можете использовать значение по умолчанию для этого параметра.
replication_job_batch_size	Размер пакета, с которым инструмент миграции запускает задачи синхронизации данных. Вы можете использовать значение по умолчанию для этого параметра.
max_replication_data_size_per_job_in_gb	Пороговое значение размера данных, при котором инструмент миграции запускает задачи синхронизации данных. Единица измерения: ГБ. Если размер partition, которую нужно мигрировать, превышает это значение, будет запущено несколько задач синхронизации данных. Значение по умолчанию — `1024`. Вы можете использовать значение по умолчанию для этого параметра.
report_interval_seconds	Временной интервал, с которым инструмент миграции выводит информацию о прогрессе. Единица измерения: секунды. Значение по умолчанию: `300`. Вы можете использовать значение по умолчанию для этого параметра.
target_cluster_enable_persistent_index	Включить ли persistent index в целевом cluster. Если этот параметр не указан, целевой cluster соответствует исходному cluster.
ddl_job_allow_drop_inco	Разрешить ли инструменту миграции удалять partition с несогласованным временем между исходным и целевым cluster. Значение по умолчанию — `true`, что означает, что они будут

Параметр	Описание
nsistent_time_partition	удалены. Вы можете использовать значение по умолчанию для этого параметра. Инструмент миграции автоматически синхронизирует удалённые partition во время миграции.
enable_bitmap_index_sync	Включить ли синхронизацию Bitmap-индексов.
ddl_job_allow_drop_inconsistent_bitmap_index	Разрешить ли инструменту миграции удалять несогласованные Bitmap-индексы между исходным и целевым cluster. Значение по умолчанию — `true`, что означает, что они будут удалены. Вы можете использовать значение по умолчанию для этого параметра. Инструмент миграции автоматически синхронизирует удалённые индексы во время миграции.
ddl_job_allow_drop_bitmap_index_target_only	Разрешить ли инструменту миграции удалять Bitmap-индексы, которые были удалены в исходном cluster, чтобы сохранить согласованность индексов между исходным и целевым cluster. Значение по умолчанию — `true`, что означает, что они будут удалены. Вы можете использовать значение по умолчанию для этого параметра.
enable_materialized_view_sync	Включить ли синхронизацию материализованных представлений.
ddl_job_allow_drop_inconsistent_materialized_view	Разрешить ли инструменту миграции удалять несогласованные материализованные представления между исходным и целевым cluster. Значение по умолчанию — `true`, что означает, что они будут удалены. Вы можете использовать значение по умолчанию для этого параметра. Инструмент миграции автоматически синхронизирует удалённые материализованные представления во время миграции.
ddl_job_allow_drop_materialized_view_target_only	Разрешить ли инструменту миграции удалять материализованные представления, которые были удалены в исходном cluster, чтобы сохранить согласованность материализованных представлений между исходным и целевым cluster. Значение по умолчанию — `true`, что означает, что они будут удалены. Вы можете использовать значение по умолчанию для этого параметра.
enable_view_sync	Включить ли синхронизацию представлений (view).
ddl_job_allow_drop_inconsistent_view	Разрешить ли инструменту миграции удалять несогласованные представления между исходным и целевым cluster. Значение по умолчанию — `true`, что означает, что они будут удалены. Вы можете использовать значение по умолчанию для этого параметра. Инструмент миграции автоматически синхронизирует удалённые представления во время миграции.
ddl_job_allow_drop_view_target_only	Разрешить ли инструменту миграции удалять представления, которые были удалены в исходном cluster, чтобы сохранить согласованность представлений между исходным и целевым cluster. Значение по умолчанию — `true`, что означает, что они будут удалены. Вы можете использовать значение по умолчанию для этого параметра.
enable_table_property_sync	Включить ли синхронизацию свойств таблиц.

Получение токена cluster

Токен cluster доступен в метаданных FE. Войдите на сервер, где находится узел FE, и выполните следующую команду:

```
cat fe/meta/image/VERSION | grep token
```

Вывод:

```
token=wwwwwwww-xxxx-yyyy-zzzz-uuuuuuuuuu
```

Конфигурация сети (опционально)

Во время миграции данных инструменту миграции необходим доступ ко **всем** узлам FE как исходного, так и целевого cluster, а целевому cluster необходим доступ ко **всем** узлам BE и CN исходного cluster.

Вы можете получить сетевые адреса этих узлов, выполнив следующие команды в соответствующем cluster:

```
-- Получить сетевые адреса узлов FE в cluster.
SHOW FRONTENDS;
-- Получить сетевые адреса узлов BE в cluster.
SHOW BACKENDS;
-- Получить сетевые адреса узлов CN в cluster.
SHOW COMPUTE NODES;
```

Если эти узлы используют частные адреса, которые недоступны извне cluster, такие как внутренние сетевые адреса внутри cluster Kubernetes, вам необходимо сопоставить эти частные адреса с адресами, которые могут быть доступны извне.

Перейдите в распакованную папку инструмента и измените файл конфигурации **conf/hosts.properties**.

```
cd selena-cluster-sync
vi conf/hosts.properties
```

Содержимое файла по умолчанию следующее, описывающее, как настраивается сопоставление сетевых адресов:

```
# <SOURCE/TARGET>_<host>=<mappedHost>[;<srcPort>:<dstPort>[,<srcPort>:
<dstPort>...]]
```

> **ПРИМЕЧАНИЕ**
>
> `<host>` должен соответствовать адресу, показанному в столбце `IP`, возвращаемом командами `SHOW FRONTENDS`, `SHOW BACKENDS` или `SHOW COMPUTE NODES`.

Следующий пример выполняет эти операции:

1. Сопоставляет частные сетевые адреса исходного cluster `192.1.1.1` и `192.1.1.2` с `10.1.1.1` и `10.1.1.2`.
2. Сопоставляет порты FE исходного cluster `8030` и `9030` с `38030` и `39030` на `10.1.1.1`.
3. Сопоставляет частный сетевой адрес целевого cluster `fe-0.selena.svc.cluster.local` с `10.1.2.1` и перенаправляет порт `9030`.

```
# <SOURCE/TARGET>_<host>=<mappedHost>[;<srcPort>:<dstPort>[,<srcPort>:
<dstPort>...]]
SOURCE_192.1.1.1=10.1.1.1;8030:38030,9030:39030
SOURCE_192.1.1.2=10.1.1.2
TARGET_fe-0.selena.svc.cluster.local=10.1.2.1;9030:19030
```

2.5.4 Шаг 3: Запуск инструмента миграции

После настройки инструмента запустите инструмент миграции для начала процесса миграции данных.

```
./bin/start.sh
```

ПРИМЕЧАНИЕ

- Убедитесь, что узлы BE исходного и целевого cluster могут нормально обмениваться данными по сети.
- Во время работы инструмент миграции регулярно проверяет, отстают ли данные в целевом cluster от исходного cluster. Если есть отставание, он инициирует задачи миграции данных.
- Если новые данные постоянно загружаются в исходный cluster, синхронизация данных будет продолжаться до тех пор, пока данные в целевом cluster не станут согласованными с данными в исходном cluster.
- Вы можете запрашивать таблицы в целевом cluster во время миграции, но не загружайте новые данные в таблицы, так как это может привести к несогласованности между данными в целевом cluster и исходном cluster. В настоящее время инструмент миграции не запрещает загрузку данных в целевой cluster во время миграции.
- Обратите внимание, что миграция данных не завершается автоматически. Вам необходимо вручную проверить и подтвердить завершение миграции, а затем остановить инструмент миграции.

2.5.5 Просмотр прогресса миграции

Просмотр журналов инструмента миграции

Вы можете проверить прогресс миграции через журнал инструмента миграции **log/sync.INFO.log**.

Пример 1: Просмотр прогресса задачи.

```
24/01/18 11:17:04 INFO [Thread-1] updateClusterMeta(ClusterMetaKeeper.java:57): Target cluster metadata synchronization completed.
24/01/18 11:17:07 INFO [Thread-4] report(SyncJob.java:182): DDL job queue size: 0
24/01/18 11:17:07 INFO [Thread-4] report(SyncJob.java:184): Replication job queue size: 0
24/01/18 11:17:07 INFO [Thread-4] report(SyncJob.java:189): Total estimated size of data pending synchronization: 0
24/01/18 11:17:07 INFO [Thread-4] reportDbProgress(ClusterMetaKeeper.java:981): Databases pending synchronization: []
24/01/18 11:17:07 INFO [Thread-4] reportDbProgress(ClusterMetaKeeper.java:982): Databases that only exist in the target cluster: []
24/01/18 11:17:07 INFO [Thread-4] reportDbProgress(ClusterMetaKeeper.java:1001): Database test, source size: 320.000 B, target size: 320.000 B, diff: 0.000 B
24/01/18 11:17:07 INFO [Thread-4] reportDbProgress(ClusterMetaKeeper.java:1013): Database test , source running txn: 0, source finished txn: 0, target running txn: 0, target finished txn: 0
24/01/18 11:17:07 INFO [Thread-4] reportDbProgress(ClusterMetaKeeper.java:1001): Database commerce, source size: 131.634 GB, target size: 131.634 GB, diff: 0.000 B
24/01/18 11:17:07 INFO [Thread-4] reportDbProgress(ClusterMetaKeeper.java:1013): Database commerce , source running txn: 0, source finished txn: 0, target running txn: 0, target finished txn: 224
24/01/18 11:17:07 INFO [Thread-4] reportDbProgress(ClusterMetaKeeper.java:1017): Total source size: 131.634 GB, total target size: 131.634 GB, total diff: 0.000 B
24/01/18 11:17:07 INFO [Thread-4] reportDbProgress(ClusterMetaKeeper.java:1020): Total source running txn: 0, total source finished txn: 0, total target running txn: 0, total target finished txn: 224
24/01/18 11:17:07 INFO [Thread-4] report(SyncJob.java:226): Sync progress: 100.00%, total: 4, ddlPending: 0, jobPending: 0, running: 0, finished: 4, failed: 0, unknown: 0
```

Важные метрики следующие:

- `Sync job progress`: Прогресс миграции данных. Инструмент миграции регулярно проверяет, отстают ли данные в целевом cluster от исходного cluster. Поэтому прогресс 100% означает только то, что синхронизация данных завершена в текущем

интервале проверки. Если новые данные продолжают загружаться в исходный cluster, прогресс может уменьшиться в следующем интервале проверки.

- `total`: Общее количество всех типов заданий в этой операции миграции.
- `ddlPending`: Количество DDL-заданий, ожидающих выполнения.
- `jobPending`: Количество ожидающих заданий синхронизации данных для выполнения.
- `sent`: Количество заданий синхронизации данных, отправленных из исходного cluster, но ещё не запущенных. Теоретически это значение не должно быть слишком большим. Если значение продолжает расти, пожалуйста, свяжитесь с нашими инженерами.
- `running`: Количество заданий синхронизации данных, которые в данный момент выполняются.
- `finished`: Количество завершённых заданий синхронизации данных.
- `failed`: Количество неудачных заданий синхронизации данных. Неудачные задания синхронизации данных будут повторно отправлены. Поэтому в большинстве случаев вы можете игнорировать эту метрику. Если это значение значительно велико, пожалуйста, свяжитесь с нашими инженерами.
- `unknown`: Количество заданий с неизвестным статусом. Теоретически это значение всегда должно быть `0`. Если это значение не `0`, пожалуйста, свяжитесь с нашими инженерами.

Пример 2: Просмотр прогресса миграции таблиц.

```
24/08/30 14:52:00 INFO [sync-reporter] report(SyncJob.java:423): Sync table progress, finishedTableRatio: 0.00%, expiredTableRatio: 0.00%, total table: 0, finished table: 0, unfinished t
able: 0, expired table: 0.
24/08/30 14:57:00 INFO [sync-reporter] report(SyncJob.java:423): Sync table progress, finishedTableRatio: 50.00%, expiredTableRatio: 100.00%, total table: 2, finished table: 1, unfinishe
d table: 1, expired table: 2.
[starrocks@zhxthisdbl logs]$ grep finishedTableRatio lihktest_sync.INFO.log
24/08/30 14:52:00 INFO [sync-reporter] report(SyncJob.java:423): Sync table progress, finishedTableRatio: 0.00%, expiredTableRatio: 0.00%, total table: 0, finished table: 0, unfinished t
able: 0, expired table: 0.
24/08/30 14:57:00 INFO [sync-reporter] report(SyncJob.java:423): Sync table progress, finishedTableRatio: 50.00%, expiredTableRatio: 100.00%, total table: 2, finished table: 1, unfinishe
d table: 1, expired table: 2.
24/08/30 15:02:00 INFO [sync-reporter] report(SyncJob.java:423): Sync table progress, finishedTableRatio: 100.00%, expiredTableRatio: 0.00%, total table: 2, finished table: 2, unfinished
 table: 0, expired table: 0.
```

- `Sync table progress`: Прогресс миграции таблиц, то есть соотношение таблиц, которые были мигрированы в этой задаче миграции, ко всем таблицам, которые нужно мигрировать.
- `finishedTableRatio`: Соотношение таблиц с как минимум одним успешным выполнением задачи синхронизации.
- `expiredTableRatio`: Соотношение таблиц с устаревшими данными.
- `total table`: Общее количество таблиц, участвующих в этом процессе миграции данных.
- `finished table`: Количество таблиц с как минимум одним успешным выполнением задачи синхронизации.

- `unfinished table`: Количество таблиц без выполнения задачи синхронизации.
- `expired table`: Количество таблиц с устаревшими данными.

Просмотр статуса транзакций миграции

Инструмент миграции открывает транзакцию для каждой таблицы. Вы можете просмотреть статус миграции таблицы, проверив статус соответствующей транзакции.

```
SHOW PROC "/transactions/<db_name>/running";
```

`<db_name>` — это имя базы данных, в которой находится таблица.

Просмотр версий данных partition

Вы можете сравнить версии данных соответствующих partition в исходном и целевом cluster, чтобы просмотреть статус миграции этой partition.

```
SHOW PARTITIONS FROM <table_name>;
```

`<table_name>` — это имя таблицы, которой принадлежит partition.

Просмотр объёма данных

Вы можете сравнить объёмы данных в исходном и целевом cluster, чтобы просмотреть статус миграции.

```
SHOW DATA;
```

Просмотр количества строк таблицы

Вы можете сравнить количество строк таблиц в исходном и целевом cluster, чтобы просмотреть статус миграции каждой таблицы.

```
SELECT
  TABLE_NAME,
  TABLE_ROWS
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_TYPE = 'BASE TABLE'
ORDER BY TABLE_NAME;
```

2.5.6 Ограничения

Список объектов, которые в настоящее время поддерживают синхронизацию, следующий (не включённые означают, что синхронизация не поддерживается):

- Базы данных
- Внутренние таблицы и их данные
- Схемы материализованных представлений и их команды создания (Данные в материализованном представлении не будут синхронизированы. И если базовые таблицы материализованного представления не синхронизированы в целевой cluster, фоновая задача обновления материализованного представления сообщит об ошибке.)
- Логические представления (view)

2.6 HTTP-интерфейс

Для упрощения обслуживания cluster Selena предоставляет различные типы интерфейсов для операций и запросов. В этом разделе описываются эти HTTP-интерфейсы и их использование.

2.6.1 FE

Метод запроса	Путь запроса	Описание
PUT	`/api/{db}/{table}/_stream_load`	Операция Stream Load, подробности см. в Stream Load.
POST/PUT	`/api/transaction/{txn_op}`	Интерфейс транзакций Stream Load, подробности см. в Интерфейс транзакций Stream Load.
GET	`/api/{db}/_load_info`	
GET	`/api/_set_config?config_key1=config_value1`	Обновление конфигурации FE.
GET	`/api/_get_ddl?db={}&tbl={}`	Просмотр DDL-команды таблицы.
GET	`/api/_migration?db={}&tbl={}`	Просмотр информации о tablet таблицы.

Метод запроса	Путь запроса	Описание
GET	`/api/_check_storagetype`	
POST	`/api/{db}/{table}/_cancel?label={}`	
GET	`/api/{db}/get_load_state`	
GET	`/api/health`	
GET	`/metrics?type={core/json}`	Просмотр метрик текущего FE.
GET	`/api/show_meta_info`	
GET	`/api/show_proc`	
GET	`/api/show_runtime_info`	
HEAD/GET	`/api/get_log_file`	
GET	`/api/get_small_file`	
GET	`/api/rowcount`	
GET	`/api/check_decommission`	
GET	`/api/_meta_replay_state`	
POST	`/api/colocate/bucketseq`	
GET	`/api/colocate`	
POST	`/api/colocate/group_stable`	
POST	`/api/colocate/group_unstable`	
POST	`/api/colocate/update_group`	
POST	`/api/global_dict/table/enable`	
GET	`/api/profile?query_id={}`	Получение информации profile для указанного ID запроса.
GET	`/api/query_detail`	Получение деталей запроса. См. API деталей запроса.
GET	`/api/connection`	
GET	`/api/show_data?db={}`	Запрос размера указанной базы данных.
POST	`/api/query_dump`	Получение информации query dump, подробности см. в Query Dump.
GET	`/api/stop`	
GET	`/image`	
GET	`/info`	
GET	`/version`	
GET	`/put`	
GET	`/journal_id`	
GET	`/check`	
GET	`/dump`	

Метод запроса	Путь запроса	Описание
GET	`/role`	
GET	`/api/{db}/{table}/_count`	
GET	`/api/{db}/{table}/_schema`	Просмотр схемы таблицы.
GET/POST	`/api/{db}/{table}/_query_plan`	

2.6.2 BE

Метод HTTP-запроса	Путь HTTP-запроса	Описание
PUT	`/api/{db}/{table}/_stream_load`	Операция Stream Load, подробности см. в Stream Load.
POST/PUT	`/api/transaction/{txn_op}`	Интерфейс транзакций Stream Load, подробности см. в Интерфейс транзакций Stream Load.
PUT	`/api/transaction/load`	
HEAD/GET	`/api/_download_load`	
HEAD/GET	`/api/_tablet/_download`	
HEAD/GET	`/api/_load_error_log`	
GET	`/api/health`	
GET	`/api/_stop_be`	
GET	`/pprof/heap`	
GET	`/pprof/growth`	
GET	`/pprof/profile`	
GET	`/pprof/pmuprofile`	
GET	`/pprof/contention`	
GET	`/pprof/cmdline`	
HEAD/GET/POST	`/pprof/symbol`	
GET	`/metrics`	Просмотр метрик текущего BE.
HEAD	`/api/meta/header/{tablet_id}`	
GET	`/api/checksum`	
GET	`/api/reload_tablet`	
POST	`/api/restore_tablet`	
GET	`/api/snapshot`	
GET	`/api/compaction/show?tablet_id={}`	Просмотр информации о compaction для указанного tablet.

Метод HTTP-запроса	Путь HTTP-запроса	Описание
POST	`/api/compact?tablet_id={}&compaction_type={base/cumulative}`	Ручное выполнение compaction для указанного tablet.
GET	`/api/compaction/show_repair`	
PUT	`/api/compaction/submit_repair`	
POST	`/api/update_config`	Обновление конфигурации BE, подробности см. в Обновление конфигурации BE.
GET/PUT	`/api/runtime_filter_cache/{action}`	
POST	`/api/compact_rocksdb_meta`	
GET/PUT	`/api/query_cache/{action}`	
GET	`/api/pipeline_blocking_drivers/{action}`	
GET	`/greplog`	
GET	`/varz`	Просмотр текущей конфигурации BE.

2.6.3 CN

Метод запроса	Путь запроса	Описание
GET	`/api/health`	
GET	`/pprof/heap`	
GET	`/pprof/growth`	
GET	`/pprof/profile`	
GET	`/pprof/pmuprofile`	
GET	`/pprof/contention`	
GET	`/pprof/cmdline`	
HEAD/GET/POST	`/pprof/symbol`	
GET	`/metrics`	Просмотр метрик текущего CN.

2.6.4 API деталей запроса

API **query detail** возвращает детали выполнения недавних запросов, которые кэшируются в памяти FE.

> **ПРИМЕЧАНИЕ**
>
> Записи деталей запросов собираются только когда параметр конфигурации FE `enable_collect_query_detail_info` установлен в `true`.

Конечные точки

- `GET /api/query_detail` (v1)
- `GET /api/v2/query_detail` (v2, поддерживает `is_request_all_frontend` и возвращает обёрнутый результат)

Параметры

Имя	Обязательный	По умолчанию	Описание
`event_time`	Да	-	Фильтр нижней границы. Возвращает элементы, чей `eventTime` больше этого значения. Вы можете передать `0`, чтобы получить все кэшированные элементы.
`user`	Нет	-	Фильтр по полю `user`. Регистронезависимое сопоставление.
`is_request_all_frontend`	Нет (только v2)	`false`	Когда `true`, текущий FE запрашивает другие активные FE и объединяет их результаты.

Ответ

- v1 возвращает JSON-массив объектов `QueryDetail`.
- v2 возвращает `{ "code": "0", "message": "OK", "result": [ ... ] }`, где `result` — это список `QueryDetail`.

Аутентификация и авторизация

Этот API требует HTTP Basic аутентификации. Помимо успешного входа, дополнительная проверка привилегий не выполняется. Любой аутентифицированный пользователь может получить доступ к конечной точке и просматривать все кэшированные детали запросов, если не применён фильтр `user`.

Поля QueryDetail

Поле	Тип	Описание
`queryId`	string	ID запроса.
`eventTime`	long	Внутренняя временная метка для фильтрации. Монотонная временная метка в наносекундах.
`isQuery`	boolean	Является ли команда запросом.
`remoteIP`	string	IP-адрес клиента или `System`.

Поле	Тип	Описание
`connId`	int	ID соединения.
`startTime`	long	Время начала запроса в миллисекундах с начала эпохи.
`endTime`	long	Время окончания запроса в миллисекундах с начала эпохи. `-1`, если не завершён.
`latency`	long	Задержка запроса в миллисекундах. `-1`, если не завершён.
`pendingTime`	long	Время ожидания в миллисекундах.
`netTime`	long	Чистое время выполнения в миллисекундах.
`netComputeTime`	long	Чистое время вычислений в миллисекундах.
`state`	string	Одно из: `RUNNING`, `FINISHED`, `FAILED`, `CANCELLED`.
`database`	string	Текущая база данных.
`sql`	string	Текст SQL (может быть обезличен при соответствующей настройке).
`user`	string	Пользователь входа (полное имя пользователя).
`impersonatedUser`	string	Целевой пользователь `EXECUTE AS`. `null`, если не выполняется от имени другого пользователя.
`errorMessage`	string	Сообщение об ошибке при сбое.
`explain`	string	План выполнения (уровень контролируется параметром `query_detail_explain_level`).
`profile`	string	Текст profile, если собран.
`resourceGroupName`	string	Имя resource group.
`scanRows`	long	Количество просканированных строк.
`scanBytes`	long	Количество просканированных байт.
`returnRows`	long	Количество возвращённых строк.
`cpuCostNs`	long	Стоимость CPU в наносекундах.
`memCostBytes`	long	Стоимость памяти в байтах.
`spillBytes`	long	Байты, сброшенные на диск.
`cacheMissRatio`	float	Коэффициент промахов кэша в процентах (0-100).
`warehouse`	string	Имя warehouse.
`digest`	string	Дайджест SQL.
`catalog`	string	Имя каталога.
`command`	string	Имя команды MySQL.
`preparedStmtId`	string	ID подготовленного выражения.
`queryFeMemory`	long	Память FE, выделенная запросом, в байтах.
`querySource`	string	Источник запроса: `EXTERNAL`, `INTERNAL`, `MV` или `TASK`.

Примеры

v1

```
curl -u root: "http://<fe_host>:<fe_http_port>/api/query_detail?
event_time=0"
```

v2

```
curl -u root: "http://<fe_host>:<fe_http_port>/api/v2/query_detail?
event_time=0&is_request_all_frontend=true"
```

2.7 SQL Digest

В этой теме представлена функция SQL Digest в Selena. Эта функция поддерживается начиная с версии 3.3.6.

2.7.1 Обзор

SQL Digest - это отпечаток, генерируемый из исторических SQL-операторов с удалёнными параметрами. Он помогает группировать SQL-операторы с одинаковой структурой, но разными параметрами.

Типичные варианты использования SQL Digest включают:

- Поиск других SQL-операторов с такой же структурой, но разными параметрами в истории запросов
- Отслеживание частоты выполнения, совокупного времени и другой статистики SQL с одинаковой структурой
- Определение наиболее времязатратных шаблонов SQL в системе

В Selena SQL Digest в основном записываются через журналы аудита **fe.audit.log**. Например, выполните следующие два SQL-оператора:

```
SELECT count(*) FROM lineorder WHERE lo_orderdate > '19920101';
SELECT count(*) FROM lineorder WHERE lo_orderdate > '19920202';
```

В **fe.audit.log** будут сгенерированы два одинаковых Digest:

```
Digest=f58bb71850d112014f773717830e7f77
Digest=f58bb71850d112014f773717830e7f77
```

2.7.2 Использование

Предварительные требования

Чтобы включить эту функцию, вы должны установить параметр конфигурации FE `enable_sql_digest` в значение `true`.

Выполните следующий оператор, чтобы включить её динамически:

```
ADMIN SET FRONTEND CONFIG ('enable_sql_digest'='true');
```

Чтобы включить её на постоянной основе, вы должны добавить `enable_sql_digest = true` в файл конфигурации FE `fe.conf` и перезапустить FE.

После включения этой функции вы можете установить плагин AuditLoader для выполнения статистического анализа SQL-операторов.

Поиск похожих SQL

```
SELECT * FROM selena_audit_db__.selena_audit_tbl__
WHERE digest = '<Digest>'
LIMIT 1;
```

Отслеживание ежедневного количества выполнений и времени похожих SQL

```
SELECT
    date_trunc('day', `timestamp`) query_date,
    count(*),
    sum(queryTime),
    sum(scanRows),
    sum(cpuCostNs),
    sum(memCostBytes)
FROM selena_audit_db__.selena_audit_tbl__
WHERE digest = '<Digest>'
GROUP BY query_date
ORDER BY query_date
DESC LIMIT 30;
```

Вычисление среднего времени выполнения похожих SQL

```
SELECT avg(queryTime), min(queryTime), max(queryTime), stddev(queryTime)
FROM selena_audit_db__.selena_audit_tbl__
WHERE digest = '<Digest>';
```

Агрегация похожих SQL для анализа наиболее времязатратных шаблонов

```
WITH top_sql AS (
    SELECT digest, sum(queryTime)
    FROM selena_audit_db__.selena_audit_tbl__
    GROUP BY digest
    ORDER BY sum(queryTime)
    DESC LIMIT 10
)
SELECT * FROM selena_audit_db__.selena_audit_tbl__
WHERE digest IN (SELECT digest FROM top_sql);
```

2.7.3 Правила нормализации параметров

- Константные значения в SQL будут нормализованы. Например, похожие SQL-операторы с `WHERE a = 1` и `WHERE a = 2` будут иметь одинаковый Digest.
- Предикаты IN будут нормализованы. Например, похожие SQL-операторы с `IN (1,2,3)` и `IN (1,2)` будут иметь одинаковый Digest.

- Предложения `LIMIT N` будут нормализованы. Например, похожие SQL-операторы с `LIMIT 10` и `LIMIT 30` будут иметь одинаковый Digest.

2.8 Привилегии и аутентификация

2.8.1 Authentication

Нативная аутентификация

Создание и управление пользователями с использованием встроенной аутентификации Selena через SQL-команды.

Нативная аутентификация Selena — это метод аутентификации на основе пароля. Помимо этого, Selena также поддерживает интеграцию с внешними системами аутентификации, такими как LDAP. Подробнее см. в разделе Аутентификация с помощью Security Integration.

> **ПРИМЕЧАНИЕ**
>
> Пользователи с системной ролью `user_admin` могут создавать, изменять и удалять пользователей в Selena.

Создание пользователя

Вы можете создать пользователя, указав идентификатор пользователя, метод аутентификации и, опционально, роль по умолчанию. Чтобы включить нативную аутентификацию для пользователя, необходимо явно указать пароль в открытом или зашифрованном виде.

Следующий пример создает пользователя `jack`, разрешает ему подключаться только с IP-адреса `172.10.1.10`, включает нативную аутентификацию, устанавливает пароль `12345` в открытом виде и назначает ему роль `example_role` в качестве роли по умолчанию:

```
CREATE USER jack@'172.10.1.10' IDENTIFIED BY '12345' DEFAULT ROLE
'example_role';
```

> **ПРИМЕЧАНИЕ**
>
> - Selena шифрует пароли пользователей перед их сохранением. Вы можете получить зашифрованный пароль с помощью функции password().
> - Системная роль по умолчанию `PUBLIC` назначается пользователю, если роль по умолчанию не указана при создании пользователя.

Роль по умолчанию автоматически активируется при подключении пользователя к Selena. Инструкции по включению всех ролей (по умолчанию и назначенных) для пользователя после подключения см. в разделе Включение всех ролей.

Подробнее о создании пользователя см. в разделе CREATE USER.

Изменение пользователя

Вы можете изменить пароль, роль по умолчанию или свойство пользователя.

Инструкции по изменению роли по умолчанию см. в разделе Изменение роли по умолчанию.

Изменение свойства пользователя

Вы можете установить свойство пользователя с помощью ALTER USER.

Следующий пример устанавливает максимальное количество подключений для пользователя `jack` равным `1000`. Идентификаторы пользователей с одинаковым именем используют одно и то же свойство.

Поэтому достаточно установить свойство для `jack`, и эта настройка будет действовать для всех идентификаторов пользователей с именем `jack`.

```
ALTER USER 'jack' SET PROPERTIES ("max_user_connections" = "1000");
```

Сброс пароля пользователя

Вы можете сбросить пароль пользователя с помощью SET PASSWORD или ALTER USER.

> **ПРИМЕЧАНИЕ**

> - Любой пользователь может сбросить свой собственный пароль без каких-либо привилегий.
> - Только сам пользователь `root` может установить свой пароль. Если вы потеряли его пароль и не можете подключиться к Selena, см. раздел Сброс утерянного пароля root.

Оба следующих примера сбрасывают пароль пользователя `jack` на `54321`:

- Сброс пароля с помощью SET PASSWORD:

```
SET PASSWORD FOR jack@'172.10.1.10' = PASSWORD('54321');
```

- Сброс пароля с помощью ALTER USER:

```
ALTER USER jack@'172.10.1.10' IDENTIFIED BY '54321';
```

Сброс утерянного пароля root

Если вы потеряли пароль пользователя `root` и не можете подключиться к Selena, вы можете сбросить его, выполнив следующие действия:

1. Добавьте следующий параметр конфигурации в файлы конфигурации **fe/conf/fe.conf** на **всех узлах FE**, чтобы отключить аутентификацию пользователей:

```
enable_auth_check = false
```

2. Перезапустите **все узлы FE**, чтобы конфигурация вступила в силу.

```
./fe/bin/stop_fe.sh
./fe/bin/start_fe.sh
```

3. Подключитесь к Selena через MySQL-клиент под пользователем `root`. Указывать пароль не требуется, когда аутентификация пользователей отключена.

```
mysql -h <fe_ip_or_fqdn> -P<fe_query_port> -uroot
```

4. Сбросьте пароль для пользователя `root`.

```
SET PASSWORD for root = PASSWORD('xxxxxx');
```

5. Повторно включите аутентификацию пользователей, установив параметр конфигурации `enable_auth_check` в значение `true` в файлах конфигурации **fe/conf/fe.conf** на **всех узлах FE**.

```
enable_auth_check = true
```

6. Перезапустите **все узлы FE**, чтобы конфигурация вступила в силу.

```
./fe/bin/stop_fe.sh
./fe/bin/start_fe.sh
```

7. Подключитесь к Selena через MySQL-клиент, используя пользователя `root` и новый пароль, чтобы проверить успешность сброса пароля.

```
mysql -h <fe_ip_or_fqdn> -P<fe_query_port> -uroot -p<xxxxxx>
```

Удаление пользователя

Вы можете удалить пользователя с помощью DROP USER.

Следующий пример удаляет пользователя `jack`:

```
DROP USER jack@'172.10.1.10';
```

Просмотр пользователей

Вы можете просмотреть всех пользователей в cluster Selena с помощью SHOW USERS.

```
SHOW USERS;
```

Просмотр свойств пользователя

Вы можете просмотреть свойства пользователя с помощью SHOW PROPERTY.

Следующий пример показывает свойства пользователя `jack`:

```
SHOW PROPERTY FOR 'jack';
```

Или для просмотра конкретного свойства:

```
SHOW PROPERTY FOR 'jack' LIKE 'max_user_connections';
```

Аутентификация с Security Integration

Интегрируйте Selena с внешними системами аутентификации, используя security integration.

Создав security integration в вашем cluster Selena, вы можете предоставить доступ вашего внешнего сервиса аутентификации к Selena. С security integration вам не нужно вручную создавать пользователей в Selena. Когда пользователь пытается войти, используя внешнюю идентификацию, Selena будет использовать соответствующую security integration согласно конфигурации в `authentication_chain` для аутентификации пользователя. После успешной аутентификации и разрешения входа Selena создаёт виртуального пользователя в сессии для выполнения последующих операций.

Обратите внимание, что если вы используете интеграцию безопасности для настройки внешнего метода аутентификации, вы также должны интегрировать Selena с Apache Ranger, чтобы включить внешнюю авторизацию. В настоящее время интеграция Security Integration с встроенной авторизацией Selena не поддерживается.

Вы также можете включить Group Provider для Selena, чтобы получать доступ к информации о группах во внешних системах аутентификации, что позволяет создавать, аутентифицировать и авторизовывать группы пользователей в Selena.

Ручное создание и управление пользователями с внешними сервисами аутентификации также поддерживается в случае специфических крайних случаев. Для получения дополнительных инструкций см. См. также.

Создание security integration

В настоящее время интеграция безопасности Selena поддерживает следующие системы аутентификации:

- LDAP
- JSON Web Token (JWT)
- OAuth 2.0

> **ПРИМЕЧАНИЕ**
>
> Selena не выполняет проверку подключения при создании security integration.

Создание security integration с LDAP

Синтаксис

```
CREATE SECURITY INTEGRATION <security_integration_name>
PROPERTIES (
    "type" = "authentication_ldap_simple",
    "authentication_ldap_simple_server_host" = "",
    "authentication_ldap_simple_server_port" = "",
    "authentication_ldap_simple_bind_base_dn" = "",
    "authentication_ldap_simple_user_search_attr" = ""
    "authentication_ldap_simple_bind_root_dn" = "",
    "authentication_ldap_simple_bind_root_pwd" = "",
    "authentication_ldap_simple_ssl_conn_allow_insecure" = "{true |
false}",
    "authentication_ldap_simple_ssl_conn_trust_store_path" = "",
    "authentication_ldap_simple_ssl_conn_trust_store_pwd" = "",
    "comment" = ""
)
```

Параметры

security_integration_name

- Обязательно: Да
- Описание: Имя security integration.

  ПРИМЕЧАНИЕ

  Имя security integration глобально уникально. Вы не можете указать этот параметр как `native`.

type

- Обязательно: Да
- Описание: Тип security integration. Укажите значение `authentication_ldap_simple`.

authentication_ldap_simple_server_host

- Обязательно: Нет
- Описание: IP-адрес вашего сервиса LDAP. По умолчанию: `127.0.0.1`.

authentication_ldap_simple_server_port

- Обязательно: Нет
- Описание: Порт вашего сервиса LDAP. По умолчанию: `389`.

authentication_ldap_simple_bind_base_dn

- Обязательно: Да
- Описание: Базовое Distinguished Name (DN) пользователя LDAP, для которого cluster выполняет поиск.

authentication_ldap_simple_user_search_attr

- Обязательно: Да
- Описание: Атрибут пользователя, используемый для входа в сервис LDAP, например, `uid`.

> **ПРИМЕЧАНИЕ**
>
> **Механизм передачи DN**: LDAP security integration поддерживает функциональность передачи DN.
>
> - После успешной аутентификации система записывает как логин пользователя, так и полный DN.
> - При комбинировании с Group Provider информация DN автоматически передаётся в Group Provider.
> - Если `ldap_user_search_attr` не настроен для Group Provider, DN будет использоваться для сопоставления групп.
> - Этот механизм особенно подходит для сложных сред LDAP, таких как Microsoft AD.
>
> Для получения дополнительной информации см. механизм сопоставления DN в Аутентификация групп пользователей.

authentication_ldap_simple_bind_root_dn

- Обязательно: Да
- Описание: Admin DN вашего сервиса LDAP.

authentication_ldap_simple_bind_root_pwd

- Обязательно: Да
- Описание: Пароль администратора вашего сервиса LDAP.

authentication_ldap_simple_ssl_conn_allow_insecure

- Обязательно: Нет
- Описание: Разрешать ли незашифрованные соединения с сервером LDAP. Значение по умолчанию: `true`. Установка этого значения в `false` указывает, что для доступа к LDAP требуется шифрование SSL.

authentication_ldap_simple_ssl_conn_trust_store_path

- Обязательно: Нет
- Описание: Локальный путь для хранения SSL CA сертификата сервера LDAP. Поддерживает форматы pem и jks. Вам не нужно устанавливать этот параметр, если сертификат выдан доверенной организацией.

ldap_ssl_conn_trust_store_pwd

- Обязательно: Нет
- Описание: Пароль, используемый для доступа к локально хранящемуся SSL CA сертификату сервера LDAP. Сертификаты формата pem не требуют пароля. Только сертификаты формата jsk требуют пароль.

group_provider

- Обязательно: Нет
- Описание: Имя group provider(ов), которые будут объединены с security integration. Несколько group provider разделяются запятыми. После установки Selena будет записывать информацию о группах пользователя под каждым указанным провайдером при входе. Поддерживается начиная с v1.5.2. Подробные инструкции по включению Group Provider см. в Аутентификация групп пользователей.

permitted_groups

- Обязательно: Нет
- Описание: Имя группы (групп), членам которых разрешён вход в Selena. Несколько групп разделяются запятыми. Убедитесь, что указанные группы могут быть получены объединёнными group provider(ами). Поддерживается начиная с v1.5.2.

comment

- Обязательно: Нет
- Описание: Описание security integration.

Создание интеграции безопасности с JWT

Синтаксис

```
CREATE SECURITY INTEGRATION <security_integration_name>
PROPERTIES (
    "type" = "authentication_jwt",
    "jwks_url" = "",
    "principal_field" = "",
    "required_issuer" = "",
    "required_audience" = ""
    "comment" = ""
);
```

Параметры

security_integration_name

- Обязательный: Да
- Описание: Имя интеграции безопасности.

  ПРИМЕЧАНИЕ

  Имя интеграции безопасности глобально уникально. Вы не можете указать этот параметр как `native`.

type

- Обязательный: Да
- Описание: Тип интеграции безопасности. Укажите его как `jwt`.

jwks_url

- Обязательный: Да
- Описание: URL-адрес службы JSON Web Key Set (JWKS) или путь к локальному файлу в каталоге `fe/conf`.

principal_field

- Обязательный: Да
- Описание: Строка, используемая для идентификации поля, указывающего субъект (`sub`) в JWT. Значение по умолчанию — `sub`. Значение этого поля должно совпадать с именем пользователя для входа в Selena.

required_issuer

- Обязательный: Нет
- Описание: Список строк, используемых для идентификации издателей (`iss`) в JWT. JWT считается действительным, только если одно из значений в списке совпадает с издателем JWT.

required_audience

- Обязательный: Нет
- Описание: Список строк, используемых для идентификации аудитории (`aud`) в JWT. JWT считается действительным, только если одно из значений в списке совпадает с аудиторией JWT.

comment

- Обязательный: Нет

- Описание: Описание интеграции безопасности.

Создание интеграции безопасности с OAuth 2.0

Синтаксис

```
CREATE SECURITY INTEGRATION <security_integration_name>
PROPERTIES (
    "type" = "authentication_oauth2",
    "auth_server_url" = "",
    "token_server_url" = "",
    "client_id" = "",
    "client_secret" = "",
    "redirect_url" = "",
    "jwks_url" = "",
    "principal_field" = "",
    "required_issuer" = "",
    "required_audience" = ""
    "comment" = ""
)
```

Параметры

security_integration_name

- Обязательный: Да
- Описание: Имя интеграции безопасности.

  ПРИМЕЧАНИЕ

  Имя интеграции безопасности глобально уникально. Вы не можете указать этот параметр как `native`.

auth_server_url

- Обязательный: Да
- Описание: URL авторизации. URL-адрес, на который будет перенаправлен браузер пользователя для начала процесса авторизации OAuth 2.0.

token_server_url

- Обязательный: Да
- Описание: URL-адрес конечной точки на сервере авторизации, от которой Selena получает токен доступа.

client_id

- Обязательный: Да
- Описание: Публичный идентификатор клиента Selena.

client_secret

- Обязательный: Да
- Описание: Секрет, используемый для авторизации клиента Selena на сервере авторизации.

redirect_url

- Обязательный: Да
- Описание: URL-адрес, на который будет перенаправлен браузер пользователя после успешной аутентификации OAuth 2.0. Код авторизации будет отправлен на этот URL. В большинстве случаев его необходимо настроить как `http://<selena_fe_url>:<fe_http_port>/api/oauth2`.

type

- Обязательный: Да
- Описание: Тип интеграции безопасности. Укажите его как `authentication_oauth2`.

jwks_url

- Обязательный: Да
- Описание: URL-адрес службы JSON Web Key Set (JWKS) или путь к локальному файлу в каталоге `fe/conf`.

principal_field

- Обязательный: Да
- Описание: Строка, используемая для идентификации поля, указывающего субъект (`sub`) в JWT. Значение по умолчанию — `sub`. Значение этого поля должно совпадать с именем пользователя для входа в Selena.

required_issuer

- Обязательный: Нет
- Описание: Список строк, используемых для идентификации издателей (`iss`) в JWT. JWT считается действительным, только если одно из значений в списке совпадает с издателем JWT.

required_audience

- Обязательный: Нет
- Описание: Список строк, используемых для идентификации аудитории (`aud`) в JWT. JWT считается действительным, только если одно из значений в списке совпадает с аудиторией JWT.

comment

- Обязательный: Нет
- Описание: Описание интеграции безопасности.

Настройка цепочки аутентификации

После создания security integration она добавляется в ваш cluster Selena как новый метод аутентификации. Вы должны включить security integration, установив порядок методов аутентификации через динамический параметр конфигурации FE `authentication_chain`.

```
ADMIN SET FRONTEND CONFIG (
    "authentication_chain" = "<security_integration_name>[... ,]"
);
```

> **ПРИМЕЧАНИЕ**
>
> - Selena приоритизирует нативную аутентификацию для локальных пользователей. Если локального пользователя с таким же именем не существует, аутентификация выполняется в порядке, настроенном в `authentication_chain`. Если вход с использованием нативного метода аутентификации не удаётся, cluster попытается использовать следующий метод аутентификации в указанном порядке.
> - Вы можете указать несколько security integration в `authentication_chain`, за исключением OAuth 2.0 security integration. Вы не можете указать несколько OAuth 2.0 security integration или одну вместе с другими security integration.

Вы можете проверить значение `authentication_chain`, используя следующий запрос:

```
ADMIN SHOW FRONTEND CONFIG LIKE 'authentication_chain';
```

Управление security integrations

Изменение security integration

Вы можете изменить конфигурацию существующей security integration, используя следующий запрос:

```
ALTER SECURITY INTEGRATION <security_integration_name> SET
(
    "key"="value"[, ...]
)
```

> **ПРИМЕЧАНИЕ**
>
> Вы не можете изменить `type` security integration.

Удаление security integration

Вы можете удалить существующую security integration, используя следующий запрос:

```
DROP SECURITY INTEGRATION <security_integration_name>
```

Просмотр security integration

Вы можете просмотреть все security integration в вашем cluster, используя следующий запрос:

```
SHOW SECURITY INTEGRATIONS;
```

Пример:

```
SHOW SECURITY INTEGRATIONS;
+--------+--------+---------+
| Name   | Type   | Comment |
+--------+--------+---------+
| LDAP1  | LDAP   | NULL    |
+--------+--------+---------+
```

Параметр	Описание
Name	Имя security integration.
Type	Тип security integration.
Comment	Описание security integration. `NULL` возвращается, когда описание не указано для security integration.

Вы можете проверить детали security integration, используя следующий запрос:

```
SHOW CREATE SECURITY INTEGRATION <integration_name>
```

Пример:

```
SHOW CREATE SECURITY INTEGRATION LDAP1;

+----------------------+-----------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------+
| Security Integration  | Create Security Integration
|
+----------------------+-----------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------+
| LDAP1                | CREATE SECURITY INTEGRATION LDAP1
    PROPERTIES (
      "type" = "authentication_ldap_simple",
      "authentication_ldap_simple_server_host" = "",
      "authentication_ldap_simple_server_port" = "",
      "authentication_ldap_simple_bind_base_dn" = "",
      "authentication_ldap_simple_user_search_attr" = ""
      "authentication_ldap_simple_bind_root_dn" = "",
      "authentication_ldap_simple_bind_root_pwd" = "",
      "authentication_ldap_simple_ssl_conn_allow_insecure" = "{true |
false}",
      "authentication_ldap_simple_ssl_conn_trust_store_path" = "",
      "authentication_ldap_simple_ssl_conn_trust_store_pwd" = "",
      "comment" = ""
)|
+----------------------+-----------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
-----------------------------+
```

ПРИМЕЧАНИЕ

`ldap_bind_root_pwd` маскируется при выполнении SHOW CREATE SECURITY INTEGRATION.

Подключение к Selena через интеграцию безопасности

- Инструкции по подключению к Selena через LDAP см. в разделе Аутентификация LDAP - Подключение к Selena.
- Инструкции по подключению к Selena через JWT см. в разделе Аутентификация JSON Web Token - Подключение к Selena.
- Инструкции по подключению к Selena через OAuth 2.0 см. в разделе Аутентификация OAuth 2.0 - Подключение к Selena.

См. также

- Инструкции по ручной аутентификации пользователей через LDAP в Selena см. в разделе Аутентификация LDAP.
- Инструкции по ручной аутентификации пользователей через JSON Web Token в Selena см. в разделе Аутентификация JSON Web Token.
- Инструкции по ручной аутентификации пользователей через OAuth 2.0 в Selena см. в разделе Аутентификация OAuth 2.0.
- Инструкции по аутентификации групп пользователей см. в разделе Аутентификация групп пользователей.

Аутентификация LDAP

Помимо нативной аутентификации на основе пароля, Selena также поддерживает аутентификацию LDAP.

В этом разделе описывается, как вручную создавать и аутентифицировать пользователей с использованием LDAP в Selena. Инструкции по интеграции Selena с вашим сервисом LDAP с помощью security integration см. в разделе Аутентификация с помощью Security Integration. Подробнее об аутентификации групп пользователей в вашем сервисе LDAP см. в разделе Аутентификация групп пользователей.

Включение аутентификации LDAP

Для использования аутентификации LDAP необходимо сначала добавить сервис LDAP в конфигурацию узла FE.

```
# Добавьте IP-адрес сервиса LDAP.
authentication_ldap_simple_server_host =
# Добавьте порт сервиса LDAP, значение по умолчанию 389.
authentication_ldap_simple_server_port =
# Разрешить ли незашифрованные подключения к серверу LDAP. Значение по
умолчанию: `true`. Установка этого значения в `false` означает, что для
доступа к LDAP требуется шифрование SSL.
authentication_ldap_simple_ssl_conn_allow_insecure =
# Локальный путь для хранения SSL CA сертификата сервера LDAP.
Поддерживаются форматы pem и jks. Этот параметр не нужно устанавливать,
если сертификат выдан доверенной организацией.
authentication_ldap_simple_ssl_conn_trust_store_path =
# Пароль для доступа к локально хранящемуся SSL CA сертификату сервера
LDAP. Сертификаты в формате pem не требуют пароля. Только сертификаты в
формате jks требуют пароля.
authentication_ldap_simple_ssl_conn_trust_store_pwd =
```

Если вы хотите аутентифицировать пользователей путем прямого поиска в системе LDAP, вам необходимо **добавить следующие дополнительные параметры конфигурации**.

```
# Добавьте Base DN пользователя, указывающий диапазон поиска пользователя.
authentication_ldap_simple_bind_base_dn =
# Добавьте имя атрибута, идентифицирующего пользователя в объекте LDAP. По
умолчанию: uid.
authentication_ldap_simple_user_search_attr =
# Добавьте DN администратора для поиска пользователей.
authentication_ldap_simple_bind_root_dn =
# Добавьте пароль администратора для поиска пользователей.
authentication_ldap_simple_bind_root_pwd =
```

Механизм сопоставления DN

Начиная с версии v1.5.2, Selena поддерживает запись и передачу информации Distinguished Name (DN) пользователя во время аутентификации LDAP для более точного разрешения групп.

Принцип работы

1. **Фаза аутентификации**: LDAPAuthProvider записывает обе части информации после успешной аутентификации пользователя:
    - Имя пользователя для входа (для традиционного сопоставления групп)

- Полный DN пользователя (для сопоставления групп на основе DN)

2. **Фаза разрешения групп**: LDAPGroupProvider определяет стратегию сопоставления на основе конфигурации параметра `ldap_user_search_attr`:
   - **Когда `ldap_user_search_attr` настроен**, используется имя пользователя в качестве ключа для сопоставления групп.
   - **Когда `ldap_user_search_attr` не настроен**, используется DN в качестве ключа для сопоставления групп.

Сценарии использования

- **Традиционная среда LDAP**: Члены группы используют простые имена пользователей (например, атрибут `cn`). Администраторам необходимо настроить `ldap_user_search_attr`.
- **Среда Microsoft AD**: Члены группы могут не иметь атрибутов имени пользователя. `ldap_user_search_attr` настроить невозможно. Система будет использовать DN напрямую для сопоставления.
- **Смешанная среда**: Поддерживается гибкое переключение между обоими методами сопоставления.

Создание пользователя с LDAP

При создании пользователя укажите метод аутентификации как аутентификацию LDAP с помощью `IDENTIFIED WITH authentication_ldap_simple AS 'xxx'`. xxx — это DN (Distinguished Name) пользователя в LDAP.

Пример 1:

```
CREATE USER tom IDENTIFIED WITH authentication_ldap_simple AS
'uid=tom,ou=company,dc=example,dc=com'
```

Можно создать пользователя без указания DN пользователя в LDAP. При входе пользователя Selena выполнит поиск информации о пользователе в системе LDAP. Если найдено одно и только одно совпадение, аутентификация успешна.

Пример 2:

```
CREATE USER tom IDENTIFIED WITH authentication_ldap_simple
```

В этом случае необходимо добавить дополнительную конфигурацию в FE:

- `authentication_ldap_simple_bind_base_dn`: Base DN пользователя, указывающий диапазон поиска пользователя.
- `authentication_ldap_simple_user_search_attr`: Имя атрибута в объекте LDAP, идентифицирующего пользователя, по умолчанию uid.
- `authentication_ldap_simple_bind_root_dn`: DN учетной записи администратора, используемой для поиска информации о пользователе.
- `authentication_ldap_simple_bind_root_pwd`: Пароль учетной записи администратора, используемой при поиске информации о пользователе.

Аутентификация пользователей

Аутентификация LDAP требует, чтобы клиент передавал пароль в открытом виде в Selena. Существует три способа передачи пароля в открытом виде:

Подключение из MySQL-клиента с LDAP

Добавьте `--default-auth mysql_clear_password --enable-cleartext-plugin` при выполнении:

```
mysql -utom -P9030 -h127.0.0.1 -p --default-auth mysql_clear_password --
enable-cleartext-plugin
```

Подключение из JDBC/ODBC клиента с LDAP

- **JDBC**

Обратите внимание, что при использовании JDBC-соединений необходимо включить SSL на стороне сервера. Дополнительную информацию см. в разделе SSL-аутентификация.

JDBC 5:

```
Properties properties = new Properties();
properties.put("authenticationPlugins",
"com.mysql.jdbc.authentication.MysqlClearPasswordPlugin");
properties.put("defaultAuthenticationPlugin",
"com.mysql.jdbc.authentication.MysqlClearPasswordPlugin");
properties.put("disabledAuthenticationPlugins",
"com.mysql.jdbc.authentication.MysqlNativePasswordPlugin");
```

JDBC 8:

```
Properties properties = new Properties();
properties.put("authenticationPlugins",
"com.mysql.cj.protocol.a.authentication.MysqlClearPasswordPlugin");
properties.put("defaultAuthenticationPlugin",
"com.mysql.cj.protocol.a.authentication.MysqlClearPasswordPlugin");
properties.put("disabledAuthenticationPlugins",
"com.mysql.cj.protocol.a.authentication.MysqlNativePasswordPlugin");
```

- **ODBC**

Добавьте `default\_auth=mysql_clear_password` и `ENABLE_CLEARTEXT\_PLUGIN=1` в DSN ODBC, а также имя пользователя и пароль.

Аутентификация JSON Web Token

В этом разделе описывается, как включить аутентификацию JSON Web Token в Selena.

Начиная с версии v1.5.2, Selena поддерживает аутентификацию клиентского доступа с использованием JSON Web Tokens.

JSON Web Token (JWT) — это открытый стандарт (RFC 7519), который определяет компактный и самодостаточный способ безопасной передачи информации между сторонами в виде JSON-объекта. Эта информация может быть проверена и считается доверенной, поскольку она имеет цифровую подпись. JWT могут быть подписаны с использованием секрета (с алгоритмом HMAC) или пары открытого/закрытого ключа с использованием RSA или ECDSA.

В этом разделе описывается, как вручную создавать и аутентифицировать пользователей с использованием JWT в Selena. Инструкции по интеграции Selena с аутентификацией JWT с помощью security integration см. в разделе Аутентификация с помощью Security Integration. Подробнее об аутентификации групп пользователей с использованием JWT см. в разделе Аутентификация групп пользователей.

Предварительные требования

Если вы хотите подключиться к Selena из MySQL-клиента, версия MySQL-клиента должна быть 9.2 или выше.

Создание пользователя с JWT

При создании пользователя укажите метод аутентификации как JWT с помощью `IDENTIFIED WITH authentication_jwt [AS '{xxx}']`. `{xxx}` — это свойства JWT пользователя. Помимо следующего метода, вы можете настроить свойства JWT по умолчанию в файле конфигурации FE. Вам необходимо вручную изменить все файлы **fe.conf** и перезапустить все FE, чтобы конфигурация вступила в силу. После установки конфигурации FE, Selena будет использовать свойства по умолчанию, указанные в вашем файле конфигурации, и вы можете опустить часть `AS '{xxx}'`.

Синтаксис:

```
CREATE USER <username> IDENTIFIED WITH authentication_jwt [AS
'{
  "jwks_url": "<jwks_url>",
  "principal_field": "<principal_field>",
  "required_issuer": "<required_issuer>",
  "required_audience": "<required_audience>"
}']
```

Свойство	Соответствующая конфигурация FE	Описание
`jwks_url`	`jwt_jwks_url`	URL к сервису JSON Web Key Set (JWKS) или путь к локальному файлу открытого ключа в директории `fe/conf`.
`principal_field`	`jwt_principal_field`	Строка, используемая для идентификации поля субъекта (`sub`) в JWT. Значение по умолчанию — `sub`. Значение этого поля должно совпадать с именем пользователя для входа в Selena.
`required_issuer`	`jwt_required_issuer`	(Опционально) Список строк для идентификации издателей (`iss`) в JWT. JWT считается действительным, только если одно из значений в списке совпадает с издателем JWT.
`required_audience`	`jwt_required_audience`	(Опционально) Список строк для идентификации аудитории (`aud`) в JWT. JWT считается действительным, только если одно из значений в списке совпадает с аудиторией JWT.

Пример:

```
CREATE USER tom IDENTIFIED WITH authentication_jwt AS
'{
  "jwks_url": "http://localhost:38080/realms/master/protocol/jwt/certs",
  "principal_field": "preferred_username",
  "required_issuer": "http://localhost:38080/realms/master",
  "required_audience": "selena"
}';
```

Если вы установили свойства JWT в файлах конфигурации FE, вы можете напрямую выполнить следующий оператор:

```
CREATE USER tom IDENTIFIED WITH authentication_jwt;
```

Подключение из MySQL-клиента с JWT

Для подключения к Selena из MySQL-клиента с использованием JWT необходимо включить плагин `authentication_openid-connect_client` и передать необходимый токен (используя путь к файлу токена) для аутентификации сопоставленного пользователя.

Синтаксис:

```
mysql -h <hostname> -P <query_port> --authentication-openid-connect-client-id-token-file=<path_to_token_file> -u <username>
```

Пример:

```
mysql -h 127.0.0.1 -P 9030 --authentication-openid-connect-client-id-token-file=/path/to/token/file -u tom
```

Аутентификация OAuth 2.0

В этом разделе описывается, как включить аутентификацию OAuth 2.0 в Selena.

Начиная с версии v1.5.2, Selena поддерживает аутентификацию клиентского доступа с использованием OAuth 2.0. Вы можете включить аутентификацию OAuth 2.0 через HTTP для Web UI и драйвера JDBC.

Selena использует поток Authorization Code, который обменивает код авторизации на токен. В общем случае поток включает следующие шаги:

1. Координатор Selena перенаправляет браузер пользователя на сервер авторизации.
2. Пользователь аутентифицируется на сервере авторизации.
3. После одобрения запроса браузер перенаправляется обратно на Selena FE с кодом авторизации.
4. Координатор Selena обменивает код авторизации на токен.

В этом разделе описывается, как вручную создавать и аутентифицировать пользователей с использованием OAuth 2.0 в Selena. Инструкции по интеграции Selena с вашим сервисом OAuth 2.0 с помощью security integration см. в разделе Аутентификация с помощью Security Integration. Подробнее об аутентификации групп пользователей в вашем сервисе OAuth 2.0 см. в разделе Аутентификация групп пользователей.

Предварительные требования

Если вы хотите подключиться к Selena из MySQL-клиента, версия MySQL-клиента должна быть 9.2 или выше. Подробнее см. в официальной документации MySQL.

Создание пользователя с OAuth 2.0

При создании пользователя укажите метод аутентификации как OAuth 2.0 с помощью `IDENTIFIED WITH authentication_oauth2 [AS '{xxx}']`. `{xxx}` — это свойства OAuth 2.0 пользователя. Помимо следующего метода, вы можете настроить свойства OAuth 2.0 по умолчанию в файле конфигурации FE. Вам необходимо вручную изменить все файлы **fe.conf** и перезапустить все FE, чтобы конфигурация вступила в силу. После установки конфигурации FE, Selena будет использовать свойства по умолчанию, указанные в вашем файле конфигурации, и вы можете опустить часть `AS '{xxx}'`.

Синтаксис:

```
CREATE USER <username> IDENTIFIED WITH authentication_oauth2 [AS
'{
  "auth_server_url": "<auth_server_url>",
  "token_server_url": "<token_server_url>",
  "client_id": "<client_id>",
  "client_secret": "<client_secret>",
  "redirect_url": "<redirect_url>",
  "jwks_url": "<jwks_url>",
  "principal_field": "<principal_field>",
  "required_issuer": "<required_issuer>",
  "required_audience": "<required_audience>"
}']
```

Свойство	Соответствующая конфигурация FE	Описание
`auth_server_url`	`oauth2_auth_server_url`	URL авторизации. URL, на который будет перенаправлен браузер пользователя для начала процесса авторизации OAuth 2.0.
`token_server_url`	`oauth2_token_server_url`	URL конечной точки на сервере авторизации, с которой Selena получает токен доступа.
`client_id`	`oauth2_client_id`	Публичный идентификатор клиента Selena.
`client_secret`	`oauth2_client_secret`	Секрет, используемый для авторизации клиента Selena на сервере авторизации.
`redirect_url`	`oauth2_redirect_url`	URL, на который будет перенаправлен браузер пользователя после успешной аутентификации OAuth 2.0. Код авторизации будет отправлен на этот URL. В большинстве случаев его необходимо настроить как `http://<selena_fe_url>:<fe_http_port>/api/oauth2`.
`jwks_url`	`oauth2_jwks_url`	URL к сервису JSON Web Key Set (JWKS) или путь к локальному файлу в директории `conf`.
`principal_field`	`oauth2_principal_field`	Строка, используемая для идентификации поля субъекта (`sub`) в JWT. Значение по умолчанию — `sub`. Значение этого поля должно совпадать с именем пользователя для входа в Selena.
`required_issuer`	`oauth2_required_issuer`	(Опционально) Список строк для идентификации издателей (`iss`) в JWT. JWT считается действительным, только если одно из значений в списке совпадает с издателем JWT.

Свойство	Соответствующая конфигурация FE	Описание
`required_audience`	`oauth2_required_audience`	(Опционально) Список строк для идентификации аудитории (`aud`) в JWT. JWT считается действительным, только если одно из значений в списке совпадает с аудиторией JWT.

Пример:

```
CREATE USER tom IDENTIFIED WITH authentication_oauth2 AS
'{
  "auth_server_url": "http://localhost:38080/realms/master/protocol/openid-connect/auth",
  "token_server_url": "http://localhost:38080/realms/master/protocol/openid-connect/token",
  "client_id": "12345",
  "client_secret": "LsWyD9vPcM3LHxLZfzJsuoBwWQFBLcoR",
  "redirect_url": "http://localhost:8030/api/oauth2",
  "jwks_url": "http://localhost:38080/realms/master/protocol/openid-connect/certs",
  "principal_field": "preferred_username",
  "required_issuer": "http://localhost:38080/realms/master",
  "required_audience": "12345"
}';
```

Если вы установили свойства OAuth 2.0 в файлах конфигурации FE, вы можете напрямую выполнить следующий оператор:

```
CREATE USER tom IDENTIFIED WITH authentication_oauth2;
```

Подключение из JDBC-клиента с OAuth 2.0

Selena поддерживает протокол MySQL. Вы можете настроить плагин MySQL для автоматического запуска метода входа через браузер.

Пример кода плагина JDBC OAuth2 см. в официальной документации selena-jdbc-oauth2-plugin.

Подключение из MySQL-клиента с OAuth 2.0

Если у вас нет доступа к браузеру в вашей среде (например, при использовании терминала или сервера), вы также можете получить доступ к Selena через нативный MySQL-клиент или драйвер JDBC:

- При первом подключении к Selena будет возвращен URL.
- Вам необходимо перейти по этому URL в браузере и завершить аутентификацию.
- После аутентификации вы сможете взаимодействовать с Selena.

2.8.2 Authorization

Обзор привилегий

Эта тема описывает базовые концепции системы привилегий Selena. Привилегии определяют, какие пользователи могут выполнять какие операции над какими объектами, что позволяет более безопасно управлять данными и ресурсами с детальной настройкой.

> ПРИМЕЧАНИЕ: Привилегии, описанные в этой теме, доступны только начиная с версии v1.5.2. Структура привилегий и синтаксис в v1.5.2 не совместимы с предыдущими версиями. После обновления до v1.5.2 большинство ваших исходных привилегий сохраняются, за исключением привилегий для определенных операций. Подробные различия см. в разделе [Примечания по обновлению] в Привилегии, поддерживаемые в Selena.

Selena использует две модели привилегий:

- Управление доступом на основе ролей (RBAC): Привилегии назначаются ролям, которые затем назначаются пользователям. В этом случае привилегии передаются пользователям через роли.
- Управление доступом на основе идентификации (IBAC): Привилегии назначаются непосредственно идентификаторам пользователей.

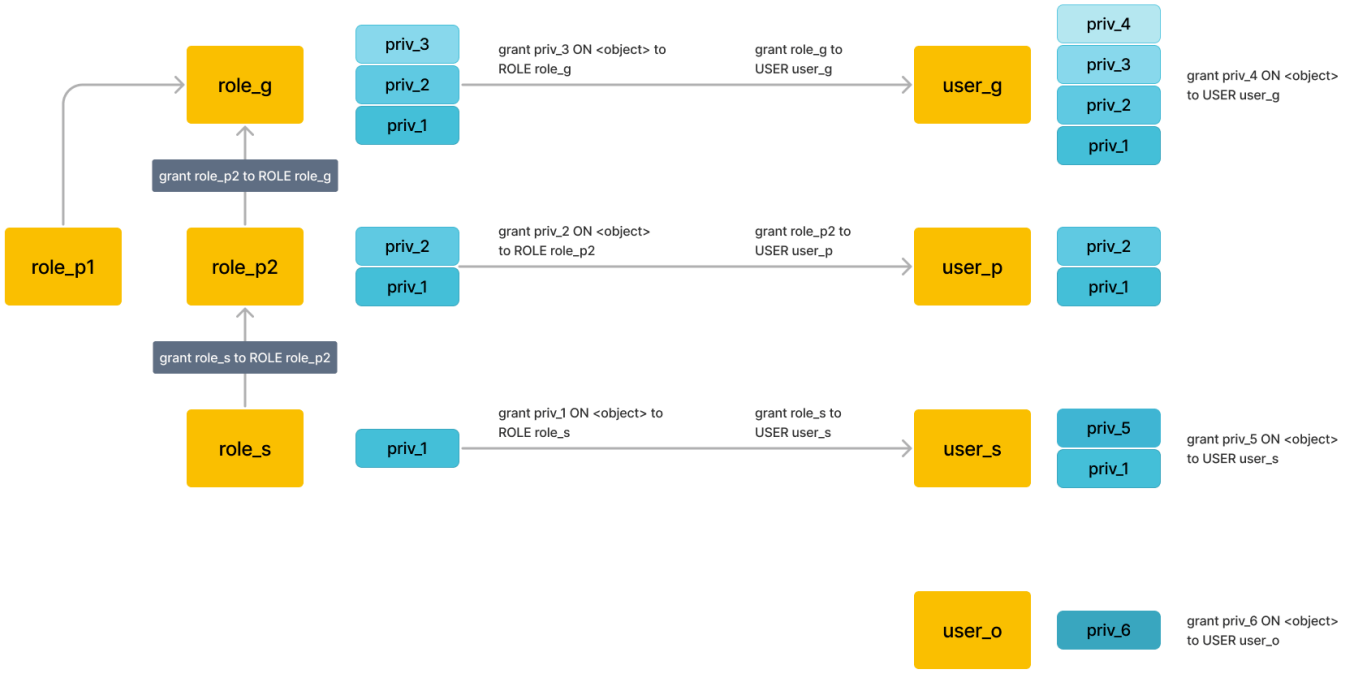
Таким образом, максимальная область привилегий каждой идентификации пользователя является объединением её собственных привилегий и привилегий ролей, назначенных этой идентификации пользователя.

Базовые концепции для понимания системы привилегий Selena:

- **Объект**: Сущность, к которой может быть предоставлен доступ. Если доступ не предоставлен, он запрещен. Примеры объектов включают CATALOG, DATABASE, TABLE и VIEW. Для получения дополнительной информации см. Привилегии, поддерживаемые в Selena.
- **Привилегия**: Определенный уровень доступа к объекту. Несколько привилегий можно использовать для управления степенью детализации доступа, предоставляемого к объекту. Привилегии специфичны для объекта. Разные объекты могут иметь разные привилегии. Примеры привилегий включают SELECT, ALTER и DROP.
- **Идентификация пользователя**: уникальная идентификация пользователя, а также сущность, которой могут быть предоставлены привилегии. Идентификация пользователя представлена как `username@'userhost'`, состоящая из имени пользователя и IP-адреса, с которого пользователь входит в систему. Использование идентификации упрощает настройку атрибутов. Идентификации пользователей, которые имеют одинаковое имя пользователя, имеют одинаковый атрибут. Если вы настраиваете атрибут для имени пользователя, этот атрибут применяется ко всем идентификациям пользователей, которые имеют это имя пользователя.
- **Роль**: Сущность, которой могут быть предоставлены привилегии. Роли представляют собой абстрактную коллекцию привилегий. Роли в свою очередь могут быть назначены пользователям. Роли также могут быть назначены другим ролям, создавая иерархию ролей. Для упрощения управления данными Selena предоставляет системные роли. Для большей гибкости вы также можете создавать пользовательские роли в соответствии с бизнес-требованиями.

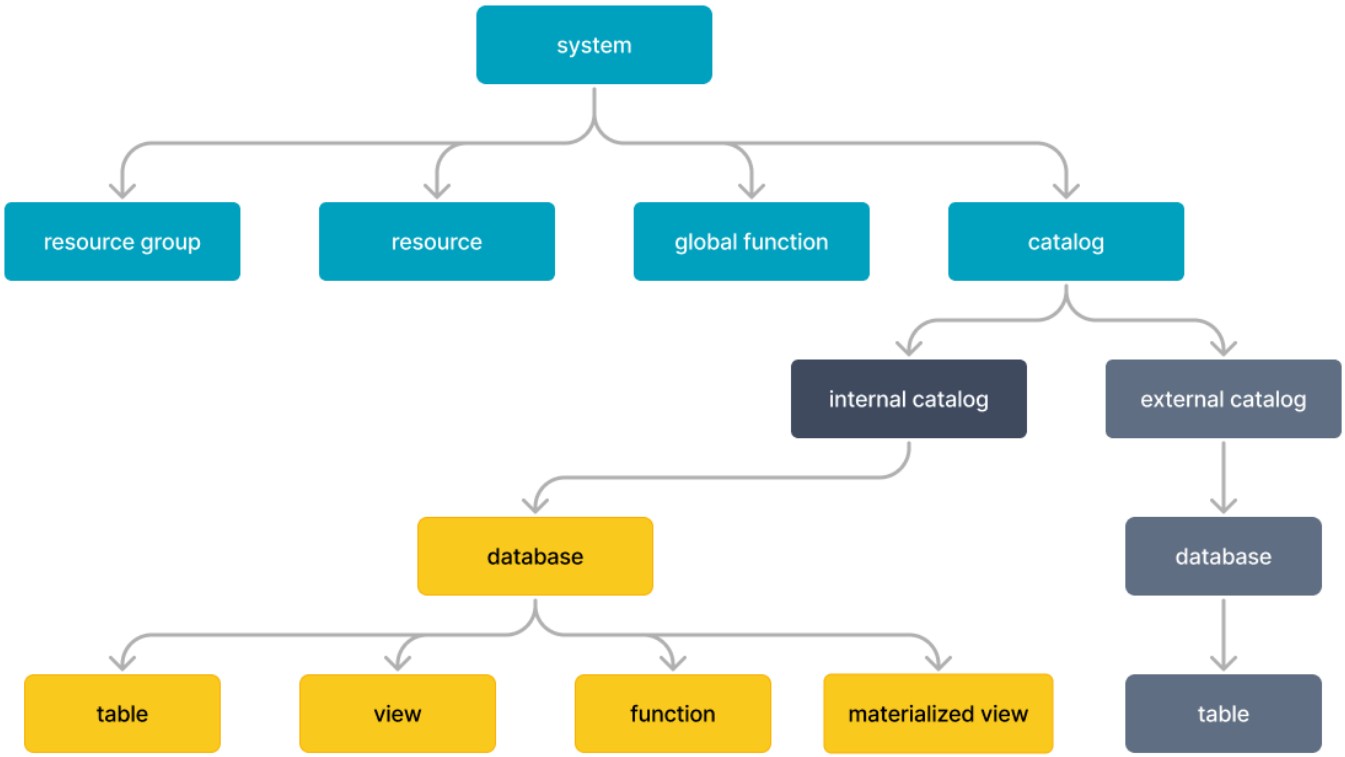
На следующем рисунке показан пример управления привилегиями в рамках моделей привилегий RBAC и IBAC.

В моделях доступ к объектам разрешается через привилегии, назначенные ролям и пользователям. Роли в свою очередь назначаются другим ролям или пользователям.



Объекты и привилегии

Объекты имеют логическую иерархию, которая связана с концепцией, которую они представляют. Например, Database содержится в Catalog, а Table, View, Materialized View и Function содержатся в Database. На следующем рисунке показана иерархия объектов в системе Selena.



Каждый объект имеет набор элементов привилегий, которые могут быть предоставлены. Эти привилегии определяют, какие операции могут быть выполнены над этими объектами. Вы можете предоставлять и отзывать привилегии у ролей или пользователей с помощью команд GRANT и REVOKE.

Пользователи

Идентификация пользователя

В Selena каждый пользователь идентифицируется уникальным ID пользователя. Он состоит из IP-адреса (user host) и имени пользователя в формате `username@'userhost'`. Selena идентифицирует пользователей с одинаковым именем пользователя, но с разных IP-адресов, как разные идентификации пользователей. Например, `user1@'172.65.xx.1'` и `user1@'172.65.xx.2'` - это две идентификации пользователей.

Другое представление идентификации пользователя - это `username@['domain']`, где `domain` - это доменное имя, которое может быть разрешено через DNS как набор IP-адресов. `username@['domain']` в конечном итоге представлено как набор `username@'userhost'`. Вы можете использовать `%` для части `userhost` для нечеткого поиска. Если `userhost` не указан, по умолчанию используется `'%'`, что означает пользователей с одинаковым именем, вошедших с любого хоста.

Предоставление привилегий пользователям

Пользователи являются сущностями, которым могут быть предоставлены привилегии. Как привилегии, так и роли могут быть назначены пользователям. Максимальная область привилегий каждой идентификации пользователя является объединением её собственных привилегий и привилегий ролей, назначенных этой идентификации пользователя. Selena гарантирует, что каждый пользователь может выполнять только авторизованные операции.

Мы рекомендуем **использовать роли для передачи привилегий** в большинстве случаев. Например, после создания роли вы можете предоставить привилегии роли, а затем назначить роль пользователям. Если вы хотите предоставить временные или специальные привилегии, вы можете напрямую предоставить их пользователям. Это упрощает управление привилегиями и обеспечивает гибкость.

Роли

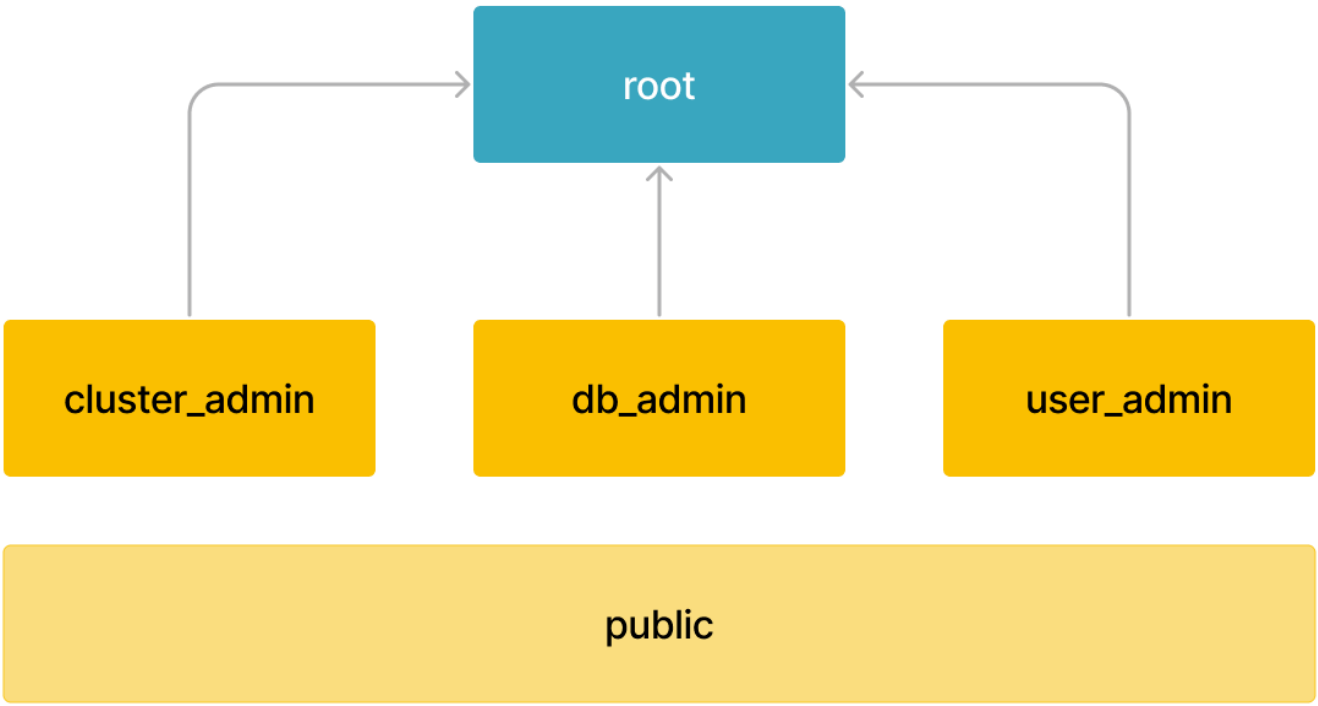
Роли - это сущности, которым могут быть предоставлены и отозваны привилегии. Роли можно рассматривать как коллекцию привилегий, которые могут быть назначены пользователям, чтобы позволить им выполнять необходимые действия. Пользователю может быть назначено несколько ролей, чтобы он мог выполнять различные действия с использованием отдельных наборов привилегий. Для упрощения управления Selena рекомендует **управлять привилегиями через роли**. Специальные и временные привилегии могут быть предоставлены непосредственно пользователям.

Для облегчения управления Selena предоставляет несколько **системных ролей** с определенными привилегиями, которые помогают вам удовлетворять требования повседневного управления и обслуживания. Вы также можете гибко **настраивать роли** для удовлетворения конкретных бизнес-потребностей и требований безопасности. Обратите внимание, что область привилегий системных ролей не может быть изменена.

После активации роли пользователи могут выполнять операции, авторизованные ролью. Вы можете установить **роли по умолчанию**, которые автоматически активируются при входе пользователя в систему. Пользователи также могут вручную активировать роль, принадлежащую этому пользователю, в текущей сессии.

Системные роли

Selena предоставляет несколько типов системных ролей.



- `root`: имеет глобальные привилегии. По умолчанию пользователь `root` имеет роль `root`. После создания кластера Selena система автоматически генерирует пользователя root с привилегиями root. Поскольку пользователь и роль root имеют все привилегии системы, мы рекомендуем создать новых пользователей и роли для последующих операций, чтобы предотвратить любые рискованные операции. Храните пароль пользователя root в безопасности.

- `cluster_admin`: имеет привилегии управления кластером для выполнения операций, связанных с узлами, таких как добавление или удаление узлов. `cluster_admin` имеет привилегии добавлять, удалять и выводить из эксплуатации узлы кластера. Мы рекомендуем не назначать `cluster_admin` или любые пользовательские роли, содержащие эту роль, в качестве роли по умолчанию для любого пользователя, чтобы предотвратить непредвиденные изменения узлов.

- `db_admin`: имеет привилегии управления базой данных, включая привилегии на выполнение всех операций с catalog, database, table, view, materialized view, function, global function, resource group и плагинами.

- `user_admin`: имеет административные привилегии на пользователей и роли, включая привилегии создавать пользователей, роли и привилегии.

Вышеуказанные системные роли предназначены для агрегирования сложных привилегий базы данных, чтобы облегчить ваше повседневное управление. **Область привилегий вышеуказанных ролей не может быть изменена.**

Кроме того, если вам нужно предоставить определенные привилегии всем пользователям, Selena также предоставляет системную роль `public`.

- `public`: Эта роль принадлежит любому пользователю и активируется по умолчанию в любой сессии, включая добавление новых пользователей. Роль `public` не имеет никаких привилегий по умолчанию. Вы можете изменить область привилегий этой роли.

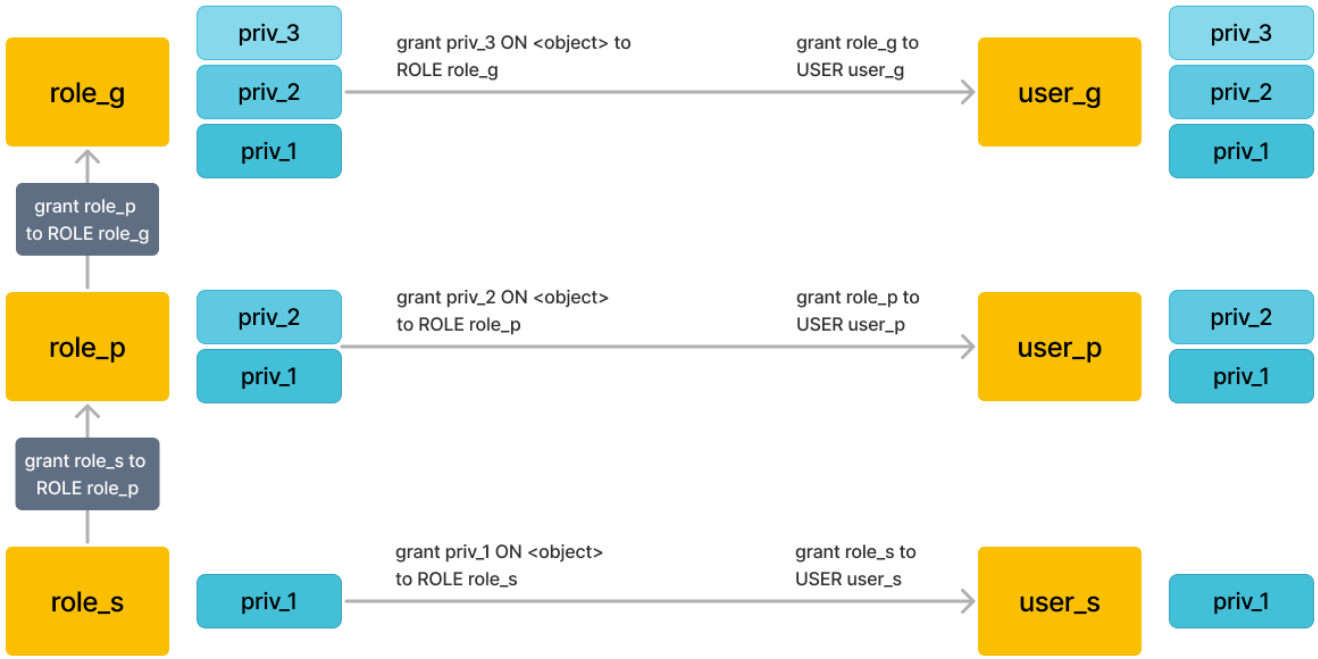
Пользовательские роли

Вы можете создавать пользовательские роли для удовлетворения конкретных бизнес-требований и изменять их область привилегий. В то же время, для удобства управления, вы можете назначать роли другим ролям для создания иерархии и наследования привилегий. Затем привилегии, связанные с ролью, наследуются другой ролью.

Иерархия ролей и наследование привилегий

На следующем рисунке показан пример наследования привилегий.

> Примечание: Максимальное количество уровней наследования для роли по умолчанию равно 16. Отношение наследования не может быть двунаправленным.



Как показано на рисунке:

- `role_s` назначена `role_p`. `role_p` неявно наследует `priv_1` от `role_s`.
- `role_p` назначена `role_g`, `role_g` неявно наследует `priv_2` от `role_p` и `priv_1` от `role_s`.
- После назначения роли пользователю, пользователь также получает привилегии этой роли.

Активные роли

Активные роли позволяют пользователям применять привилегии роли в текущей сессии. Вы можете использовать `SELECT CURRENT_ROLE();` для просмотра активных ролей в текущей сессии. Для получения дополнительной информации см. current_role.

Роли по умолчанию

Роли по умолчанию автоматически активируются при входе пользователя в кластер. Это может быть роль, принадлежащая одному или нескольким пользователям. Администратор может устанавливать роли по умолчанию с помощью ключевого слова `DEFAULT ROLE` в CREATE USER и может изменять роли по умолчанию с помощью ALTER USER.

Пользователи также могут изменять свои роли по умолчанию с помощью SET DEFAULT ROLE.

Роли по умолчанию обеспечивают базовую защиту привилегий для пользователей. Например, пользователь A имеет `role_query` и `role_delete`, которые имеют привилегии запроса и удаления соответственно. Мы рекомендуем использовать только `role_query` в качестве роли по умолчанию, чтобы предотвратить потерю данных, вызванную рискованными операциями, такими как `DELETE` или `TRUNCATE`. Если вам нужно выполнить эти операции, вы можете сделать это после ручной установки активных ролей.

Пользователь, у которого нет роли по умолчанию, по-прежнему имеет роль `public`, которая автоматически активируется после входа пользователя в кластер.

Ручная активация ролей

Помимо ролей по умолчанию, пользователи также могут вручную активировать одну или несколько существующих ролей в рамках сессии. Вы можете использовать SHOW GRANTS для просмотра привилегий и ролей, которые могут быть активированы, и

использовать SET ROLE для настройки активных ролей, которые действуют в текущей сессии.

Обратите внимание, что команда SET ROLE перезаписывает друг друга. Например, после входа пользователя в систему `default_role` активируется по умолчанию. Затем пользователь выполняет `SET ROLE role_s`. В этот момент пользователь имеет только привилегии `role_s` и свои собственные привилегии. `default_role` перезаписывается.

Ссылки

- Привилегии, поддерживаемые Selena
- Управление привилегиями пользователей

Привилегии, поддерживаемые Selena

Привилегии, предоставленные пользователю или роли, определяют, какие операции пользователь или роль может выполнять на определенных объектах. Привилегии можно использовать для реализации детального контроля доступа для защиты безопасности данных.

В этом разделе описываются привилегии, предоставляемые Selena для различных объектов, и их значения. Привилегии предоставляются и отзываются с помощью GRANT и REVOKE. Привилегии, которые могут быть предоставлены на объект, специфичны для типа объекта. Например, привилегии таблицы отличаются от привилегий базы данных.

> **IMPORTANT**
>
> Привилегии, описанные в этом разделе, доступны только начиная с v1.5.2. Структура привилегий и синтаксис в v1.5.2 не обратно совместимы с таковыми в более ранних версиях. После обновления до v1.5.2 большинство ваших исходных привилегий сохраняются, за исключением тех, которые связаны с определенными операциями. Подробные различия см. в разделе Примечания к обновлению в конце этого раздела.

Список привилегий

В этом разделе описываются привилегии, доступные для различных объектов.

SYSTEM

Привилегия	Описание
NODE	Управление узлами, например, добавление, удаление или вывод узлов из эксплуатации. Для обеспечения безопасности cluster эта привилегия не может быть напрямую предоставлена пользователям или ролям. Роль `cluster_admin` обладает этой привилегией.
GRANT	Создание пользователя или роли, изменение пользователя или роли, предоставление привилегий пользователю или роли. Эта привилегия не может быть напрямую предоставлена пользователям или ролям. Роль `user_admin` обладает этой привилегией.
CREATE RESOURCE GROUP	Создание группы ресурсов.
CREATE RESOURCE	Создание ресурсов для заданий Spark Load или внешних таблиц.
CREATE EXTERNAL CATALOG	Создание внешнего каталога.
PLUGIN	Установка или удаление плагина.
REPOSITORY	Создание, удаление или просмотр репозиториев.
BLACKLIST	Создание, удаление или отображение чёрных списков SQL и BE Blacklist.
FILE	Создание, удаление или просмотр файлов.
OPERATE	Управление replica, элементами конфигурации, переменными и транзакциями.
CREATE GLOBAL FUNCTION	Создание глобальной UDF.
CREATE STORAGE VOLUME	Создание тома хранения для удалённой системы хранения.
SECURITY	Создание или управление интеграциями безопасности и провайдерами групп.

RESOURCE GROUP

Привилегия	Описание
ALTER	Добавляет или удаляет классификаторы для группы ресурсов.
DROP	Удаляет группу ресурсов.
ALL	Имеет все вышеперечисленные привилегии на группу ресурсов.

RESOURCE

Привилегия	Описание
USAGE	Использует ресурс.
ALTER	Изменяет ресурс.
DROP	Удаляет ресурс.
ALL	Имеет все вышеперечисленные привилегии на ресурс.

USER

Привилегия	Описание
IMPERSONATE	Позволяет пользователю A выполнять операции от имени пользователя B.

GLOBAL FUNCTION (Глобальные UDF)

Привилегия	Описание
USAGE	Использует функцию в запросе.
DROP	Удаляет функцию.
ALL	Имеет все вышеперечисленные привилегии на функцию.

CATALOG

Объект	Привилегия	Описание
CATALOG (внутренний catalog)	USAGE	Использует внутренний catalog (default_catalog).
CATALOG (внутренний catalog)	CREATE DATABASE	Создает базы данных во внутреннем catalog.
CATALOG (внутренний catalog)	ALL	Имеет все вышеперечисленные привилегии на внутренний catalog.
CATALOG (внешний catalog)	USAGE	Использует внешний catalog.
CATALOG (внешний catalog)	DROP	Удаляет внешний catalog.
CATALOG (внешний catalog)	ALL	Имеет все вышеперечисленные привилегии на внешний catalog.

> Примечания: внутренний catalog Selena не может быть удален.

DATABASE

Привилегия	Описание
ALTER	Устанавливает свойства для базы данных, переименовывает базу данных или устанавливает квоты для базы данных.
DROP	Удаляет базу данных.

Привилегия	Описание
CREATE TABLE	Создает таблицы в базе данных.
CREATE VIEW	Создает представление.
CREATE FUNCTION	Создает функцию.
CREATE MATERIALIZED VIEW	Создает материализованное представление.
ALL	Имеет все вышеперечисленные привилегии на базу данных.

TABLE

Привилегия	Описание
ALTER	Изменяет таблицу или обновляет метаданные во внешней таблице.
DROP	Удаляет таблицу.
SELECT	Запрашивает данные в таблице.
INSERT	Вставляет данные в таблицу.
UPDATE	Обновляет данные в таблице.
EXPORT	Экспортирует данные из таблицы Selena.
DELETE	Удаляет данные из таблицы на основе указанного условия или удаляет все данные из таблицы.
ALL	Имеет все вышеперечисленные привилегии на таблицу.

VIEW

Привилегия	Описание
SELECT	Запрашивает данные в представлении.
ALTER	Изменяет определение представления.
DROP	Удаляет логическое представление.
ALL	Имеет все вышеперечисленные привилегии на представление.

MATERIALIZED VIEW

Привилегия	Описание
SELECT	Запрашивает материализованное представление для ускорения запросов.
ALTER	Изменяет материализованное представление.
REFRESH	Обновляет материализованное представление.
DROP	Удаляет материализованное представление.
ALL	Имеет все вышеперечисленные привилегии на материализованное представление.

FUNCTION (UDF уровня базы данных)

Привилегия	Описание
USAGE	Использует функцию.
DROP	Удаляет функцию.
ALL	Имеет все вышеперечисленные привилегии на функцию.

STORAGE VOLUME

Привилегия	Описание
ALTER	Изменяет свойства учетных данных, комментарий или статус (включено) тома хранилища.
DROP	Удаляет том хранилища.
USAGE	Описывает том хранилища и устанавливает том хранилища в качестве тома хранилища по умолчанию.
ALL	Имеет все вышеперечисленные привилегии на том хранилища.

Примечания к обновлению

Во время обновления с v2.x до v1.5.2 некоторые ваши операции могут быть невыполнимы из-за введения новой системы привилегий. В следующей таблице описаны изменения до и после обновления.

Операция	Вовлеченные команды	До	После
Изменение таблицы	ALTER TABLE, CANCEL ALTER TABLE	Пользователи, имеющие привилегию `LOAD_PRIV` на таблицу или базу данных, к которой принадлежит таблица, могут выполнять операции `ALTER TABLE` и `CANCEL ALTER TABLE`.	Вы должны иметь привилегию ALTER на таблицу, чтобы выполнять эти две операции.
Обновление внешней таблицы	REFRESH EXTERNAL TABLE	Пользователи, имеющие привилегию `LOAD_PRIV` на внешнюю таблицу, могут обновлять внешнюю таблицу.	Вы должны иметь привилегию ALTER на внешнюю таблицу, чтобы выполнить эту операцию.
Резервное копирование и восстановление	BACKUP, RESTORE	Пользователи, имеющие привилегию `LOAD_PRIV` на базу данных, могут создавать резервную копию и восстанавливать базу данных или любую таблицу в базе данных.	Администратор должен снова предоставить привилегии резервного копирования и восстановления пользователям после обновления.

Операция	Вовлеченные команды	До	После
Восстановление после удаления	RECOVER	Пользователи, имеющие привилегии `ALTER_PRIV`, `CREATE_PRIV` и `DROP_PRIV` на базу данных и таблицу, могут восстанавливать базу данных и таблицу.	Вы должны иметь привилегию CREATE DATABASE на default_catalog, чтобы восстановить базу данных. Вы должны иметь привилегию CREATE TABLE на базу данных и привилегию DROP на таблицу.
Создание и изменение пользователей	CREATE USER, ALTER USER	Пользователи, имеющие привилегию `GRANT_PRIV` на базу данных, могут создавать и изменять пользователей.	Вы должны иметь роль `user_admin`, чтобы создавать и изменять пользователей.
Предоставление и отзыв привилегий	GRANT, REVOKE	Пользователи, имеющие привилегию `GRANT_PRIV` на объект, могут предоставлять привилегии на объект другим пользователям или ролям.	После обновления вы все еще можете предоставлять привилегии, которые у вас уже есть на этот объект, другим пользователям или ролям после обновления. В новой системе привилегий: • Вы должны иметь роль `user_admin`, чтобы предоставлять привилегии другим пользователям или ролям. • Если ваше утверждение GRANT включает `WITH GRANT OPTION`, вы можете предоставлять привилегии, участвующие в утверждении, другим пользователям или ролям.

В v2.x Selena не полностью реализует управление доступом на основе ролей (RBAC). Когда вы назначаете роль пользователю, Selena напрямую предоставляет все привилегии роли пользователю, а не саму роль. Поэтому пользователь фактически не владеет ролью.

В v1.5.2 Selena обновляет свою систему привилегий. После обновления до v1.5.2 ваши исходные роли сохраняются, но по-прежнему нет владения между пользователями и ролями. Если вы хотите использовать новую систему RBAC, выполните операцию GRANT, чтобы назначить роли и привилегии.

Встроенные роли, поддерживаемые Selena

В cluster Selena есть ПЯТЬ встроенных ролей:

- `db_admin`

- `cluster_admin`
- `user_admin`
- `security_admin`
- `public`

Каждая из ролей `admin` наделена различными привилегиями для выполнения административных операций в своей специфической области. По умолчанию роль `public` не имеет привилегий и предоставляется каждому пользователю, который может получить доступ к cluster.

Подробности о привилегиях, описанных ниже, см. в разделе Элементы привилегий.

Db_admin

`db_admin` — встроенная роль администратора базы данных. Она имеет все привилегии, связанные с данными, и некоторые базовые привилегии на операции и обслуживание.

- Ориентирована на управление базами данных и данными
- Недоступна для управления пользователями или cluster
- Неизменяемая роль

Область привилегий:

Уровень привилегий	Элемент привилегий
SYSTEM	• CREATE RESOURCE • PLUGIN • FILE • BLACKLIST • OPERATE • CREATE EXTERNAL CATALOG • REPOSITORY • CREATE RESOURCE GROUP • CREATE GLOBAL FUNCTION • CREATE STORAGE VOLUME • SECURITY
CATALOG	• USAGE • DROP • ALTER • CREATE DATABASE
DATABASE	• DROP

Уровень привилегий	Элемент привилегий
	• ALTER • CREATE TABLE • CREATE VIEW • CREATE MATERIALIZED VIEW • CREATE FUNCTION • CREATE PIPE • CREATE MASKING POLICY • CREATE ROW ACCESS POLICY
TABLE	• DROP • ALTER • INSERT • UPDATE • DELETE • SELECT • EXPORT
VIEW	• DROP • ALTER • SELECT
MATERIALIZED VIEW	• DROP • ALTER • SELECT • REFRESH
RESOURCE	• USAGE • DROP • ALTER
RESOURCE GROUP	• DROP • ALTER
FUNCTION	• USAGE • DROP
GLOBAL FUNCTION	• USAGE • DROP
STORAGE VOLUME	• USAGE • DROP • ALTER
PIPE	• USAGE

Уровень привилегий	Элемент привилегий
	• DROP • ALTER

Cluster_admin

`cluster_admin` — встроенная роль администратора cluster.

- Ориентирована на управление инфраструктурой cluster
- Предоставляет привилегии на управление узлами
- Неизменяемая роль

Область привилегий:

Уровень привилегий	Элемент привилегий
SYSTEM	NODE

User_admin

`user_admin` — это встроенный администратор пользователей. Может использоваться для управления пользователями, ролями и авторизацией.

- Сфокусирована на управлении пользователями и привилегиями
- Может создавать, изменять и удалять пользователей
- Может предоставлять или отзывать привилегии или роли
- Неизменяемая роль

Область привилегий:

Уровень привилегии	Элемент привилегии
SYSTEM	GRANT

Security_admin

`security_admin` — это встроенный администратор безопасности. Может использоваться для управления интеграциями безопасности и группами провайдеров.

- Сфокусирована на управлении безопасностью системы
- Может управлять конфигурациями и стратегиями, связанными с безопасностью
- Неизменяемая роль

Область привилегий:

Уровень привилегии	Элемент привилегии
SYSTEM	• SECURITY • OPERATE

Public

`public` — это встроенная роль, которая предоставляется каждому пользователю, имеющему доступ к cluster. По умолчанию она не имеет привилегий.

- Автоматически предоставляется и активируется для всех пользователей cluster
- Изменяемая роль. Вы можете предоставить привилегии или роли этой роли, если хотите предоставить их всем пользователям cluster.

Управление привилегиями пользователей

В этой статье описывается, как управлять пользователями, ролями и привилегиями в Selena.

Selena использует как управление доступом на основе ролей (RBAC), так и управление доступом на основе идентификации (IBAC) для управления привилегиями внутри cluster'a Selena, что позволяет администраторам cluster'a легко ограничивать привилегии внутри cluster'a на различных уровнях детализации.

Внутри cluster'a Selena привилегии могут быть предоставлены пользователям или ролям. Роль — это набор привилегий, которые могут быть назначены пользователям или другим ролям в cluster'e по мере необходимости. Пользователю может быть предоставлена одна или несколько ролей, которые определяют их разрешения для различных объектов.

Просмотр информации о привилегиях и ролях

Пользователи с системной ролью `user_admin` могут просматривать всю информацию о пользователях и ролях внутри cluster'a Selena.

Просмотр информации о привилегиях

Вы можете просмотреть привилегии, предоставленные пользователю или роли, используя SHOW GRANTS.

- Просмотр привилегий текущего пользователя.

```
SHOW GRANTS;
```

> **NOTE**
>
> Любой пользователь может просматривать свои собственные привилегии без необходимости в каких-либо дополнительных привилегиях.

- Просмотр привилегий конкретного пользователя.

  Следующий пример показывает привилегии пользователя `jack`:

  ```
  SHOW GRANTS FOR jack@'172.10.1.10';
  ```

- Просмотр привилегий конкретной роли.

  Следующий пример показывает привилегии роли `example_role`:

  ```
  SHOW GRANTS FOR ROLE example_role;
  ```

Просмотр ролей

Вы можете просмотреть все роли внутри cluster'a Selena, используя SHOW ROLES.

```
SHOW ROLES;
```

Управление ролями

Пользователи с системной ролью `user_admin` могут создавать, предоставлять, отзывать или удалять роли в Selena.

Создание роли

Вы можете создать роль, используя CREATE ROLE. По умолчанию пользователь может иметь максимум 64 роли. Вы можете настроить этот параметр, используя динамический параметр FE `privilege_max_total_roles_per_user`. Роль может иметь максимум 16 уровней наследования. Вы можете настроить этот параметр, используя динамический параметр FE `privilege_max_role_depth`.

Следующий пример создаёт роль `example_role`:

```
CREATE ROLE example_role;
```

Предоставление роли

Вы можете предоставить роли пользователю или другой роли, используя GRANT.

- Предоставление роли пользователю.

  Следующий пример предоставляет роль `example_role` пользователю `jack`:

  ```
  GRANT example_role TO USER jack@'172.10.1.10';
  ```

- Предоставление роли другой роли.

  Следующий пример предоставляет роль `example_role` роли `test_role`:

  ```
  GRANT example_role TO ROLE test_role;
  ```

- Предоставление роли группе пользователей. Вы можете идентифицировать группы пользователей из внешних систем аутентификации через Group Provider.

  Следующий пример предоставляет роль `example_role` группе пользователей `analysts`:

  ```
  GRANT example_role TO EXTERNAL GROUP analysts;
  ```

Изменение роли по умолчанию для пользователя

Роль по умолчанию пользователя автоматически активируется при подключении пользователя к Selena. Инструкции о том, как включить все роли (по умолчанию и предоставленные) для пользователя после подключения, см. в разделе Включение всех ролей.

Вы можете установить роль по умолчанию для пользователя, используя SET DEFAULT ROLE или ALTER USER.

Оба следующих примера устанавливают роль по умолчанию для `jack` как `db1_admin`. Обратите внимание, что `db1_admin` должна быть назначена `jack`.

- Установка роли по умолчанию с помощью SET DEFAULT ROLE:

```
SET DEFAULT ROLE 'db1_admin' TO jack@'172.10.1.10';
```

- Установка роли по умолчанию с помощью ALTER USER:

```
ALTER USER jack@'172.10.1.10' DEFAULT ROLE 'db1_admin';
```

Отзыв роли

Вы можете отозвать роли у пользователя или другой роли, используя REVOKE.

> **NOTE**
>
> Вы не можете отозвать системную роль по умолчанию `PUBLIC` у пользователя.

- Отзыв роли у пользователя.

  Следующий пример отзывает роль `example_role` у пользователя `jack`:

```
REVOKE example_role FROM USER jack@'172.10.1.10';
```

- Отзыв роли у другой роли.

  Следующий пример отзывает роль `example_role` у роли `test_role`:

```
REVOKE example_role FROM ROLE test_role;
```

Удаление роли

Вы можете удалить роль, используя DROP ROLE.

Следующий пример удаляет роль `example_role`:

```
DROP ROLE example_role;
```

> **CAUTION**
>
> Системные роли не могут быть удалены.

Включение всех ролей

Роли по умолчанию пользователя — это роли, которые автоматически активируются каждый раз, когда пользователь подключается к cluster'у Selena.

Если вы хотите включить все роли (роли по умолчанию и предоставленные роли) для всех пользователей Selena при подключении к cluster'у Selena, вы можете выполнить следующую операцию.

Эта операция требует системной привилегии OPERATE.

```
SET GLOBAL activate_all_roles_on_login = TRUE;
```

Вы также можете использовать SET ROLE для активации ролей, назначенных вам. Например, пользователь `jack@'172.10.1.10'` имеет роли `db_admin` и `user_admin`, но они не являются ролями по умолчанию пользователя и не активируются автоматически при подключении пользователя к Selena. Если `jack@'172.10.1.10'` необходимо активировать `db_admin` и `user_admin`, он может выполнить `SET ROLE db_admin, user_admin;`. Обратите внимание, что SET ROLE перезаписывает исходные роли. Если вы хотите включить все свои роли, выполните SET ROLE ALL.

Управление привилегиями

Пользователи с системной ролью `user_admin` могут предоставлять или отзывать привилегии в Selena.

Предоставление привилегий

Вы можете предоставить привилегии пользователю или роли, используя GRANT.

- Предоставление привилегии пользователю.

Следующий пример предоставляет привилегию SELECT на таблицу `sr_member` пользователю `jack` и позволяет `jack` предоставлять эту привилегию другим пользователям или ролям (указав WITH GRANT OPTION в SQL):

```
GRANT SELECT ON TABLE sr_member TO USER jack@'172.10.1.10' WITH GRANT
OPTION;
```

- Предоставление привилегии роли.

  Следующий пример предоставляет привилегию SELECT на таблицу `sr_member` роли `example_role`:

  ```
  GRANT SELECT ON TABLE sr_member TO ROLE example_role;
  ```

Отзыв привилегий

Вы можете отозвать привилегии у пользователя или роли, используя REVOKE.

- Отзыв привилегии у пользователя.

  Следующий пример отзывает привилегию SELECT на таблицу `sr_member` у пользователя `jack` и запрещает `jack` предоставлять эту привилегию другим пользователям или ролям:

  ```
  REVOKE SELECT ON TABLE sr_member FROM USER jack@'172.10.1.10';
  ```

- Отзыв привилегии у роли.

  Следующий пример отзывает привилегию SELECT на таблицу `sr_member` у роли `example_role`:

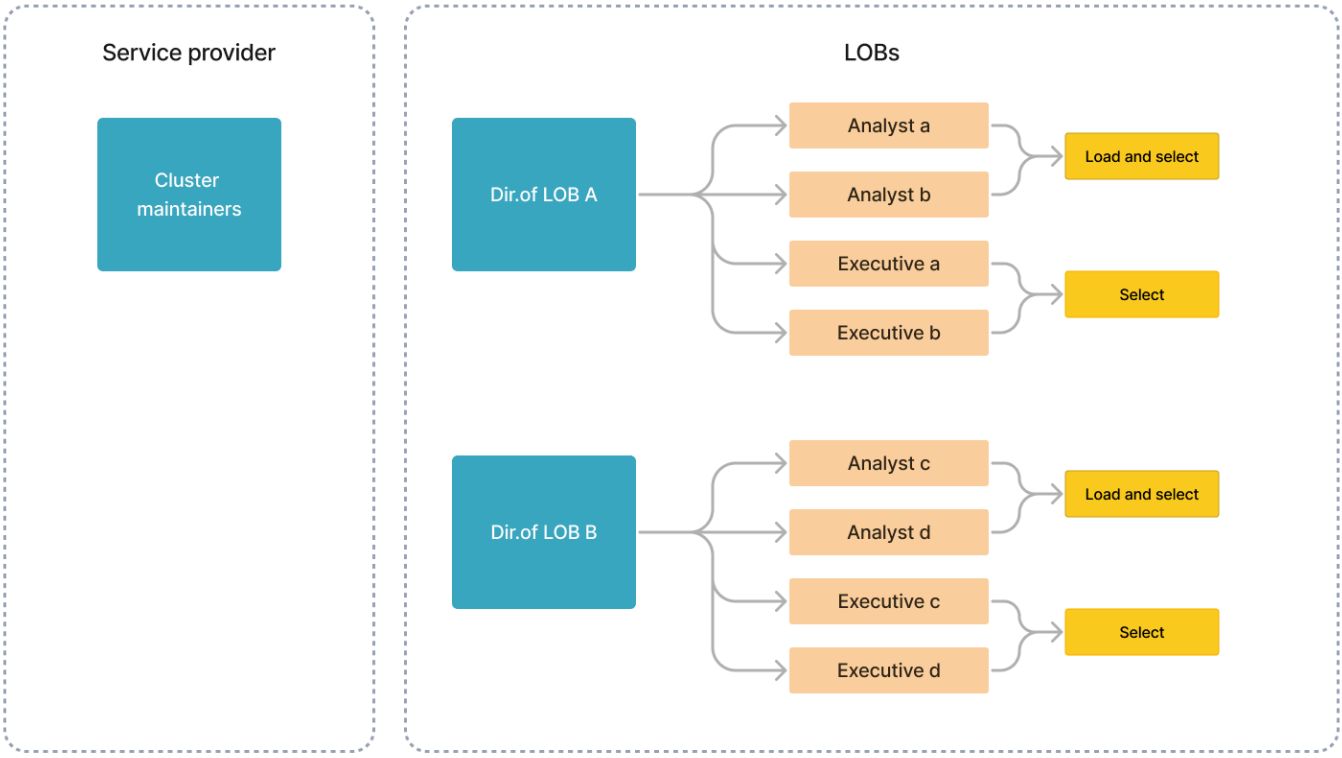
  ```
  REVOKE SELECT ON TABLE sr_member FROM ROLE example_role;
  ```

Лучшие практики

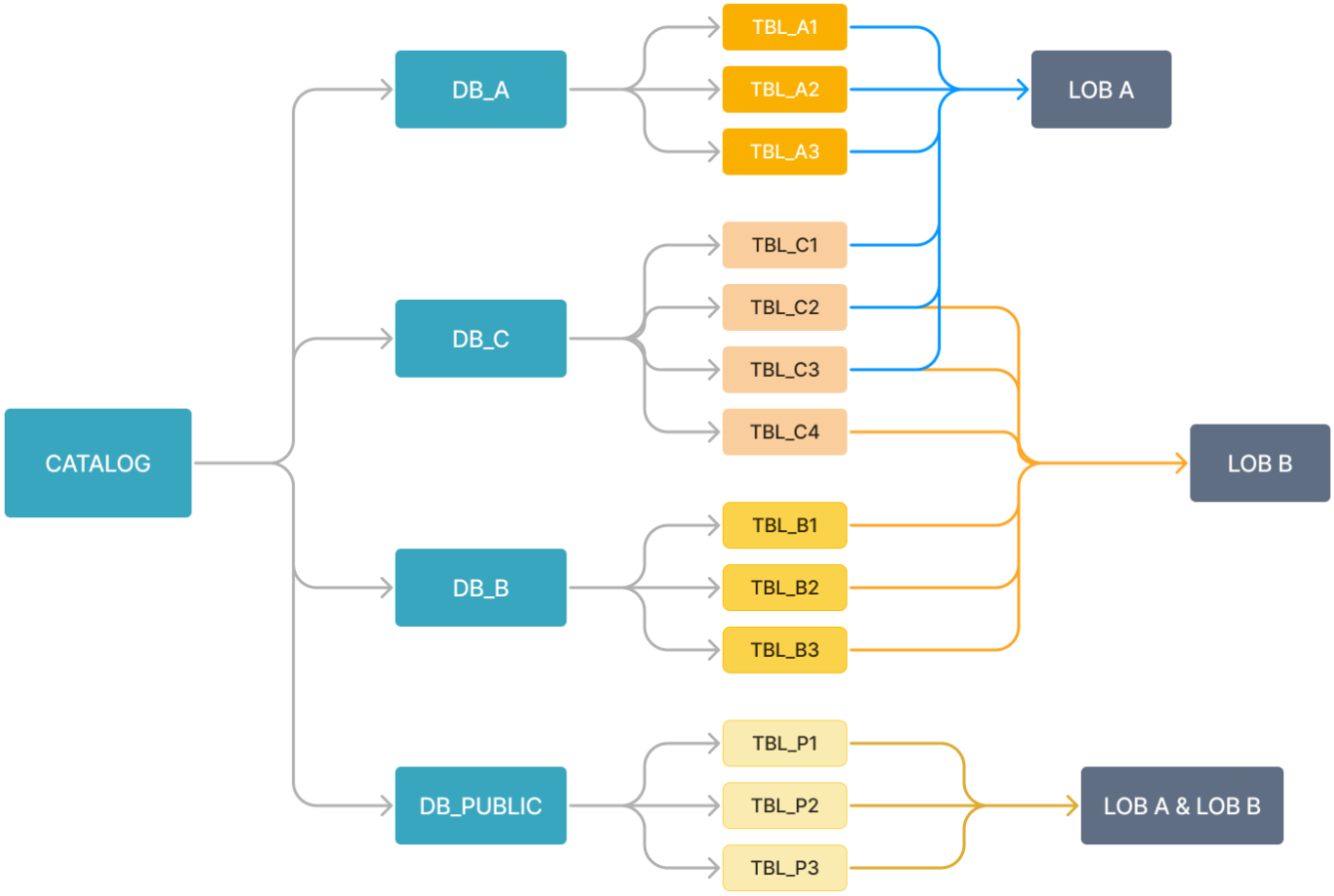
Управление доступом для нескольких сервисов

Обычно cluster Selena, принадлежащий компании, управляется единственным поставщиком услуг и поддерживает несколько направлений бизнеса (LOB), каждое из которых использует одну или несколько баз данных.

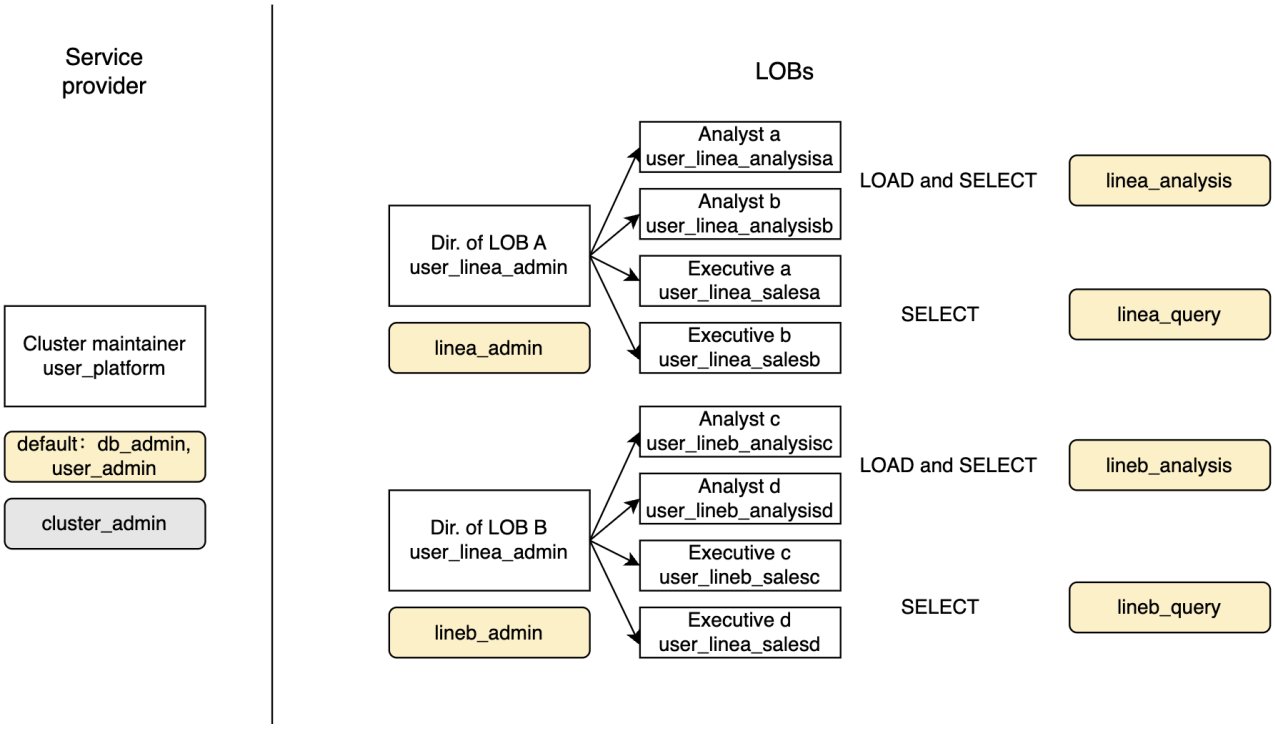
Как показано ниже, пользователи cluster'a Selena включают членов от поставщика услуг и двух направлений бизнеса (A и B). Каждое направление бизнеса управляется двумя ролями — аналитиками и руководителями. Аналитики генерируют и анализируют бизнес-отчёты, а руководители запрашивают эти отчёты.



Направление бизнеса A независимо управляет базой данных `DB_A`, а направление бизнеса B — базой данных `DB_B`. Направления бизнеса A и B используют разные таблицы в `DB_C`. `DB_PUBLIC` доступна всем членам обоих направлений бизнеса.



Поскольку разные члены выполняют разные операции с разными базами данных и таблицами, мы рекомендуем создавать роли в соответствии с их сервисами и должностями, применять только необходимые привилегии к каждой роли и назначать эти роли соответствующим членам. Как показано ниже:



1. Назначьте системные роли `db_admin`, `user_admin` и `cluster_admin` сопровождающим cluster'a, установите `db_admin` и `user_admin` в качестве их ролей по умолчанию для ежедневного обслуживания и вручную активируйте роль `cluster_admin`, когда им нужно выполнять операции с узлами cluster'a.

   Пример:

   ```
   GRANT db_admin, user_admin, cluster_admin TO USER user_platform;
   ALTER USER user_platform DEFAULT ROLE db_admin, user_admin;
   ```

2. Создайте пользователей для каждого члена внутри направлений бизнеса и установите сложные пароли для каждого пользователя.

3. Создайте роли для каждой должности внутри направлений бизнеса и примените соответствующие привилегии к каждой роли.

   Для директора каждого направления бизнеса предоставьте их роли максимальный набор привилегий, необходимых их направлению бизнеса, и соответствующие привилегии GRANT (указав WITH GRANT OPTION в выражении). Таким образом, они смогут назначать эти привилегии членам своего направления бизнеса. Установите роль в качестве их роли по умолчанию, если их ежедневная работа требует этого.

Пример:

```
GRANT SELECT, ALTER, INSERT, UPDATE, DELETE ON ALL TABLES IN DATABASE
DB_A TO ROLE linea_admin WITH GRANT OPTION;
GRANT SELECT, ALTER, INSERT, UPDATE, DELETE ON TABLE TABLE_C1,
TABLE_C2, TABLE_C3 TO ROLE linea_admin WITH GRANT OPTION;
GRANT linea_admin TO USER user_linea_admin;
ALTER USER user_linea_admin DEFAULT ROLE linea_admin;
```

Для аналитиков и руководителей назначьте им роль с соответствующими привилегиями.

Пример:

```
GRANT SELECT ON ALL TABLES IN DATABASE DB_A TO ROLE linea_query;
GRANT SELECT ON TABLE TABLE_C1, TABLE_C2, TABLE_C3 TO ROLE linea_query;
GRANT linea_query TO USER user_linea_salesa;
GRANT linea_query TO USER user_linea_salesb;
ALTER USER user_linea_salesa DEFAULT ROLE linea_query;
ALTER USER user_linea_salesb DEFAULT ROLE linea_query;
```

4. Для базы данных `DB_PUBLIC`, к которой могут получить доступ все пользователи cluster'a, предоставьте привилегию SELECT на `DB_PUBLIC` системной роли `public`.

   Пример:

```
GRANT SELECT ON ALL TABLES IN DATABASE DB_PUBLIC TO ROLE public;
```

Вы можете назначать роли другим ролям для достижения наследования ролей в сложных сценариях.

Например, если аналитикам требуются привилегии на запись и запрос таблиц в `DB_PUBLIC`, а руководители могут только запрашивать эти таблицы, вы можете создать роли `public_analysis` и `public_sales`, применить соответствующие привилегии к ролям и назначить их исходным ролям аналитиков и руководителей соответственно.

Пример:

```
CREATE ROLE public_analysis;
CREATE ROLE public_sales;
GRANT SELECT, ALTER, INSERT, UPDATE, DELETE ON ALL TABLES IN DATABASE
DB_PUBLIC TO ROLE public_analysis;
GRANT SELECT ON ALL TABLES IN DATABASE DB_PUBLIC TO ROLE public_sales;
GRANT public_analysis TO ROLE linea_analysis;
GRANT public_analysis TO ROLE lineb_analysis;
GRANT public_sales TO ROLE linea_query;
GRANT public_sales TO ROLE lineb_query;
```

Настройка ролей на основе сценариев

Мы рекомендуем создавать пользовательские роли для управления привилегиями и пользователями. Следующие примеры классифицируют несколько комбинаций привилегий для распространённых сценариев.

Предоставление глобальных привилегий только на чтение для таблиц Selena

```
-- Создать роль.
CREATE ROLE read_only;
-- Предоставить привилегию USAGE на все каталоги роли.
GRANT USAGE ON ALL CATALOGS TO ROLE read_only;
-- Предоставить привилегию на запрос всех таблиц роли.
GRANT SELECT ON ALL TABLES IN ALL DATABASES TO ROLE read_only;
-- Предоставить привилегию на запрос всех представлений роли.
GRANT SELECT ON ALL VIEWS IN ALL DATABASES TO ROLE read_only;
-- Предоставить привилегию на запрос всех материализованных представлений и
привилегию на ускорение запросов с их помощью роли.
GRANT SELECT ON ALL MATERIALIZED VIEWS IN ALL DATABASES TO ROLE read_only;
```

Вы также можете предоставить привилегию на использование UDF в запросах:

```
-- Предоставить привилегию USAGE на все UDF уровня базы данных роли.
GRANT USAGE ON ALL FUNCTIONS IN ALL DATABASES TO ROLE read_only;
-- Предоставить привилегию USAGE на глобальные UDF роли.
GRANT USAGE ON ALL GLOBAL FUNCTIONS TO ROLE read_only;
```

Предоставление глобальных привилегий только на запись для таблиц Selena

```
-- Создать роль.
CREATE ROLE write_only;
-- Предоставить привилегию USAGE на все каталоги роли.
GRANT USAGE ON ALL CATALOGS TO ROLE write_only;
-- Предоставить привилегии INSERT и UPDATE на все таблицы роли.
GRANT INSERT, UPDATE ON ALL TABLES IN ALL DATABASES TO ROLE write_only;
-- Предоставить привилегию REFRESH на все материализованные представления
роли.
GRANT REFRESH ON ALL MATERIALIZED VIEWS IN ALL DATABASES TO ROLE
write_only;
```

Предоставление привилегий только на чтение для определённого внешнего каталога

```
-- Создать роль.
CREATE ROLE read_catalog_only;
-- Предоставить привилегию USAGE на целевой каталог роли.
GRANT USAGE ON CATALOG hive_catalog TO ROLE read_catalog_only;
-- Переключиться на соответствующий каталог.
SET CATALOG hive_catalog;
-- Предоставить привилегии на запрос всех таблиц и всех представлений во
внешнем каталоге.
GRANT SELECT ON ALL TABLES IN ALL DATABASES TO ROLE read_catalog_only;
```

ПОДСКАЗКА

Для представлений во внешних каталогах можно запрашивать только представления таблиц Hive (начиная с v1.5.2).

Предоставление привилегий только на запись для определённого внешнего каталога

Вы можете записывать данные только в таблицы Iceberg (начиная с v1.5.2) и таблицы Hive (начиная с v1.5.2).

```
-- Создать роль.
CREATE ROLE write_catalog_only;
-- Предоставить привилегию USAGE на целевой каталог роли.
GRANT USAGE ON CATALOG iceberg_catalog TO ROLE read_catalog_only;
-- Переключиться на соответствующий каталог.
SET CATALOG iceberg_catalog;
-- Предоставить привилегию на запись данных в таблицы Iceberg.
GRANT INSERT ON ALL TABLES IN ALL DATABASES TO ROLE write_catalog_only;
```

Предоставление административных привилегий для определённой базы данных

```
-- Создать роль.
CREATE ROLE db1_admin;
-- Предоставить ВСЕ привилегии на целевую базу данных роли. Эта роль может
создавать таблицы, представления, материализованные представления и UDF в
этой базе данных. Она также может удалять или изменять эту базу данных.
GRANT ALL ON DATABASE db1 TO ROLE db1_admin;
-- Переключиться на соответствующий каталог.
SET CATALOG iceberg_catalog;
-- Предоставить все привилегии на таблицы, представления, материализованные
представления и UDF в этой базе данных роли.
GRANT ALL ON ALL TABLES IN DATABASE db1 TO ROLE db1_admin;
GRANT ALL ON ALL VIEWS IN DATABASE db1 TO ROLE db1_admin;
GRANT ALL ON ALL MATERIALIZED VIEWS IN DATABASE db1 TO ROLE db1_admin;
GRANT ALL ON ALL FUNCTIONS IN DATABASE db1 TO ROLE db1_admin;
```

Предоставление привилегий для выполнения операций резервного копирования и восстановления на глобальном уровне, уровне базы данных, таблицы и partition

- Предоставление привилегий для выполнения глобальных операций резервного копирования и восстановления:

  Привилегии для выполнения глобальных операций резервного копирования и восстановления позволяют роли создавать резервные копии и восстанавливать любую базу данных, таблицу или partition. Требуется привилегия REPOSITORY на уровне SYSTEM, привилегии на создание баз данных в каталоге по умолчанию, на создание таблиц в любой базе данных, а также на загрузку и экспорт данных в любой таблице.

```
-- Создать роль.
CREATE ROLE recover;
-- Предоставить привилегию REPOSITORY на уровне SYSTEM.
GRANT REPOSITORY ON SYSTEM TO ROLE recover;
-- Предоставить привилегию на создание баз данных в каталоге по
умолчанию.
GRANT CREATE DATABASE ON CATALOG default_catalog TO ROLE recover;
-- Предоставить привилегию на создание таблиц в любой базе данных.
GRANT CREATE TABLE ON ALL DATABASES TO ROLE recover;
-- Предоставить привилегию на загрузку и экспорт данных в любой
таблице.
GRANT INSERT, EXPORT ON ALL TABLES IN ALL DATABASES TO ROLE recover;
```

- Предоставление привилегий для выполнения операций резервного копирования и восстановления на уровне базы данных:

  Привилегии для выполнения операций резервного копирования и восстановления на уровне базы данных требуют привилегии REPOSITORY на уровне SYSTEM, привилегии на создание баз данных в каталоге по умолчанию, привилегии на создание таблиц в любой базе данных, привилегии на загрузку данных в любую таблицу и привилегии на экспорт данных из любой таблицы в базе данных для резервного копирования.

```
-- Создать роль.
CREATE ROLE recover_db;
-- Предоставить привилегию REPOSITORY на уровне SYSTEM.
GRANT REPOSITORY ON SYSTEM TO ROLE recover_db;
-- Предоставить привилегию на создание баз данных.
GRANT CREATE DATABASE ON CATALOG default_catalog TO ROLE recover_db;
-- Предоставить привилегию на создание таблиц.
GRANT CREATE TABLE ON ALL DATABASES TO ROLE recover_db;
-- Предоставить привилегию на загрузку данных в любую таблицу.
GRANT INSERT ON ALL TABLES IN ALL DATABASES TO ROLE recover_db;
-- Предоставить привилегию на экспорт данных из любой таблицы в базе
данных для резервного копирования.
GRANT EXPORT ON ALL TABLES IN DATABASE <db_name> TO ROLE recover_db;
```

- Предоставление привилегий для выполнения операций резервного копирования и восстановления на уровне таблицы:

Привилегии для выполнения операций резервного копирования и восстановления на уровне таблицы требуют привилегии REPOSITORY на уровне SYSTEM, привилегии на создание таблиц в соответствующих базах данных, привилегии на загрузку данных в любую таблицу в базе данных и привилегии на экспорт данных из таблицы для резервного копирования.

```
-- Создать роль.
CREATE ROLE recover_tbl;
-- Предоставить привилегию REPOSITORY на уровне SYSTEM.
GRANT REPOSITORY ON SYSTEM TO ROLE recover_tbl;
-- Предоставить привилегию на создание таблиц в соответствующих базах
данных.
GRANT CREATE TABLE ON DATABASE <db_name> TO ROLE recover_tbl;
-- Предоставить привилегию на загрузку данных в любую таблицу в базе
данных.
GRANT INSERT ON ALL TABLES IN DATABASE <db_name> TO ROLE recover_db;
-- Предоставить привилегию на экспорт данных из таблицы, которую вы
хотите скопировать.
GRANT EXPORT ON TABLE <table_name> TO ROLE recover_tbl;
```

- Предоставление привилегий для выполнения операций резервного копирования и восстановления на уровне partition:

  Привилегии для выполнения операций резервного копирования и восстановления на уровне partition требуют привилегии REPOSITORY на уровне SYSTEM и привилегии на загрузку и экспорт данных в соответствующей таблице.

```
-- Создать роль.
CREATE ROLE recover_par;
-- Предоставить привилегию REPOSITORY на уровне SYSTEM.
GRANT REPOSITORY ON SYSTEM TO ROLE recover_par;
-- Предоставить привилегию на загрузку и экспорт данных в
соответствующей таблице.
GRANT INSERT, EXPORT ON TABLE <table_name> TO ROLE recover_par;
```

Управление правами доступа с помощью Apache Ranger

Apache Ranger предоставляет централизованную платформу управления безопасностью, которая позволяет пользователям настраивать политики доступа через визуальный веб-интерфейс. Это помогает определить, какие роли могут получить доступ к каким данным и

обеспечивает детализированный контроль доступа к данным для различных компонентов и сервисов в экосистеме Hadoop.

Apache Ranger предоставляет следующие основные модули:

- **Ranger Admin**: основной модуль Ranger со встроенным веб-интерфейсом. Пользователи могут создавать и обновлять политики безопасности на этой странице или через REST-интерфейс. Плагины различных компонентов экосистемы Hadoop периодически опрашивают и получают эти политики.
- **Agent Plugin**: плагины компонентов, встроенные в экосистему Hadoop. Эти плагины регулярно получают политики безопасности от Ranger Admin и сохраняют их в локальных файлах. Когда пользователи обращаются к компоненту, соответствующий плагин оценивает запрос на основе настроенной политики безопасности и отправляет результаты аутентификации соответствующему компоненту.
- **User Sync**: используется для получения информации о пользователях и группах пользователей, а также для синхронизации данных о правах пользователей и групп пользователей с базой данных Ranger.

Помимо встроенной системы привилегий RBAC, Selena v1.5.2 также поддерживает контроль доступа через Apache Ranger. В настоящее время Selena поддерживает:

- Создание политик доступа, политик маскирования и политик фильтрации на уровне строк через Apache Ranger.
- Журналы аудита Ranger.
- **Серверы Ranger, использующие Kerberos для аутентификации, не поддерживаются.**
- Вы можете зарегистрировать несколько сервисов Selena в одном сервисе Apache Ranger для управления привилегиями в разных кластерах Selena.

Эта тема описывает методы контроля доступа и процесс интеграции Selena и Apache Ranger. Для получения информации о том, как создавать политики безопасности в Ranger для управления безопасностью данных, см. официальный сайт Apache Ranger.

Начиная с версии v1.5.2, Selena поддерживает Group Provider для сбора информации о группах из внешних систем аутентификации для управления группами пользователей. Для получения дополнительной информации см. Аутентификация групп пользователей.

Метод контроля доступа

Selena, интегрированная с Apache Ranger, предоставляет следующие методы контроля доступа:

- Создание сервиса Selena в Ranger для реализации контроля доступа. Когда пользователи обращаются к внутренним таблицам Selena, внешним таблицам или другим объектам, контроль доступа выполняется в соответствии с политиками доступа, настроенными в сервисе Selena.
- Когда пользователи обращаются к внешнему источнику данных, внешний сервис (например, Hive Service) в Apache Ranger может быть повторно использован для контроля доступа. Selena может сопоставлять сервисы Ranger с различными External Catalogs и реализовывать контроль доступа на основе сервиса Ranger, соответствующего источнику данных.

После интеграции Selena с Apache Ranger вы можете реализовать следующие шаблоны контроля доступа:

- Использовать Apache Ranger для единого управления доступом к внутренним таблицам Selena, внешним таблицам и всем объектам.
- Использовать Apache Ranger для управления доступом к внутренним таблицам и объектам Selena. Для External Catalogs повторно использовать политику соответствующего внешнего сервиса в Ranger для контроля доступа.
- Использовать Apache Ranger для управления доступом к External Catalogs путем повторного использования сервиса, соответствующего внешнему источнику данных. Использовать встроенную систему привилегий RBAC Selena для управления доступом к внутренним таблицам и объектам Selena.

Процесс аутентификации

1. Вы также можете использовать LDAP для аутентификации пользователей, затем использовать Ranger для синхронизации пользователей LDAP и настройки правил доступа для них. Selena также может выполнить аутентификацию входа пользователя через LDAP.
2. Когда пользователи инициируют запрос, Selena анализирует оператор запроса, передает информацию о пользователе и необходимые привилегии в Apache Ranger. Ranger определяет, имеет ли пользователь необходимую привилегию на основе политики доступа, настроенной в соответствующем сервисе, и возвращает результат

аутентификации в Selena. Если у пользователя есть доступ, Selena возвращает данные запроса; если нет, Selena возвращает ошибку.

Предварительные требования

- Установлен Apache Ranger версии 2.1.0 или более поздней. Инструкции по установке Apache Ranger см. в руководстве по быстрому запуску Ranger.

- Все машины Selena FE имеют доступ к Apache Ranger. Вы можете проверить это, выполнив следующую команду на каждой машине FE:

  ```
  telnet <ranger-ip> <ranger-port>
  ```

  Если отображается `Connected to <ip>`, соединение установлено успешно.

Интеграция сервиса Selena с Ranger

(Опционально) Установка ranger-selena-plugin

> **ПРИМЕЧАНИЕ**
>
> Основная цель этого шага - использовать функцию автозаполнения имен ресурсов Ranger. При создании политик в Ranger Admin пользователям необходимо вводить имя ресурсов, доступ к которым требуется защитить. Чтобы упростить ввод имен ресурсов для пользователей, Ranger Admin предоставляет функцию автозаполнения, которая ищет доступные ресурсы в сервисе, соответствующие введенным данным, и автоматически дополняет имя ресурса.
>
> Если у вас нет разрешений на работу с кластером Ranger или вам не нужна эта функция, вы можете пропустить этот шаг.
>
> Также обратите внимание, что если вы не установили ranger-selena-plugin, то вы не сможете использовать `test connection` при создании сервиса Selena. Однако это не означает, что вы не можете успешно создать сервис.

1. Создайте папку `selena` в директории Ranger Admin `ews/webapp/WEB-INF/classes/ranger-plugins`.

```
mkdir {path-to-ranger}/ews/webapp/WEB-INF/classes/ranger-plugins/selena
```

2. Загрузите `ranger-selena-plugin-3.0.0-SNAPSHOT.jar` (обратитесь в техническую поддержку для получения дистрибутива) и mysql-connector-j.jar и поместите их в папку `selena`.

3. Перезапустите Ranger Admin.

   - Ranger 0.5.x:

     ```
     ranger-admin restart
     ```

   - Ranger 2.x и выше:

     ```
     ./ews/ranger-admin-services.sh restart
     ```

Настройка сервиса Selena в Ranger Admin

> **ПРИМЕЧАНИЕ**
>
> На этом шаге настраивается сервис Selena в Ranger, чтобы пользователи могли выполнять контроль доступа к объектам Selena через Ranger.

1. Скопируйте ranger-servicedef-selena.json в любую директорию машины Selena FE или машины Ranger.

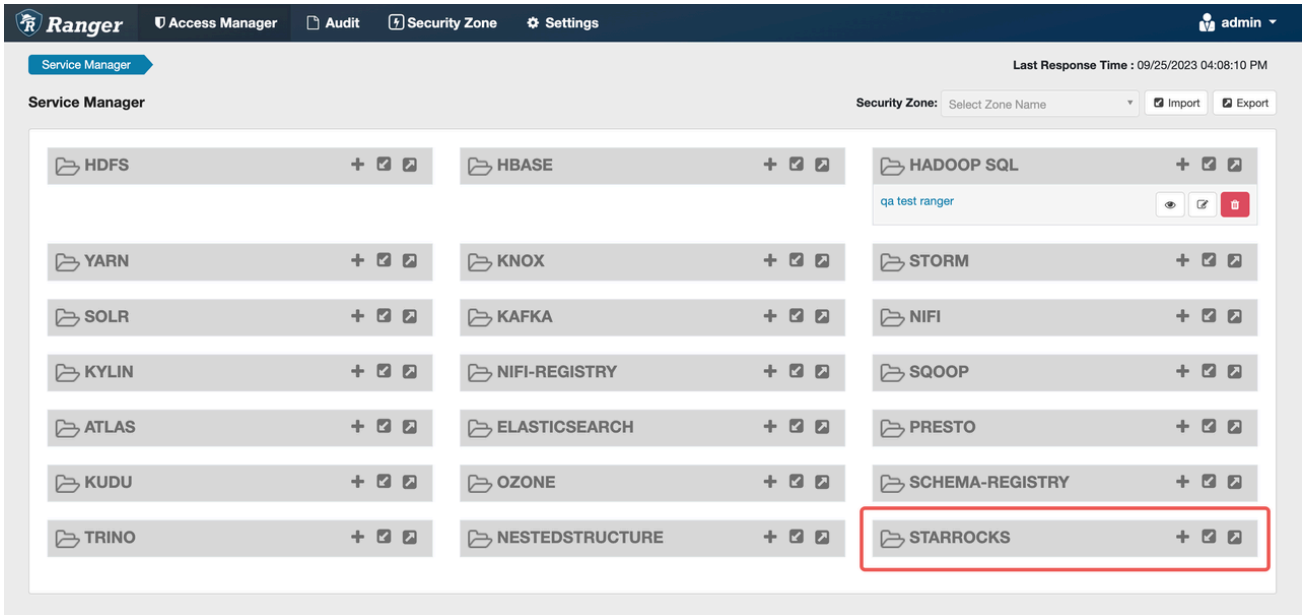
   ```
   wget
   https://raw.githubusercontent.com/Selena/selena/main/conf/ranger/ranger-servicedef-selena.json
   ```

> **ПРИМЕЧАНИЕ**
>
> Если вам не нужна функция автозаполнения Ranger (что означает, что вы не устанавливали ranger-selena-plugin), вы должны установить `implClass` в файле .json в пустое значение:
>
> ```
> "implClass": "",
> ```
>
> Если вам нужна функция автозаполнения Ranger (что означает, что вы установили ranger-selena-plugin), вы должны установить `implClass` в файле .json в значение `org.apache.ranger.services.selena.RangerServiceSelena`:
>
> ```
> "implClass":
> "org.apache.ranger.services.selena.RangerServiceSelena",
> ```

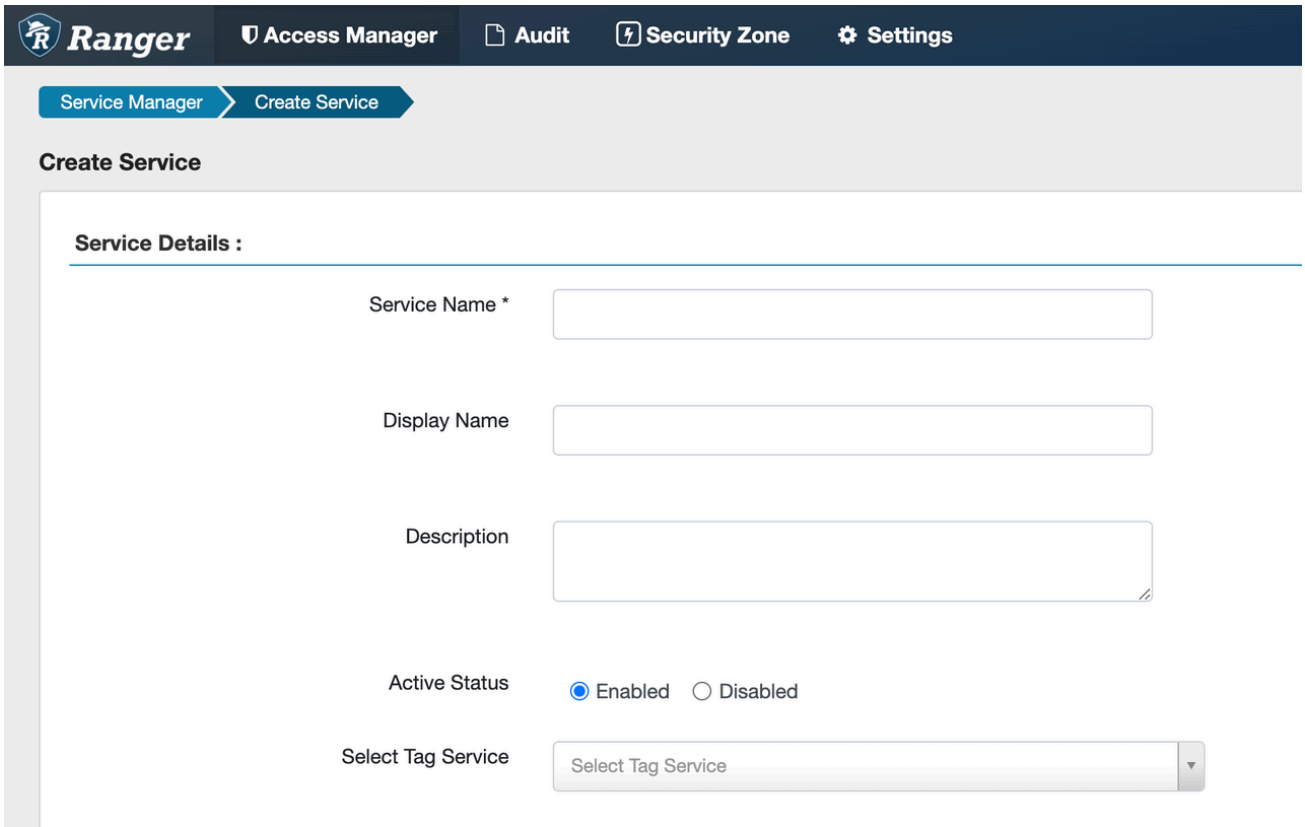
2. Добавьте сервис Selena, выполнив следующую команду от имени администратора Ranger.

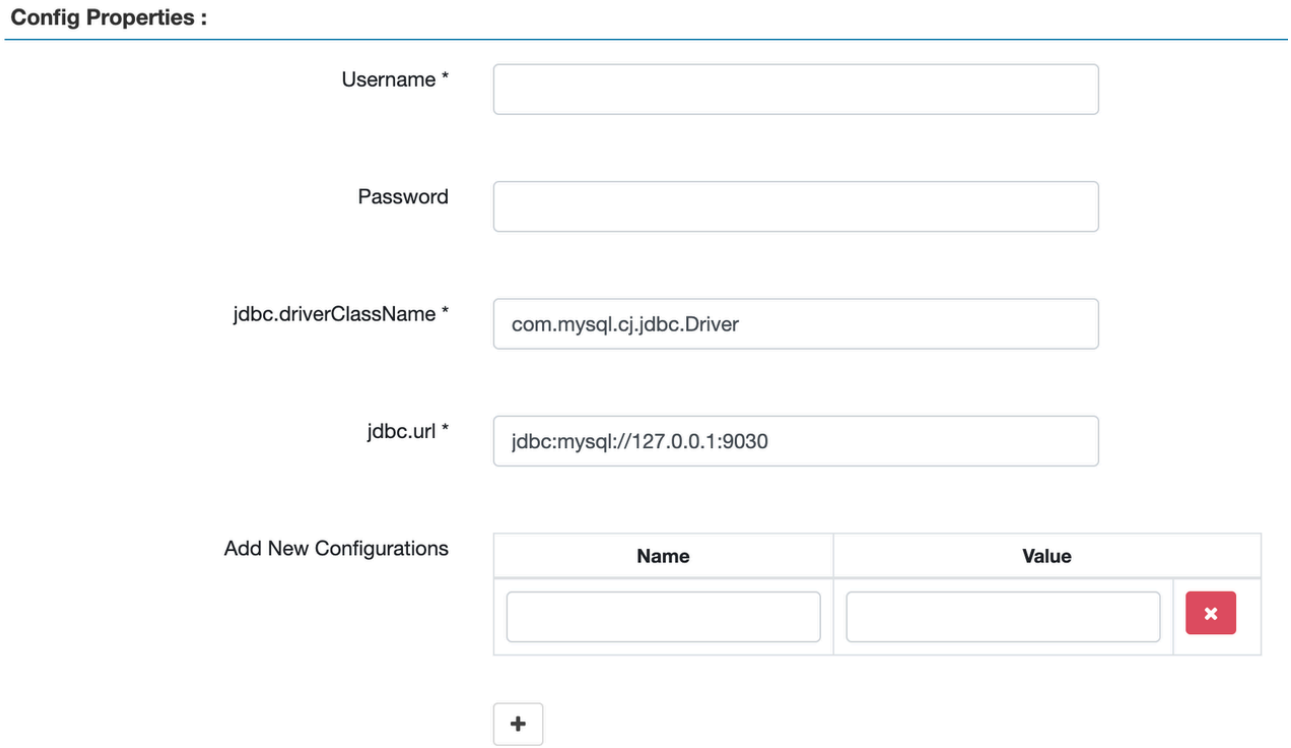
```
curl -u <ranger_adminuser>:<ranger_adminpwd> \
-X POST -H "Accept: application/json" \
-H "Content-Type: application/json" http://<ranger-ip>:<ranger-
port>/service/plugins/definitions -d@ranger-servicedef-selena.json
```

3. Откройте `http://<ranger-ip>:<ranger-host>/login.jsp` для входа на страницу Apache Ranger. Сервис SELENA появится на странице.



4. Нажмите знак плюс (+) после **SELENA**, чтобы настроить сервис Selena.





- `Service Name`: Вы должны ввести имя сервиса.
- `Display Name`: Имя, которое вы хотите отобразить для сервиса под SELENA. Если не указано, будет отображаться `Service Name`.
- `Username` и `Password`: Имя пользователя и пароль FE, используемые для автозаполнения имен объектов при создании политик. Эти два параметра не влияют на подключение между Selena и Ranger. Если вы хотите использовать автозаполнение, настройте хотя бы одного пользователя с активированной ролью `db_admin`.
- `jdbc.url`: Введите IP-адрес и порт Selena FE.

На следующем рисунке показан пример конфигурации.

Service Details

Service Details :

Service Name	starrocks-test
Display Name	starrocks-test
Description	--
Active Status	Enabled
Tag Service	--

Config Properties :

Username	root
Password	--
jdbc.driverClassName	com.mysql.cj.jdbc.Driver
jdbc.url	jdbc:mysql://172.26.80.4:9030
Add New Configurations :	
--	--

На следующем рисунке показан добавленный сервис.

5. Нажмите **Test connection**, чтобы проверить подключение, и сохраните его после успешного подключения. Если вы не устанавливали ranger-selena-plugin, вы можете пропустить проверку подключения и создать сервис напрямую.

6. На каждой машине FE кластера Selena создайте файл ranger-selena-security.xml в папке `fe/conf` и скопируйте содержимое. Вы должны изменить следующие два параметра и сохранить изменения:

   - `ranger.plugin.selena.service.name`: Измените на имя сервиса Selena, которое вы создали на шаге 4.

- `ranger.plugin.selena.policy.rest.url`: Измените на адрес Ranger Admin.

Если вам нужно изменить другие конфигурации, обратитесь к официальной документации Apache Ranger. Например, вы можете изменить `ranger.plugin.selena.policy.pollIntervalMs`, чтобы изменить интервал получения изменений политики.

```
vim ranger-selena-security.xml

...
    <property>
        <name>ranger.plugin.selena.service.name</name>
        <value>selena</value> -- Измените на имя сервиса Selena.
        <description>
            Name of the Ranger service containing policies for this
Selena instance
        </description>
    </property>
...

...
    <property>
        <name>ranger.plugin.selena.policy.rest.url</name>
        <value>http://localhost:6080</value> -- Измените на адрес
Ranger Admin.
        <description>
            URL to Ranger Admin
        </description>
    </property>
...
```

7. (Опционально) Если вы хотите использовать сервис журнала аудита Ranger, вам нужно создать файл ranger-selena-audit.xml в папке `fe/conf` каждой машины FE. Скопируйте содержимое, **замените `solr_url` в `xasecure.audit.solr.solr_url` на ваш собственный `solr_url`**, и сохраните файл.

8. Добавьте конфигурацию `access_control = ranger` во все файлы конфигурации FE.

```
vim fe.conf
access_control=ranger
```

9. Перезапустите все машины FE.

```
-- Переключитесь на папку FE.
cd..

bin/stop_fe.sh
bin/start_fe.sh
```

Повторное использование других сервисов для контроля доступа к внешним таблицам

Для External Catalog вы можете повторно использовать внешние сервисы (такие как Hive Service) для контроля доступа. Selena поддерживает сопоставление различных внешних сервисов Ranger для различных каталогов. Когда пользователи обращаются к внешней таблице, система реализует контроль доступа на основе политики доступа сервиса Ranger, соответствующего внешней таблице. Права пользователей согласуются с пользователем Ranger с таким же именем.

1. Скопируйте файлы конфигурации Ranger для Hive ranger-hive-security.xml и ranger-hive-audit.xml в файл `fe/conf` всех машин FE. Убедитесь, что IP и порт Ranger указаны правильно.

2. Перезапустите все машины FE.

3. Настройте External Catalog.

   - При создании External Catalog добавьте свойство `"ranger.plugin.hive.service.name"`.

     ```
     CREATE EXTERNAL CATALOG hive_catalog_1
     PROPERTIES (
         "type" = "hive",
         "hive.metastore.type" = "hive",
         "hive.metastore.uris" = "thrift://xx.xx.xx.xx:9083",
         "ranger.plugin.hive.service.name" = "
<ranger_hive_service_name>"
     )
     ```

   - Вы также можете добавить это свойство в существующий External Catalog.

```
ALTER CATALOG hive_catalog_1
SET ("ranger.plugin.hive.service.name" = "
<ranger_hive_service_name>");
```

Эта операция изменяет метод аутентификации существующего каталога на аутентификацию на основе Ranger.

Что делать дальше

После добавления сервиса Selena вы можете щелкнуть на сервис, чтобы создать политики контроля доступа для сервиса и назначить различные права доступа различным пользователям или группам пользователей. Когда пользователи обращаются к данным Selena, контроль доступа будет реализован на основе этих политик.

Часто задаваемые вопросы о привилегиях

Почему сообщение об ошибке "нет разрешения" все еще появляется, даже после того, как необходимая роль была назначена пользователю?

Эта ошибка может возникнуть, если роль не активирована. Вы можете выполнить `select current_role();`, чтобы запросить роли, активированные для пользователя в текущей сессии. Если необходимая роль не активирована, выполните SET ROLE, чтобы активировать эту роль и выполнять операции с использованием этой роли.

Если вы хотите, чтобы роли автоматически активировались при входе в систему, роль `user_admin` может выполнить SET DEFAULT ROLE или ALTER USER DEFAULT ROLE, чтобы установить роль по умолчанию для каждого пользователя. После установки роли по умолчанию она будет автоматически активироваться при входе пользователя в систему.

Если вы хотите, чтобы все назначенные роли всех пользователей автоматически активировались при входе в систему, вы можете выполнить следующую команду. Эта операция требует разрешения OPERATE на уровне System.

```
SET GLOBAL activate_all_roles_on_login = TRUE;
```

Однако мы рекомендуем следовать принципу "наименьших привилегий", устанавливая роли по умолчанию с ограниченными привилегиями для предотвращения потенциальных рисков. Например:

- Обычные пользователи могут установить роль `read_only`, которая имеет только привилегию SELECT, в качестве роли по умолчанию, избегая установки ролей с привилегиями, такими как ALTER, DROP и INSERT, в качестве ролей по умолчанию.
- Администраторы могут установить роль `db_admin` в качестве роли по умолчанию, избегая установки роли `node_admin`, которая имеет привилегию добавлять и удалять узлы, в качестве роли по умолчанию.

Этот подход помогает гарантировать, что пользователям назначены роли с соответствующими разрешениями, снижая риск непреднамеренных операций.

Вы можете выполнить GRANT, чтобы назначить необходимые привилегии или роли пользователям.

Я предоставил пользователю привилегию на все таблицы в базе данных (GRANT ALL ON ALL TABLES IN DATABASE <db_name> TO USER <user_identity>;), но пользователь все равно не может создавать таблицы в базе данных. Почему?

Создание таблиц в базе данных требует привилегии CREATE TABLE на уровне базы данных. Вам нужно предоставить эту привилегию пользователю.

```
GRANT CREATE TABLE ON DATABASE <db_name> TO USER <user_identity>;;
```

Я предоставил пользователю все привилегии на базу данных с помощью GRANT ALL ON DATABASE <db_name> TO USER <user_identity>;, но ничего не возвращается, когда пользователь выполняет SHOW TABLES; в этой базе данных. Почему?

`SHOW TABLES;` возвращает только те таблицы, на которые у пользователя есть какие-либо привилегии. Если у пользователя нет привилегий на таблицу, эта таблица не будет возвращена. Вы можете предоставить любую привилегию на все таблицы в этой базе данных (используя, например, SELECT) пользователю:

```
GRANT SELECT ON ALL TABLES IN DATABASE <db_name> TO USER <user_identity>;
```

Приведенное выше утверждение эквивалентно `GRANT select_priv ON db.* TO <user_identity>;`, используемому в версиях до v1.5.2.

Какие привилегии требуются для доступа к Selena Web Console http://<fe_ip>:<fe_http_port>?

Пользователь должен иметь роль `cluster_admin`.

Как изменился механизм сохранения привилегий до и после Selena v1.5.2?

До v1.5.2, после того как пользователю были предоставлены привилегии на таблицу, привилегии сохранялись, даже если таблица была удалена и воссоздана. Начиная с v1.5.2, привилегии больше не сохраняются после удаления и пересоздания таблицы.

Как запросить пользователей и предоставленные привилегии в Selena?

Вы можете получить полный список пользователей, запросив системное представление `sys.grants_to_users` или выполнив SHOW USERS, а затем запросить каждого пользователя отдельно, используя `SHOW GRANTS FOR <user_identity>`.

Каково влияние на ресурсы FE при запросе системных представлений о метаданных привилегий большого количества пользователей и таблиц?

Когда количество пользователей или таблиц очень велико, запросы к системным представлениям `sys.grants_to_users`, `sys.grants_to_roles` и `sys.role_edges` могут занять много времени. Эти представления вычисляются в режиме реального времени, потребляя часть ресурсов FE. Поэтому не рекомендуется часто выполнять такие операции в большом масштабе.

Приведет ли пересоздание catalog к потере разрешений? Как следует резервировать и восстанавливать привилегии?

Да. Пересоздание catalog приведет к потере связанных с ним привилегий. Вам следует сначала создать резервную копию всех привилегий пользователей, а затем восстановить их после пересоздания catalog.

Есть ли инструмент, поддерживающий автоматическую миграцию разрешений?

На данный момент нет. Пользователи должны вручную резервировать и восстанавливать разрешения, используя SHOW GRANTS для каждого пользователя.

Есть ли ограничения на использование команды KILL? Можно ли ограничить её только завершением собственных запросов пользователя?

Да. Команда KILL теперь требует привилегии OPERATE, и пользователь может завершать только те запросы, которые были инициированы им самим.

Почему предоставленные привилегии изменяются после переименования или удаления таблицы? Может ли система сохранить старые разрешения при добавлении разрешений для переименованной таблицы?

Для нативных таблиц привилегии привязаны к ID таблицы, а не к её имени. Это обеспечивает безопасность данных, поскольку имена таблиц могут изменяться произвольно. Если бы привилегии следовали за именами таблиц, это могло бы привести к утечке данных. Аналогично, когда таблица удаляется, её разрешения удаляются, потому что объект больше не существует.

Для внешних таблиц исторические версии вели себя так же, как и внутренние таблицы. Однако, поскольку метаданные внешних таблиц не управляются Selena, могут возникать задержки или потери привилегий. Для решения этой проблемы будущие версии будут использовать управление разрешениями на основе имени таблицы для внешних таблиц, что соответствует ожидаемому поведению.

Как создать резервную копию привилегий пользователей?

Ниже приведен пример скрипта для резервного копирования информации о привилегиях пользователей в cluster.

```bash
#!/bin/bash

# MySQL connection info
HOST=""
PORT="9030"
USER="root"
PASSWORD=""
OUTPUT_FILE="user_privileges.txt"

# Clear output file
> $OUTPUT_FILE

# Get user list
users=$(mysql -h$HOST -P$PORT -u$USER -p$PASSWORD -e "SHOW USERS;" | sed -e
'1d' -e '/^+/d')

# Loop through users and get privileges
for user in $users; do
    echo "Privileges for $user:" >> $OUTPUT_FILE
    mysql -h$HOST -P$PORT -u$USER -p$PASSWORD -e "SHOW GRANTS FOR $user;"
>> $OUTPUT_FILE
    echo "" >> $OUTPUT_FILE
done

echo "All user privileges have been written to $OUTPUT_FILE"
```

Почему предоставление USAGE для обычной функции выдает ошибку "Unexpected input 'IN', the most similar input is TO."? Каков правильный способ предоставления разрешений на функции?

Обычным функциям нельзя предоставлять разрешения, используя IN ALL DATABASES; они могут быть предоставлены только в пределах текущей базы данных. В то время как глобальные функции предоставляются в масштабе ALL DATABASES.

2.8.3 Аутентификация групп пользователей

Включите Group Provider в Selena для аутентификации и авторизации групп пользователей из внешних систем аутентификации.

Начиная с версии v1.5.2, Selena поддерживает Group Provider для сбора информации о группах из внешних систем аутентификации для управления группами пользователей.

Обзор

Для углубления интеграции с внешними системами аутентификации и авторизации пользователей, такими как LDAP и Apache Ranger, Selena поддерживает сбор информации о группах пользователей для улучшения работы с коллективным управлением пользователями.

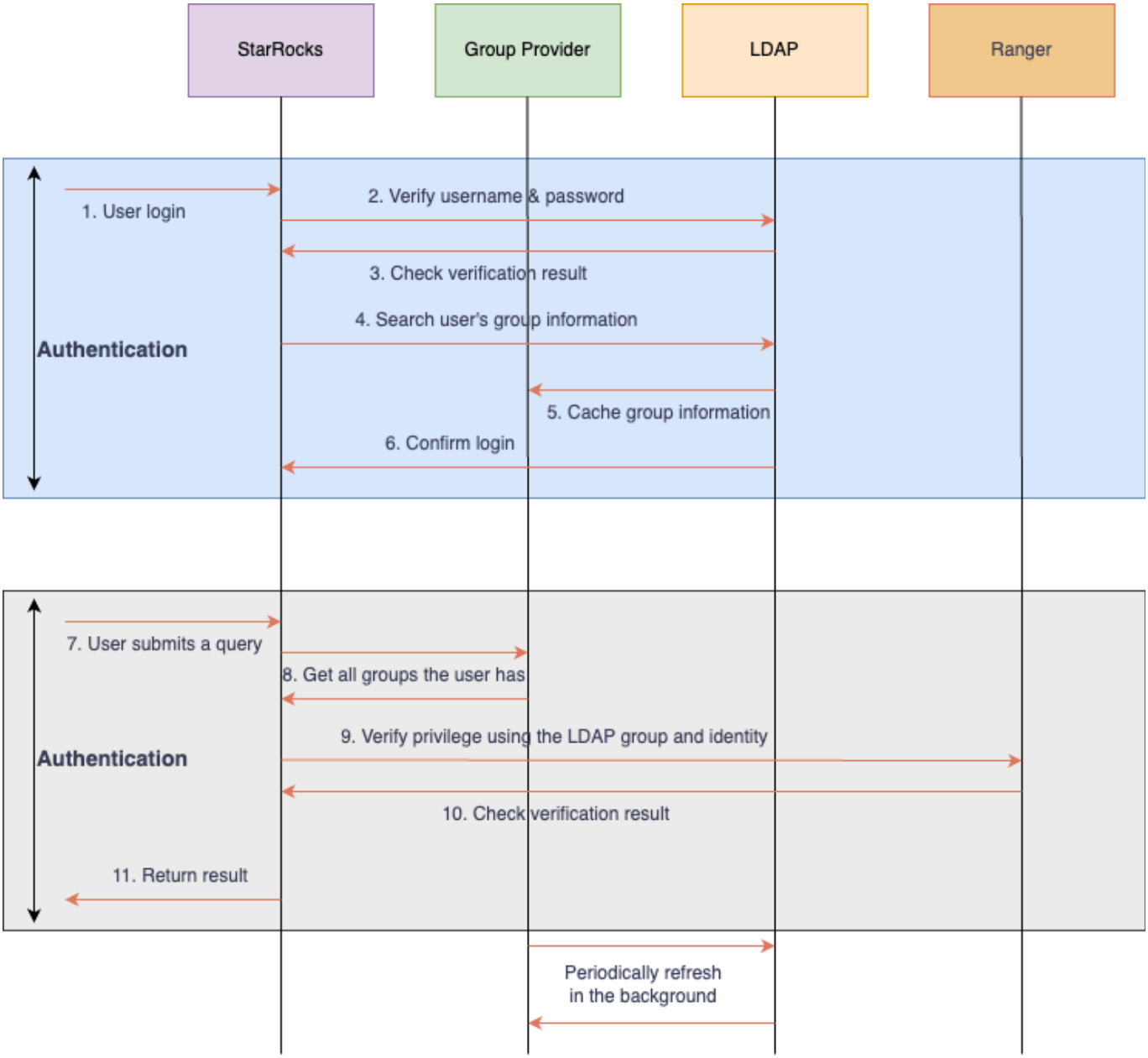
С помощью группового провайдера вы можете получать информацию о группах из внешних пользовательских систем для различных целей. Информация о группах независима и может быть гибко интегрирована в процессы аутентификации, авторизации или другие процессы, не будучи жестко связанной с каким-либо конкретным рабочим процессом.

Group Provider по сути представляет собой сопоставление между пользователями и группами. Любой процесс, требующий информации о группах, может запрашивать это сопоставление по мере необходимости.

Рабочий процесс

На следующей блок-схеме используются LDAP и Apache Ranger в качестве примера для объяснения рабочего процесса Group Provider.

© ООО «Платформа Данных Селена», 2026



Создание группового провайдера

Selena поддерживает три типа провайдеров групп:

- **Провайдер групп LDAP**: Поиск и сопоставление пользователей с группами в вашем сервисе LDAP
- **Провайдер групп Unix**: Поиск и сопоставление пользователей с группами в вашей операционной системе
- **Провайдер групп на основе файла**: Поиск и сопоставление пользователей с группами, определёнными в файле

Синтаксис

- LDAP group provider:

```
CREATE GROUP PROVIDER <group_provider_name>
PROPERTIES (
    "type" = "ldap",
    ldap_info,
    ldap_search_group_arg,
    ldap_search_attr,
    [ldap_cache_attr]
)

ldap_info ::=
    "ldap_conn_url" = "",
    "ldap_bind_root_dn" = "",
    "ldap_bind_root_pwd" = "",
    "ldap_bind_base_dn" = "",
    ["ldap_conn_timeout" = "",]
    ["ldap_conn_read_timeout" = ""]
    ["ldap_ssl_conn_allow_insecure" = ""]
    ["ldap_ssl_conn_trust_store_path" = ""]
    ["ldap_ssl_conn_trust_store_pwd" = ""]

ldap_search_group_arg ::=
    { "ldap_group_dn" = ""
    | "ldap_group_filter" = "" },
    "ldap_group_identifier_attr" = ""

ldap_search_user_arg ::=
    "ldap_group_member_attr" = "",
    "ldap_user_search_attr" = ""

ldap_cache_arg ::=
    "ldap_cache_refresh_interval" = ""
```

- Unix group provider:

```
CREATE GROUP PROVIDER <group_provider_name>
PROPERTIES (
    "type" = "unix"
)
```

- File group provider:

```
CREATE GROUP PROVIDER <group_provider_name>
PROPERTIES (
    "type" = "file",
    "group_file_url" = ""
)
```

Параметры

`type`

Тип создаваемого group provider. Допустимые значения:

- `ldap`: Создаёт LDAP group provider. При установке этого значения необходимо указать `ldap_info`, `ldap_search_group_arg`, `ldap_search_user_arg` и, опционально, `ldap_cache_arg`.
- `unix`: Создаёт Unix group provider.
- `file`: Создаёт File group provider. При установке этого значения необходимо указать `group_file_url`.

`group_file_url`

URL или относительный путь (в директории `fe/conf`) к файлу, определяющему группы пользователей.

> **ПРИМЕЧАНИЕ**
>
> Файл групп содержит список групп и их участников. Вы можете определить группу в каждой строке, где имя группы и участники разделены двоеточием. Несколько пользователей разделяются запятыми. Пример: `group_name:user_1,user_2,user_3`.

Группа параметров `ldap_info`

Информация, используемая для подключения к вашему сервису LDAP.

`ldap_conn_url`

URL вашего сервера LDAP. Формат: `ldap://<ldap_server_host>:<ldap_server_port>)`.

`ldap_bind_root_dn`

Администраторское Distinguished Name (DN) вашего сервиса LDAP.

`ldap_bind_root_pwd`

Пароль администратора вашего сервиса LDAP.

`ldap_bind_base_dn`

Базовый DN пользователя LDAP, для которого кластер выполняет поиск.

`ldap_conn_timeout`

Опционально. Время ожидания подключения к вашему сервису LDAP.

`ldap_conn_read_timeout`

Опционально. Время ожидания операций чтения в подключении к вашему сервису LDAP.

`ldap_ssl_conn_allow_insecure`

Опционально. Разрешать ли незашифрованные подключения к серверу LDAP. Значение по умолчанию: `true`. Установка этого значения в `false` означает, что для доступа к LDAP требуется шифрование SSL.

`ldap_ssl_conn_trust_store_path`

Опционально. Локальный путь для хранения сертификата SSL CA сервера LDAP. Поддерживает форматы pem и jks. Вам не нужно устанавливать этот параметр, если сертификат выдан доверенной организацией.

`ldap_ssl_conn_trust_store_pwd`

Опционально. Пароль, используемый для доступа к локально сохраненному сертификату SSL CA сервера LDAP. Сертификаты в формате pem не требуют пароля. Только сертификаты в формате jsk требуют пароль.

Группа параметров `ldap_search_group_arg`

Аргументы, используемые для управления тем, как Selena ищет группу.

> **ПРИМЕЧАНИЕ**
>
> Вы можете указать только `ldap_group_dn` или `ldap_group_filter`. Указание обоих параметров не поддерживается.

`ldap_group_dn`

DN группы для поиска. Группа будет запрошена напрямую с использованием этого DN. Пример: `"cn=ldapgroup1,ou=Group,dc=selena,dc=com;cn=ldapgroup2,ou=Group,dc=selena,dc=com"`.

`ldap_group_filter`

Пользовательский фильтр группы, который может быть распознан сервером LDAP. Он будет отправлен напрямую на ваш сервер LDAP для поиска группы. Пример: `(&(objectClass=groupOfNames)(cn=testgroup))`.

`ldap_group_identifier_attr`

Атрибут, используемый в качестве идентификатора имени группы.

Группа параметров `ldap_search_user_arg`

Аргументы, используемые для управления тем, как Selena идентифицирует пользователя в группе.

`ldap_group_member_attr`

Атрибут, представляющий членов группы. Допустимые значения: `member` и `memberUid`.

`ldap_user_search_attr`

Указывает, как извлечь идентификатор пользователя из значения атрибута member. Вы можете явно определить атрибут (например, `cn` или `uid`) или использовать регулярное выражение.

> **ПРИМЕЧАНИЕ**
>
> **Механизм сопоставления DN**
>
> - **Когда настроен** `ldap_user_search_attr`, система извлекает указанное значение из DN членов группы и использует его в качестве имен пользователей, и использует имя пользователя для входа в качестве ключа при поиске группы.
> - **Когда** `ldap_user_search_attr` **не настроен**, система использует полный DN напрямую в качестве идентификатора пользователя и использует DN, записанный во время аутентификации, в качестве ключа при поиске группы.
>
> Этот дизайн позволяет LDAP Group Provider адаптироваться к различным средам LDAP, особенно к сложным средам, таким как Microsoft AD.

Группа параметров `ldap_cache_arg`

Аргумент, используемый для определения поведения кэширования информации о группах LDAP.

`ldap_cache_refresh_interval`

Опционально. Интервал, с которым Selena автоматически обновляет кэшированную информацию о группах LDAP. Единица измерения: секунды. По умолчанию: `900`.

Пример

Предположим, что сервер LDAP содержит следующую информацию о группах и членах.

```
-- Информация о группе
# testgroup, Group, selena.com
dn: cn=testgroup,ou=Group,dc=selena,dc=com
objectClass: groupOfNames
cn: testgroup
member: uid=test,ou=people,dc=selena,dc=com
member: uid=tom,ou=people,dc=selena,dc=com

-- Информация о пользователе
# test, People, selena.com
dn: cn=test,ou=People,dc=selena,dc=com
objectClass: inetOrgPerson
cn: test
uid: test
sn: FTE
userPassword::
```

Создайте групповой провайдер `ldap_group_provider` для членов в `testgroup`:

```
CREATE GROUP PROVIDER ldap_group_provider
PROPERTIES(
    "type"="ldap",
    "ldap_conn_url"="ldap://xxxx:xxx",
    "ldap_bind_root_dn"="cn=admin,dc=selena,dc=com",
    "ldap_bind_root_pwd"="123456",
    "ldap_bind_base_dn"="dc=selena,dc=com",
    "ldap_group_filter"="(&(objectClass=groupOfNames)(cn=testgroup))",
    "ldap_group_identifier_attr"="cn",
    "ldap_group_member_attr"="member",
    "ldap_user_search_attr"="uid=([^,]+)"
)
```

Вышеприведенный пример использует `ldap_group_filter` для поиска группы с objectClass `groupOfNames` и `cn` равным `testgroup`. Поэтому `cn` указан в `ldap_group_identifier_attr` для идентификации группы. `ldap_group_member_attr` установлен в `member`, чтобы атрибут `member` использовался в objectClass `groupOfNames` для идентификации членов. `ldap_user_search_attr` установлен в выражение `uid=([^,]+)`, которое используется для идентификации пользователей в атрибуте `member`.

Пример среды Microsoft AD

Предположим, что сервер Microsoft AD содержит следующую информацию о группах и членах:

```
-- Информация о группе
# ADGroup, Groups, company.com
dn: CN=ADGroup,OU=Groups,DC=company,DC=com
objectClass: group
cn: ADGroup
member: CN=John Doe,OU=Users,DC=company,DC=com
member: CN=Jane Smith,OU=Users,DC=company,DC=com

-- Информация о пользователе
# John Doe, Users, company.com
dn: CN=John Doe,OU=Users,DC=company,DC=com
objectClass: user
cn: John Doe
sAMAccountName: johndoe
```

Создайте Group Provider для среды Microsoft AD:

```
CREATE GROUP PROVIDER ad_group_provider
PROPERTIES(
    "type"="ldap",
    "ldap_conn_url"="ldap://ad.company.com:389",
    "ldap_bind_root_dn"="CN=admin,OU=Users,DC=company,DC=com",
    "ldap_bind_root_pwd"="password",
    "ldap_bind_base_dn"="DC=company,DC=com",
    "ldap_group_filter"="(&(objectClass=group)(cn=ADGroup))",
    "ldap_group_identifier_attr"="cn",
    "ldap_group_member_attr"="member"
    -- Примечание: Не настраивайте ldap_user_search_attr, система будет
использовать полный DN для сопоставления
)
```

В этом примере, поскольку `ldap_user_search_attr` не настроен, система будет:

1. Во время создания кэша группы напрямую использовать полный DN (например, `CN=John Doe,OU=Users,DC=company,DC=com`) в качестве идентификатора пользователя.

2. Во время поиска группы использовать DN, записанный во время аутентификации, в качестве ключа для поиска групп пользователя.

Этот подход особенно подходит для сред Microsoft AD, поскольку члены групп в AD могут не иметь простых атрибутов имени пользователя.

Объединение группового провайдера с security integration

После создания группового провайдера вы можете объединить его с security integration, чтобы разрешить пользователям, указанным групповым провайдером, входить в Selena. Для получения дополнительной информации о создании security integration см. Аутентификация с помощью Security Integration.

Синтаксис

```
ALTER SECURITY INTEGRATION <security_integration_name> SET
(
    "group_provider" = "",
    "permitted_groups" = ""
)
```

Параметры

`group_provider`

Имя группового провайдера(ов) для объединения с security integration. Несколько групповых провайдеров разделяются запятыми. После установки Selena будет записывать информацию о группе пользователя под каждым указанным провайдером при входе в систему.

`permitted_groups`

Опционально. Имя группы(групп), члены которых имеют право входить в Selena. Несколько групп разделяются запятыми. Убедитесь, что указанные группы могут быть получены объединенным групповым провайдером(ами).

Пример

```
ALTER SECURITY INTEGRATION LDAP SET
(
        "group_provider"="ldap_group_provider",
        "permitted_groups"="testgroup"
);
```

Предоставление роли группе пользователей

Вы можете предоставлять роли группе пользователей через GRANT.

Следующий пример предоставляет роль `example_role` группе пользователей `analysts`:

```
GRANT example_role TO EXTERNAL GROUP analysts;
```

Объединение группового провайдера с внешней системой авторизации (Apache Ranger)

После настройки связанного группового провайдера в security integration, Selena будет записывать информацию о группе пользователя при входе в систему. Эта информация о группе будет затем автоматически включена в процесс авторизации с Ranger, что устраняет необходимость в дополнительной настройке.

Дополнительные инструкции по интеграции Selena с Ranger см. в разделе Управление разрешениями с помощью Apache Ranger.

2.8.4 SSL-аутентификация

Начиная с версии v1.5.2, Selena поддерживает безопасные соединения, зашифрованные с помощью SSL. В отличие от традиционных незашифрованных соединений с СУБД, соединения SSL обеспечивают проверку конечных точек и шифрование данных, гарантируя, что данные, передаваемые между клиентами и Selena, не могут быть прочитаны неавторизованными пользователями.

Включение SSL-аутентификации

Настройка FE-узлов

Чтобы включить SSL-аутентификацию в Selena, настройте следующие параметры в конфигурационном файле FE **fe.conf**:

- `ssl_keystore_location`: Указывает путь к файлу keystore, который хранит SSL-сертификат и ключ.
- `ssl_keystore_password`: Пароль для доступа к файлу keystore. Selena требует этот пароль для чтения файла keystore.
- `ssl_key_password`: Пароль для доступа к ключу. Selena требует этот пароль для извлечения ключа из keystore.
- `ssl_force_secure_transport`: Принудительно использовать SSL-аутентификацию. Значение по умолчанию: `FALSE`. Если этот параметр установлен в `TRUE`, система будет отклонять соединения, не зашифрованные с помощью SSL.

Пример:

```
ssl_keystore_location = // Путь к файлу keystore
ssl_keystore_password = // Пароль для файла keystore
ssl_key_password = // Пароль для доступа к ключу
```

Генерация SSL-сертификата

В production-среде рекомендуется использовать сертификаты, предоставленные центрами сертификации.

В среде разработки вы можете сгенерировать пользовательский SSL-сертификат. Используйте следующую команду для генерации SSL-сертификата:

```
keytool -genkeypair -alias selena \
    -keypass <ssl_key_password> \
    -keyalg RSA -keysize 1024 -validity 365 \
    -keystore <ssl_keystore_location> \
    -storepass <ssl_keystore_password>
```

Параметры:

- `-keypass`: Пароль ключа, соответствует `ssl_key_password` в **fe.conf**.
- `-storepass`: Пароль файла keystore, соответствует `ssl_keystore_password` в **fe.conf**.
- `-keystore`: Путь хранения файла keystore, соответствует `ssl_keystore_location` в **fe.conf**.

Включение SSL на клиенте

Selena совместима с протоколом MySQL. Для клиентов MySQL SSL-аутентификация включена по умолчанию.

Для JDBC-соединений добавьте следующие опции:

```
useSSL=true
verifyServerCertificate=false
```

Отключение SSL-аутентификации

Чтобы отключить SSL-аутентификацию, выполните следующие шаги:

- **Клиент MySQL**: Добавьте опцию `--ssl-mode=DISABLED`.
- **JDBC**: Удалите `useSSL=true` и `verifyServerCertificate=false`.

LDAP-аутентификация

См. Методы аутентификации для подробных инструкций по включению LDAP-аутентификации.

Для JDBC-соединений, поскольку Selena поддерживает SSL-аутентификацию, вам не нужно настраивать пользовательский `AuthPlugin`. Вы можете использовать встроенный `MysqlClearPasswordPlugin`.

- При использовании JDBC 5 с LDAP-аутентификацией настройте следующие параметры:

```
authenticationPlugins:
com.mysql.jdbc.authentication.MysqlClearPasswordPlugin
defaultAuthenticationPlugin:
com.mysql.jdbc.authentication.MysqlClearPasswordPlugin
disabledAuthenticationPlugins:
com.mysql.jdbc.authentication.MysqlNativePasswordPlugin
```

- При использовании JDBC 8 с LDAP-аутентификацией настройте следующие параметры:

```
authenticationPlugins:
com.mysql.cj.protocol.a.authentication.MysqlClearPasswordPlugin
defaultAuthenticationPlugin:
com.mysql.cj.protocol.a.authentication.MysqlClearPasswordPlugin
disabledAuthenticationPlugins:
com.mysql.cj.protocol.a.authentication.MysqlNativePasswordPlugin
```

FAQ

Q1: При подключении к Selena с использованием DBeaver возвращается ошибка "Unable to load authentication plugin 'mysql_native_password'".

A: Вам необходимо обновить JDBC 5 до версии 5.1.46 или выше.

3 Benchmarks

3.1 SSB Flat-table Benchmarking

Star schema benchmark (SSB) разработан для тестирования базовых метрик производительности продуктов OLAP-баз данных. SSB использует набор тестов со схемой "звезда", который широко применяется в академических и промышленных кругах. Для получения дополнительной информации см. Star Schema Benchmark.

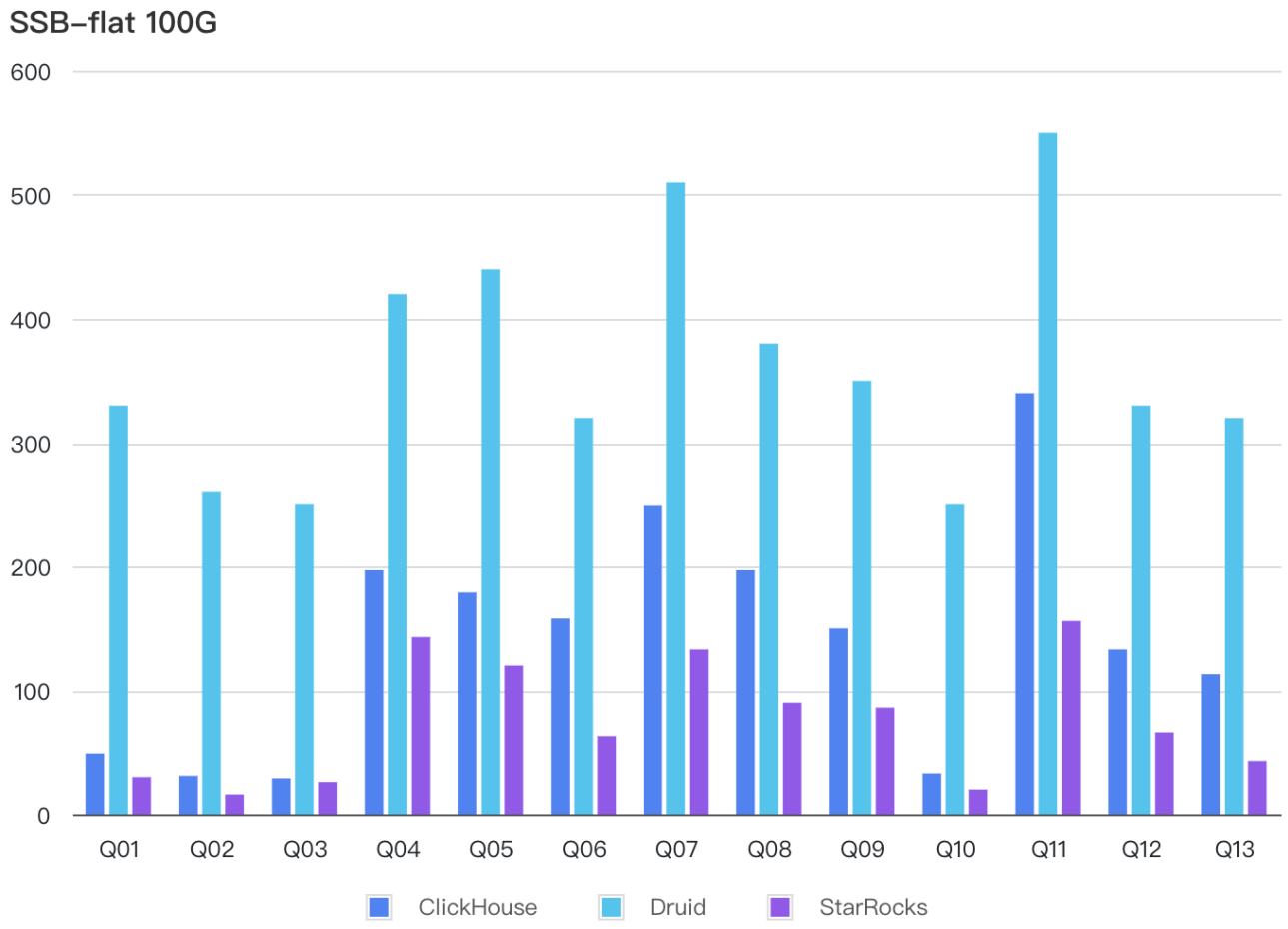
ClickHouse преобразует схему "звезда" в широкую плоскую таблицу и переписывает SSB в однотабличный benchmark. Для получения дополнительной информации см. Star schema benchmark of ClickHouse

Этот тест сравнивает производительность Selena, Apache Druid и ClickHouse на однотабличных наборах данных SSB.

3.1.1 Выводы тестирования

Тестирование выполняется на OLAP-таблице в shared-nothing cluster Selena, вместе с ClickHouse и Apache Druid, на одном и том же наборе данных.

Основываясь на результатах 13 запросов, выполненных на наборе данных SSB-Flat размером 100 ГБ, общая производительность запросов Selena **в 1.87 раза выше, чем у ClickHouse, и в 4.75 раза выше, чем у Apache Druid**. Единица измерения результатов — миллисекунды.



3.1.2 Подготовка к тестированию

Оборудование

Selena, Apache Druid и ClickHouse развернуты на хостах с одинаковой конфигурацией - AWS m7i.4xlarge.

	Спецификация
Количество инстансов	5
vCPU	16
Память (GiB)	64
Пропускная способность сети (Gbps)	До 12.5
Пропускная способность EBS (Gbps)	До 10

Программное обеспечение

	Selena	ClickHouse	Apache Druid
Размер cluster	Один FE, три BE	Три узла	Один Master Server, один Query Server и три Data Server

	Selena	ClickHouse	Apache Druid
Версия	3.5.0	25.3.3.42	33.0.0
Дата релиза	2025.6.13	2025.4.22	2025.4.29
Конфигурация	По умолчанию	По умолчанию	По умолчанию

3.1.3 Результаты тестирования

В следующей таблице показаны результаты тестирования производительности для 13 запросов. Единица измерения задержки запроса — миллисекунды. Все запросы прогреваются 1 раз, затем выполняются 3 раза, и берется среднее значение в качестве результата. `ClickHouse vs Selena` и `Druid vs Selena` в заголовке таблицы означают деление времени ответа запроса ClickHouse/Druid на время ответа запроса Selena. Большее значение указывает на лучшую производительность Selena.

Запрос	Selena	ClickHouse	Druid	ClickHouse vs Selena	Druid vs Selena
SUM	992	1858	4710	1.87	4.75
Q01	30	49	330	1.63	11.00
Q02	16	31	260	1.94	16.25
Q03	26	29	250	1.12	9.62
Q04	143	197	420	1.38	2.94
Q05	120	179	440	1.49	3.67
Q06	63	158	320	2.51	5.08
Q07	133	249	510	1.87	3.83
Q08	90	197	380	2.19	4.22
Q09	86	150	350	1.74	4.07
Q10	20	33	250	1.65	12.50
Q11	156	340	550	2.18	3.53
Q12	66	133	330	2.02	5.00
Q13	43	113	320	2.63	7.44

3.2 TPC-DS Benchmarking

TPC-DS — это benchmark системы поддержки принятия решений, разработанный Transaction Processing Performance Council (TPC). Он использует более полные тестовые наборы данных и сложные SQL-запросы, чем TPC-H.

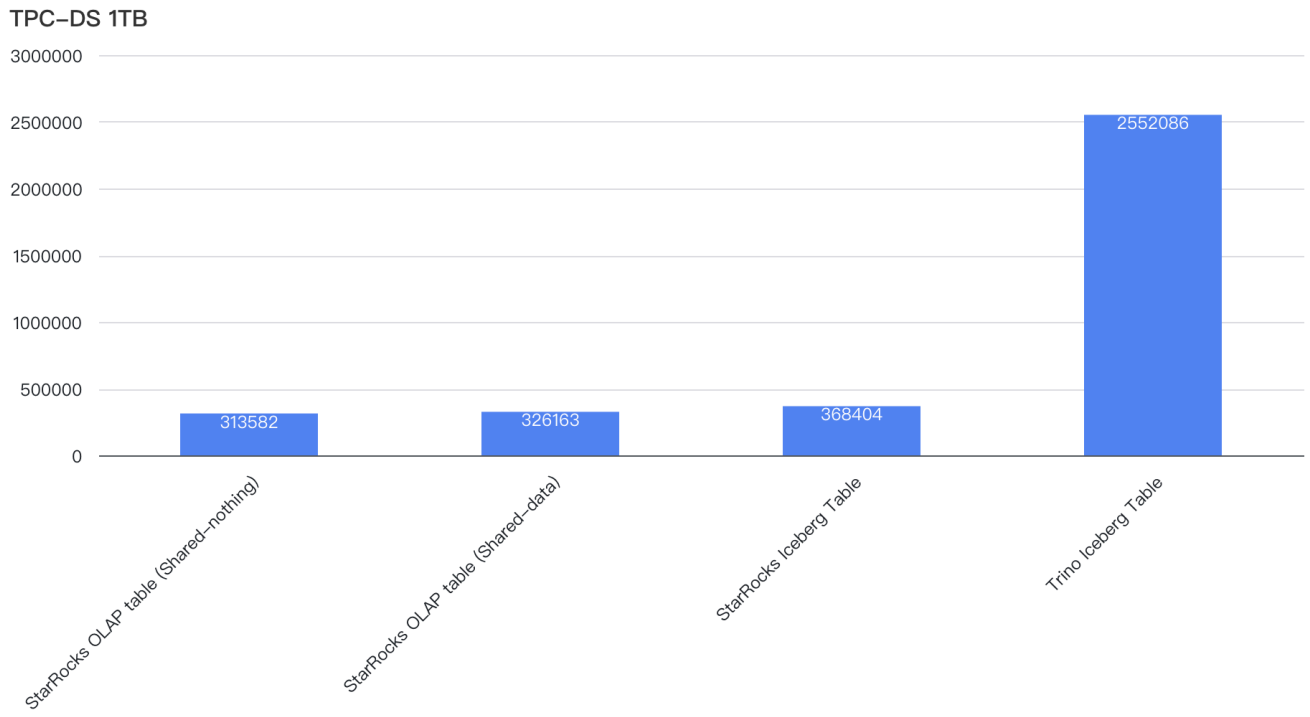
TPC-DS моделирует несколько общеприменимых аспектов системы поддержки принятия решений, включая запросы и обслуживание данных. TPC-DS нацелен на предоставление

комплексной и реалистичной рабочей нагрузки для тестирования и оценки производительности систем баз данных в среде розничной торговли. Benchmark TPC-DS моделирует данные о продажах и возвратах трех каналов продаж (магазины, интернет и каталог) в розничном предприятии. Помимо создания таблиц для моделей данных о продажах и возвратах, он также включает простую систему учета запасов и систему акций.

Этот benchmark тестирует в общей сложности 99 сложных SQL-запросов к 24 таблицам, размер данных которых варьируется от 1 GB до 3 TB. Основной метрикой производительности является время отклика каждого запроса, которое представляет собой длительность между моментом отправки запроса и моментом возврата результата.

3.2.1 Результаты тестирования

Тест выполнен на наборе данных TPC-DS объемом 1 TB в Selena и Trino. Единица измерения следующих результатов — миллисекунды.



Для Selena тестовые запросы выполняются на нативных таблицах (в архитектурах shared-nothing и shared-data) и Iceberg Catalog (в архитектуре shared-data). Для Trino тестовые запросы выполняются на том же Iceberg Catalog. Оба теста для Selena и Trino на Iceberg Catalog используют AWS Glue в качестве metastore и данные в формате Parquet со сжатием ZSTD.

Результаты теста:

- Запросы к OLAP-таблице в cluster Selena в режиме shared-nothing заняли 314 секунд.
- Запросы к OLAP-таблице в cluster Selena в режиме shared-data заняли 326 секунд.
- Запросы к Iceberg Catalog в cluster Selena в режиме shared-data заняли 368 секунд.
- Запросы к Iceberg Catalog в Trino заняли 2552 секунды.

Выводы:

- **Selena обеспечивает сопоставимую производительность как в архитектуре shared-nothing, так и в shared-data**

  В сценариях запросов к OLAP-таблицам Selena в режиме shared-nothing достигает производительности в 1.04× от режима shared-data, что указывает на минимальные накладные расходы от разделения хранения и вычислений.

- **Selena значительно превосходит Trino в сценариях запросов Lakehouse из коробки, с еще большими преимуществами в запросах к OLAP-таблицам**

  - В сценариях запросов к Iceberg Catalog Selena обеспечивает производительность в 6.93× выше Trino.
  - В запросах к OLAP-таблицам разрыв в производительности еще больше — Selena достигает производительности в 8.13× (в режиме shared-nothing) и 7.82× (в режиме shared-data) выше Trino.

3.2.2 Подготовка к тестированию

Оборудование

Selena и Trino развернуты на хостах с одинаковыми конфигурациями - AWS m6id.4xlarge.

	Spec
Instance Number	5
vCPU	16
Memory (GiB)	64
Instance Storage (GB)	1x950 NVMe SSD
Network Bandwidth (Gbps)	Up to 12.5
EBS Bandwidth (Gbps)	Up to 10

Программное обеспечение

	Selena	Trino
Cluster Size	Один FE, четыре BE	Один Coordinator, четыре Workers
Version	3.5.0	475
Release Date	2025.6.13	2025.4.23
Configuration	`set query_mem_limit=66571993088;` (62 GB)	`query.max-memory=224GB` `query.max-memory-per-node=35GB`

3.2.3 Результаты тестирования

В следующей таблице показаны результаты тестирования производительности на 99 запросах. Единица измерения задержки запроса — миллисекунды. Все запросы прогреваются 1 раз, затем выполняются 3 раза, и среднее значение берется в качестве результата. `Trino vs Selena on Iceberg` в заголовке таблицы означает деление времени отклика запроса Trino на время отклика запроса Selena. Большее значение указывает на лучшую производительность Selena. Перед тестированием на Iceberg Catalog в Selena выполняются операторы `ANALYZE TABLE` и `EXPLAIN COST` для сбора статистики.

Query	Selena			Trino	Trino vs Selena on Iceberg
	OLAP Table		Iceberg Catalog	Iceberg Catalog	
	Shared-nothing	Shared-data	Share-data		
SUM	313582	326163	368404	2552076	6.93
Q01	675	636	727	2276	3.13
Q02	935	925	1949	21289	10.92
Q03	220	140	399	1764	4.42
Q04	17185	16206	16744	200249	11.96
Q05	417	543	1724	5574	3.23
Q06	200	180	343	2909	8.48
Q07	814	959	1795	3638	2.03
Q08	235	181	319	2534	7.94
Q09	7097	6503	23153	36125	1.56
Q10	297	343	497	2471	4.97
Q11	11229	10249	9777	133230	13.63
Q12	128	130	207	1067	5.15
Q13	417	447	2218	18297	8.25

Query	Selena			Trino	Trino vs Selena on Iceberg
	OLAP Table		Iceberg Catalog	Iceberg Catalog	
	Shared-nothing	Shared-data	Share-data		
Q14	15112	15055	15310	143172	9.35
Q15	455	474	525	2158	4.11
Q16	476	523	975	14035	14.39
Q17	895	948	1645	5487	3.34
Q18	802	973	1544	4736	3.07
Q19	209	268	507	1870	3.69
Q20	147	176	253	1257	4.97
Q21	83	80	405	945	2.33
Q22	1462	1823	2384	7871	3.30
Q23	82887	84426	50719	488653	9.63
Q24	7649	9064	11902	43436	3.65
Q25	795	919	1404	4347	3.10
Q26	376	511	868	2961	3.41
Q27	458	532	1397	4219	3.02
Q28	6489	6004	18727	24678	1.32
Q29	1296	2168	2129	10640	5.00
Q30	295	330	529	2770	5.24
Q31	2172	2162	2535	10260	4.05
Q32	110	225	277	1035	3.74
Q33	235	245	734	2743	3.74
Q34	402	422	696	2455	3.53
Q35	1353	1393	1562	8023	5.14
Q36	553	559	1193	3342	2.80
Q37	263	243	217	6178	28.47
Q38	5002	5207	5519	38408	6.96
Q39	356	367	820	5696	6.95
Q40	124	134	575	1554	2.70
Q41	43	50	76	271	3.57
Q42	81	104	197	996	5.06
Q43	487	733	722	4108	5.69
Q44	2087	2102	9364	16468	1.76
Q45	431	470	514	2212	4.30

© ООО «Платформа Данных Селена», 2026

Query	Selena			Trino	Trino vs Selena on Iceberg
	OLAP Table		Iceberg Catalog	Iceberg Catalog	
	Shared-nothing	Shared-data	Share-data		
Q46	1468	1829	1976	4780	2.42
Q47	2921	3120	3130	42704	13.64
Q48	467	573	1441	12464	8.65
Q49	496	547	1715	3827	2.23
Q50	3768	5135	3899	21771	5.58
Q51	4826	4633	3244	12247	3.78
Q52	87	106	219	1007	4.60
Q53	446	635	955	1768	1.85
Q54	198	266	763	31864	41.76
Q55	82	105	221	1187	5.37
Q56	175	186	499	2133	4.27
Q57	1848	1787	1916	26042	13.59
Q58	241	245	503	3032	6.03
Q59	3935	3908	3160	20819	6.59
Q60	262	292	643	2749	4.28
Q61	330	344	913	2054	2.25
Q62	500	519	840	5104	6.08
Q63	448	629	937	1719	1.83
Q64	4508	8886	8602	16990	1.98
Q65	5690	5617	5445	20616	3.79
Q66	387	390	1335	3511	2.63
Q67	31286	29240	21439	89430	4.17
Q68	240	341	683	2883	4.22
Q69	260	292	501	2181	4.35
Q70	2904	3012	3751	21499	5.73
Q71	272	1111	631	2578	4.09
Q72	1642	1899	4969	504573	101.54
Q73	186	228	388	1933	4.98
Q74	9433	9133	8890	78071	8.78
Q75	6446	7718	9082	26270	2.89
Q76	2356	2331	5028	11266	2.24
Q77	238	241	701	2785	3.97

© ООО «Платформа Данных Селена», 2026

Query	Selena			Trino	Trino vs Selena on Iceberg
	OLAP Table		Iceberg Catalog	Iceberg Catalog	
	Shared-nothing	Shared-data	Share-data		
Q78	17583	16590	18614	51497	2.77
Q79	2201	2474	1597	5572	3.49
Q80	800	855	2672	4762	1.78
Q81	518	533	769	4040	5.25
Q82	623	613	854	11622	13.61
Q83	158	217	404	2274	5.63
Q84	195	434	351	3086	8.79
Q85	455	733	1087	5424	4.99
Q86	725	736	819	3422	4.18
Q87	4816	5004	5535	41300	7.46
Q88	8712	8444	19465	30738	1.58
Q89	530	730	915	2227	2.43
Q90	587	581	1150	3890	3.38
Q91	95	96	227	2035	8.96
Q92	78	140	228	994	4.36
Q93	3723	4705	5781	22926	3.97
Q94	601	963	962	5884	6.12
Q95	1747	3106	2575	29607	11.50
Q96	1444	1428	2562	4905	1.91
Q97	4795	4844	5383	22726	4.22
Q98	284	323	365	1911	5.24
Q99	1132	1184	1594	10970	6.88

© ООО «Платформа Данных Селена», 2026