



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 ПО УСТАНОВКЕ НА ВИРТУАЛЬНЫХ И ФИЗИЧЕСКИХ СЕРВЕРАХ (BAREMETAL)

Версия программы 3.0

Оглавление

1 Общая информация

- 1.1 Сценарии установки
- 1.2 Подготовительные действия
- 1.3 Быстрая проверка сервера перед установкой

2 Требования к оборудованию и программному обеспечению

- 2.1 Минимальные требования к ролям серверов
- 2.2 Поддерживаемые операционные системы

3 Подготовка хостов

- 3.1 Чек-лист подготовки хостов
- 3.2 Описание пререквизитов
- 3.3 Preflight — проверка хоста

4 Установка Selenia (online)

- 4.1 Доступ к реестру (online) + license
- 4.2 Настройка deploy.yaml
- 4.3 Установка
- 4.4 Проверка
- 4.5 После сборки Engine-кластера - перезапуск IDE (Querybook)
- 4.6 Обновление версий и конфигурации (upgrade.sh)

5 Установка Selenia (offline)

- 5.1 Бандл артефактов + инсталлятор
- 5.2 Доставка и распаковка
- 5.3 License + LLM-ключ
- 5.4 Установка
- 5.5 UI
- 5.6 Обновление дистрибутивов (engine upgrade)
- 5.7 Обновление версий и конфигурации (upgrade.sh)

6 Управление и обслуживание

- 6.1 Управление стеком
- 6.2 Дополнительные переменные окружения (extraEnv)
- 6.3 Удаление
- 6.4 Особенности работы upgrade.sh

7 Внешние PostgreSQL и Keycloak (consume-only)

- 7.1 Подготовка внешнего PostgreSQL
- 7.2 Подготовка внешнего Keycloak
- 7.3 Deploy.yaml
- 7.4 Секрет-файл
- 7.5 Установка и Engine bootstrap
- 7.6 Ограничения

8 Частые ошибки и решение

1 Общая информация

Руководство описывает нативную установку Платформы Данных Селена. Установка выполняется непосредственно в операционную систему на виртуальных или физических серверах (baremetal) с использованием системных пакетов и служб systemd. Подход не предполагает использования контейнерных сред (Podman, Docker) и предназначен для сред, где требуется прямое управление ресурсами и минимальный уровень абстракции.

⚠ ВАЖНО

Установку и конфигурацию программного обеспечения должны выполнять сертифицированные инженеры. По вопросам установки, обновления, получения дистрибутива и лицензии обращайтесь по адресу info@selenalakehouse.ru.

1.1 Сценарии установки

Таблица 1.1. Сценарии установки Selena (baremetal)

	Сценарий	Компоненты	Комментарий	Режим установки
1	All-in-One	Инфра + control-plane + Engine	Всё на одной VM	Online/offline
2	Инфра + control-plane	Инфра + control-plane (без Engine)	Engine устанавливается позже через Cluster Manager и может быть вынесен на отдельные ноды (multi-node)	

Режимы установки:

- Online — образы загружаются из hub.selena.sh.
- Offline — образы загружаются из офлайн-бандла, без обращения к внешнему реестру.

Поддерживается подключение baremetal-инсталляции Selena к уже развёрнутым экземплярам PostgreSQL и Keycloak, описание в главе 7 «Внешние PostgreSQL и Keycloak (consume-only)».

Конфигурация

Таблица 1.2. Файлы конфигурации для сценариев установки

Компоненты	Режим	deploy-конфиг	artifacts-конфиг
All-in-One (инфра + control-plane + Engine)	онлайн	examples/all-in-one.deploy.yaml	examples/artifacts.yaml
офлайн	examples/offline.deploy.yaml	examples/artifacts-offline.yaml	
Инфра + control-plane, полный набор (без Engine)	онлайн	examples/online.deploy.yaml	examples/artifacts.yaml
офлайн	examples/online.deploy.yaml	examples/artifacts-offline.yaml	
Только Cluster Manager (PostgreSQL + Keycloak + CM)	онлайн	examples/cm.deploy.yaml	examples/artifacts.yaml
офлайн	examples/cm.deploy.yaml	examples/artifacts-offline.yaml	
Любой из перечисленных + внешние PostgreSQL / Keycloak		examples/external-services.deploy.yaml как основа	По режиму

1.2 Подготовительные действия

Для установки компонентов системы вам необходимо подготовить:

- Серверы или виртуальные машины с достаточными ресурсами (см. главу 2), поддерживаемыми ОС и необходимыми компонентами (см. главу 3).
- Доступ по SSH с пользователем, который может выполнять команды через sudo.
- Для онлайн-установки - организовать сетевой доступ к hub.selena.sh.
- Для офлайн-установки - заранее подготовленный бандл с дистрибутивами компонентов.
- Лицензии на ПО.

△ За предоставлением бандла и лицензий следует обратиться по адресу: info@selena-lakehouse.ru.

1.3 Быстрая проверка сервера перед установкой

Выполните эти команды на каждой ВМ. Они помогают увидеть типовые проблемы: неподходящую ОС, нехватку диска, отсутствие sudo или сетевой недоступности.

- Проверить версию ОС:

```
cat /etc/os-release
```

- Проверить CPU, память и свободное место:

```
lscpu | egrep 'Model name|CPU\\(s\\)|Flags'  
free -h  
df -h /
```

- Проверить IP-адрес сервера:

```
ip -br addr
```

- Проверить, что пользователь может выполнять sudo:

```
sudo -v
```

ПРИМЕЧАНИЕ

Если в выводе `lscpu` нет флага `avx2`, сервер не подходит для ролей, где требуется CPU с AVX2 (BE и CN). Если свободного места меньше минимального значения, освободите диск или увеличьте том до начала установки.

2 Требования к оборудованию и программному обеспечению

Перед началом установки проверьте, что серверы соответствуют минимальным требованиям. Если ресурсы меньше указанных, установка может завершиться ошибкой или платформа будет работать нестабильно.

2.1 Минимальные требования к ролям серверов

Сервер	Кол-во	CPU	RAM	Диск	Сеть
All-in-One	1	8 vCPU, AVX2 обязателен	32 ГБ	200 ГБ SSD	1 × 10/25 Гбит/с
Инфра + control-plane (без Engine)	1	8 vCPU	16 ГБ	200 ГБ SSD	1 × 10/25 Гбит/с
Мастер-серверы (FE)	3	8 CPU	16 ГБ	200 ГБ SSD	1 × 25 Гбит/с
Вычислительные серверы (BE/CN)	3	8 CPU, AVX2 обязателен	64 ГБ	500 ГБ SSD	1 × 25 Гбит/с

2.2 Поддерживаемые операционные системы

Подробная информация приведена в главе Подготовка хостов.

3 Подготовка хостов

В главе приведены требования к хосту (хостам), которые должны быть выполнены до запуска `install.sh`. Инсталлятор намеренно не устанавливает эти компоненты (не изменяет пакеты и сетевую политику хоста) за исключением развёртывания бандлов.

3.1 Чек-лист подготовки хостов

Таблица 3.1. Чек-лист пререквизитов на хостах (по сценариям установки)

Сценарий	Требуется на хосте
Инфра + control-plane (online, без Engine)	Базовые Java 17 Python3 DNS/Nexus-креды Firewallld
Инфра + control-plane (offline, без Engine)	Базовые Java 17 Python3 bundleDir с артефактами Firewallld
+ Engine (StarRocks)	Дополнительно: ansible-core + sshpass galaxy: ansible.posix+community.general SSH-доступ к нодам LVM rootlv расширен license.jwt
BE/CN-ноды	CPU с AVX2 RHEL ≥9.8 / Ubuntu ≥22.04 NTP < 5с FE-ноды — Java 17
FE-ноды	Любая ОС с Java 17
Вынесенный Engine (multi-node)	engine.queryHost = FE leader IP

Engine-ноды настраиваются автоматически через `bootstrap-host.yaml` (пакеты, `sysctl`, лимиты, пользователь `selen`, порты). См. раздел 3.2.5 «Пререквизиты для Engine».

3.2 Описание пререквизитов

Примеры команд приведены для RHEL 9 (`dnf`), в Ubuntu замените на `apt`.

Базовые утилиты

Базовые утилиты: `bash`, `awk`, `curl`, `tar`, `openssl`, `sha256sum`, `systemctl`, — есть в RHEL и Ubuntu по умолчанию.

`yq` не требуется. `deploy.yaml` и `artifacts.yaml` парсятся встроенным `awk`-парсером (`scripts/lib/common.sh` → `yaml_get`). Сторонние исполняемые файлы не понадобятся.

Java 17

Java ≥ 17 (control-plane: CM + Keycloak)

```
sudo dnf install -y java-17-openjdk-headless
```

Python 3

Для работы скриптов инсталлятора (keycloak-realm.sh, engine.sh) на хосте должен быть установлен интерпретатор Python 3.

На большинстве дистрибутивов (Ubuntu, RHEL) Python 3 присутствует в базовой поставке, однако на minimal-образах он может отсутствовать. Убедитесь, что Python 3 установлен до запуска инсталлятора.

Инсталлятор проверяет наличие Python 3 на этапе preflight. При отсутствии — установка будет прервана с соответствующим сообщением.

ПРИМЕЧАНИЕ

Alembic-миграции (обновление схемы БД) используют встроенный интерпретатор Python из бандла (`\${OPT_ROOT}/ide/runtime/bin/python3`) и не требуют наличия системного Python на хосте.

Доступ к Nexus

Доступ к Nexus (для online-режима):

```
# DNS: hub.selena.sh не резолвится встроенным DNS тест-VM
echo "194.190.8.222 hub.selena.sh" | sudo tee -a /etc/hosts

# креды реестра (передаются install.sh как env)
export NX_USER='<логин>'
export NX_PASS='<пароль>'
```

В offline-режиме Nexus не используется — артефакты берутся из bundleDir.

Пререквизиты для Engine

ПРИМЕЧАНИЕ

Раздел актуален для сценариев, предусматривающих развертывание Engine (StarRocks) через Cluster Manager. Если Engine не разворачивается — этот раздел возможно пропустить.

Cluster Manager не выполняет установку StarRocks напрямую. Он запускает Ansible на хосте CM, который через SSH настраивает FE/BE-ноды.

ОС Engine-нод (для BE/CN)

ОС для BE/CN:

- Подходят: RHEL ≥ 9.8 / Ubuntu ≥ 22.04 / Astra 1.8 или иной EL9 с GLIBC_2.35
- Не подходят: RHEL 9.6, RHEL 8, Astra 1.7 (версия glibc ниже требуемой)

BE и CN — компоненты C++ (selena_be), требуют определённой версии системной библиотеки glibc. FE — Java-приложение, для него это не имеет значения.

Бандлы BE/CN собраны под glibc 2.35 (Ubuntu 22.04) и требуют символ `_dl_find_object@GLIBC_2.35`.

Таблица 3.2. Совместимость ОС с BE/CN

ОС BE-ноды	glibc	<code>_dl_find_object@2.35</code>	Статус запуска BE
RHEL 9.6	2.34	Нет	Ошибка: version 'GLIBC_2.35' not found → BE exit 1 → install-be.yaml timeout 9050
RHEL 9.8+	2.34	Есть (бэкпортирован Red Hat)	Запущен
Ubuntu 22.04 / 24.04	2.35 / 2.39	Есть	Запущен
Rocky Linux 9	2.34	Зависит от минорной версии	Аналогично RHEL 9
RHEL 8 / CentOS 8	2.28	Нет	Ошибка

WARNING

`ldd --version` на RHEL 9.6 и 9.8 показывает glibc 2.34 в обоих случаях. Однако Red Hat бэкпортирует некоторые символы из новых glibc в свежие минорные версии. Проверка совместимости без запуска BE:

```
objdump -T /opt/selena/be/lib/selena_be | grep GLIBC_2.35 # какие
2.35-символы нужны
objdump -T /lib64/libc.so.6 | grep _dl_find_object # есть ли он в
системном libc
```

CPU: AVX2 (для BE/CN)

StarRocks BE использует векторизацию на AVX2 — без него selena_be не стартует. Проверка:

```
grep -o avx2 /proc/cpuinfo | head -1 # должно вывести avx2
```

Требуются процессоры ($\approx$ 2013 года и новее). В виртуальных средах AVX2 может быть скрыт от гостевой ОС, если не настроен проброс флагов CPU (passthrough). Это аппаратное требование, не подлежит программной настройке.

Синхронизация времени (для multi-node)

StarRocks требует расхождение часов между FE/BE-нодами менее 5 секунд. При превышении этого порога FE не может выбрать LEADER/FOLLOWER (в fe.log регистрируется ошибка: «time deviation between FE nodes is greater than 5s»). bootstrap-host.yaml не выполняет эту настройку — её необходимо обеспечить заранее на каждой ноде:

```
sudo dnf install -y chrony && sudo systemctl enable --now chronyd # RHEL
# Ubuntu/Astra: systemd-timesyncd обычно уже активен (timedatectl)
timedatectl # проверить: System clock synchronized: yes
```

Настройка Engine-нод

Настройки, выполняемые через bootstrap-host.yaml (автоматически):

- Установка пакетов (`openjdk-17-headless/jre`, `mariadb`, `mariadb-client`, `tar`, `gzip`, `curl`, `firewalld`).
- Применение параметров ядра (`vm.max_map_count=262144`).
- Настройка лимитов ресурсов для пользователя `selen` (`nofile` и `nproc = 655350`).
- Создание служебного пользователя и группы `selen` (UID/GID 10001).
- Открытие портов в `firewalld`.

ПРИМЕЧАНИЕ

JDK 17 требуется также на BE-нодах, поскольку `selen_be` использует JNI для работы с `external catalogs` (линковка с `libjvm.so`). StarRocks официально не рекомендует использование JRE вместо JDK. Однако в Ubuntu через `apt` устанавливается `openjdk-17-jre-headless`, чего достаточно для выполнения рантайма, но может быть недостаточно для компиляции UDF-функций.

Предварительные требования (обеспечиваются администратором до установки):

- glibc-совместимая ОС (см. раздел 3.2.5.1)
- Поддержка AVX2 (раздел 3.2.5.1.1)
- Синхронизация времени через NTP (раздел 3.2.5.1.2)
- SSH-доступ от CM к Engine-нодам (раздел 3.2.5.4)
- Достаточное место в корневом разделе (раздел 3.2.5.5).

Ansible + sshpass

Для управления Engine-нодами через Ansible установите на хосте CM пакеты `ansible-core` и `sshpass`:

```
sudo dnf install -y ansible-core sshpass
```

Galaxy-коллекции

Плейбуки CM (`bootstrap-host.yaml` и др.) используют модули из `ansible.posix` (например, `ansible.posix.sysctl` для `vm.max_map_count=262144`) и `community.general`.

```
sudo ansible-galaxy collection install ansible.posix community.general  
-p /usr/share/ansible/collections
```

⚠ Флаг -p обязателен.

СМ запускает Ansible от имени пользователя selena. Без флага коллекции устанавливаются в каталог пользователя, выполнившего команду (~/.ansible/collections), и не видны selena. Флаг -p /usr/share/ansible/collections помещает коллекции в системный каталог, доступный для всех пользователей, что обеспечивает их видимость для СМ. Без этого возникает ошибка: couldn't resolve module/action 'ansible.posix.sysctl'.

Чтобы убедиться, что коллекции видны из контекста СМ, выполните:

```
sudo -u selena ansible-galaxy collection list 2>/dev/null | grep -E  
'posix|general'  
# фактические пути поиска коллекций под selena:  
sudo -u selena ansible-config dump 2>/dev/null | grep -i COLLECTIONS_PATH
```

Если путь переопределён (свой ansible.cfg рядом с плейбуками в /run/selena/cm/ansible/<id>/ или env) и /usr/share/ansible/collections не подхватывается, укажите его принудительно для сервиса СМ:

```
echo 'ANSIBLE_COLLECTIONS_PATH=/usr/share/ansible/collections' | sudo tee -  
a /etc/selena/cm.env  
sudo systemctl restart selena-cm
```

SSH-доступ к FE/BE-нодам

Ansible с хоста СМ подключается к Engine-нодам по SSH, используя учётную запись и ключ, указанные при регистрации ноды в СМ. На каждой Engine-ноде должен присутствовать этот пользователь с настроенной аутентификацией по SSH-ключу и правом выполнения sudo без пароля (NOPASSWD).

При установке в режиме All-in-One инсталлятор автоматически создаёт локального пользователя selena-ansible с необходимыми настройками. Для отдельных нод (multi-node) учётную запись и ключ необходимо подготовить на каждой FE/BE-ноде вручную.

Root LVM

СМ распаковывает StarRocks (engine-fe.tar.gz ≈1.2 ГБ, engine-be ≈1.4 ГБ) в /opt/selena, который располагается в корневом разделе. Многие облачные и enterprise-образы RHEL

используют LVM с rootlv ≈2 ГБ и распаковка завершается с ошибкой: No space left on device (gtar: ... Cannot open: No space left).

OS-диск обычно 64 ГБ, но большая часть отдан в rootlv. Требуется расширить корневой раздел на каждой Engine-ноде:

```
sudo lvextend -l +100%FREE /dev/rootvg/rootlv # забрать весь свободный
объём VG в root
sudo xfs_growfs / # растянуть ФС (для xfs; для ext4 - resize2fs /)
df -h / # проверка: должно стать ~40 ГБ
```

Альтернатива: заранее монтировать отдельный диск в `/opt/selena`. Без этого `install-fe.yaml` и `install-be.yaml` завершаются ошибкой на шаге распаковки архива.

License

Engine требует активную лицензию. Положить файл лицензии:

```
sudo install -d -m0755 /run/selena-input
sudo cp <license.jwt> /run/selena-input/license.jwt
```

Без `/run/selena-input/license.jwt` engine-фаза пропускается с выдачей предупреждения, инфраструктура и control-plane при этом загружаются без ошибок.

Вынесенный Engine (multi-node)

Если Engine развёрнут на отдельных нодах (FE/BE не на хосте control-plane), IDE и MCP по умолчанию пытаются подключиться к StarRocks по адресу 127.0.0.1:9030. При отсутствии FE на локальном хосте возникает ошибка: `(2002, "Can't connect to MySQL server on '127.0.0.1'")`.

```
engine:
  queryHost: 10.0.0.5 # IP FE-leader; all-in-one (FE на том же хосте) ->
  127.0.0.1 (дефолт)
```

Инсталлятор передаёт указанный адрес в переменные `IDE_SELENA_HOST` и `MCP_SELENA_HOST` через `detect_engine_query_host`.

Для сценария «Инфра + control-plane» (online.deploy.yaml, без Engine) этап engine_bootstrap не выполняется, так как Engine устанавливается отдельно через CM. Сборка кластера производится вручную в CM UI/API.

Firewall (доступ к UI извне)

На RHEL-хостах по умолчанию активен firewalld. Открытие портов в облачной Security Group (NSG) недостаточно — необходимо также разрешить доступ на уровне хоста.

Выполните команды:

```
sudo firewall-cmd --permanent --add-port=8080/tcp --add-port=8088/tcp
--add-port=10001/tcp --add-port=8412/tcp --add-port=3000/tcp
sudo firewall-cmd --reload
```

Порты: 8080 (Keycloak), 8088 (CM), 10001 (IDE), 8412 (AI), 3000 (Grafana).

3.3 Preflight — проверка хоста

Вам необходимо самостоятельно установить и настроить компоненты, перечисленные в разделах выше.

Preflight-проверки на этапе установки:

- Прерывание установки:
 - Неподходящая архитектура хоста (только amd64).
 - Отсутствуют базовые утилиты: `systemctl`, `openssl`, `curl`, `tar`, `sha256sum`, `awk`.
 - Порты заняты посторонними процессами.
 - Недоступен Nexus.
 - Не заданы переменные `NX_USER` и `NX_PASS`.
- Вывод предупреждения:
 - Недостаточны ресурсы хоста:
CPU < 4 ядер, RAM < 8 ГБ, свободное место на `/opt` < 10 ГБ.
 - Выключены сервисы: `postgres`, `keycloak`, `minio`.

 WARNING

Preflight проверяет ресурсы по порогу 4 vCPU / 8 ГБ, что ниже минимальных требований для All-in-One установки (8 vCPU / 32 ГБ). Прохождение preflight не гарантирует соответствие оборудования требованиям продукта.

Для offline-установки: Preflight проверяет наличие Ansible, Ansible Galaxy, sshpass и Galaxy-коллекций — при отсутствии установка прерывается.

Для online-установки: Ansible, Ansible Galaxy, sshpass и Galaxy-коллекции не проверяются. Их отсутствие будет обнаружено только на этапе установки Engine.

4 Установка Selena (online)

Раздел описывает установку Selena нативно (systemd, без контейнеров) в онлайн-режиме.

4.1 Доступ к реестру (online) + license

- DNS: hub.selena.sh может не резолвиться на тест-VM.

```
echo "194.190.8.222 hub.selena.sh" | sudo tee -a /etc/hosts
```

- Креды Nexus: передаются install.sh как env.

```
export NX_USER='<логин>'
export NX_PASS='<пароль>'
```

- license: нужна только если ставится Engine, без неё Engine-фаза пропускается.

```
sudo install -d -m0755 /run/selena-input
sudo cp <license>.jwt /run/selena-input/license.jwt
```

- LLM-ключ:  обязателен, если вы используете AI-чат.

Без него сервис selena-ai-backend не запустится (юнит уйдёт в цикл перезапусков).

Если AI-чат не нужен - отключите его в конфигурации: `components.ai: false`.

- Основной путь (рекомендуемый):

```
printf '%s' '<LLM_API_KEY>' | sudo tee
/var/lib/selena/runtime/secrets/llm-api-key >/dev/null
sudo chmod 0600 /var/lib/selena/runtime/secrets/llm-api-key
```

- Альтернативный путь (инсталлятор читает его первым и сам копирует значение в runtime/secrets):

```
printf '%s' '<LLM_API_KEY>' | sudo tee /run/selena-input/llm-api-key >/dev/null  
sudo chmod 0600 /run/selena-input/llm-api-key
```

- Назначение каталога `/run/selena-input`: входной каталог оператора для `license.jwt` и `llm-api-key`.
Инсталлятор читает файлы и переносит значения: лицензию - в CM, LLM-ключ - в `/var/lib/selena/runtime/secrets/llm-api-key`.
Ничего постоянно в нём не хранится, файлы после установки можно удалить.
- При смене ключа необходимо повторно запустить `install.sh/upgrade.sh` и перезапустить сервис: `systemctl restart selena-ai-backend`.

4.2 Настройка `deploy.yaml`

Скопируйте пример и настройте под свою среду (главное - `publicHost`):

```
cp examples/all-in-one.deploy.yaml my.deploy.yaml
```

или

```
cp examples/cm.deploy.yaml my.deploy.yaml
```

Основные параметры для изменения:

```
host:  
publicHost: <IP-или-DNS> # адрес для redirect URI Keycloak и ссылок UI  
(localhost -> снаружи не зайти)  
<br/>components: # выключить ненужное (true/false); CM и Engine не тоглятся  
ai: false # напр. поднять без AI-сервисов  
<br/>engine:  
queryHost: 127.0.0.1 # all-in-one. Если FE-нода отдельная -> её IP (напр.  
10.0.0.5)
```

4.3 Установка

Перейдите в директорию инсталлятора:

```
cd deploy/baremetal
```

Запустите установку:

```
# All-in-one
```

```
sudo NX_USER='<логин>' NX_PASS='<пароль>' bash scripts/install.sh --deploy  
my.deploy.yaml --artifacts examples/artifacts.yaml
```

```
# Инфра + control-plane (без Engine)
```

```
sudo NX_USER='<логин>' NX_PASS='<пароль>' bash scripts/install.sh --deploy  
my.deploy.yaml --artifacts examples/artifacts-cm.yaml
```

ПРИМЕЧАНИЕ

Для передачи переменных используется форма `sudo NX_USER=... NX_PASS=...`, которая работает при любых настройках `sudoers`. Альтернативно можно использовать `sudo -E` с предварительным экспортом переменных, но это требует наличия флага `SETENV` в политике `sudoers`.

Установка выполняется поэтапно:

preflight → java → secrets → fetch (скачать+sha256+распаковать) → рендер юнитов → PostgreSQL → db_migrate_apps (alembic-миграции IDE и AI-backend) → Keycloak realm → инфра → control-plane → Engine (CM через ansible).

По завершении инсталлятор создаёт файл `/etc/selena/all-in-one-credentials.txt` (или аналог для cm).

Долгий прогон удобно запускать под `setsid`, чтобы не оборвался при разрыве SSH:

```
sudo NX_USER='<логин>' NX_PASS='<пароль>' setsid bash scripts/install.sh  
... &> /var/log/selena-install.log
```

4.4 Проверка

```
sudo bash scripts/status.sh # таблица юнитов + health-коды (ждём 200)
systemctl status 'selena-*' # состояние всех юнитов
sudo cat /etc/selena/all-in-one-credentials.txt
```

Доступы (логин в CM/IDE/AI - через SSO realm selena, юзер admin, пароль = Keycloak-admin):

Сервис	URL
Cluster Manager	http://<publicHost>:8088
IDE (Querybook)	http://<publicHost>:10001
AI	http://<publicHost>:8412
Keycloak admin	http://<publicHost>:8080
Grafana	http://<publicHost>:3000

4.5 После сборки Engine-кластера - перезапуск IDE (Querybook)

All-in-one: автоматический перезапуск сервисов IDE и MCP выполняется инсталлятором по завершении установки. Дополнительных действий не требуется.

Инфра + control-plane (вынесенный Engine): после сборки кластера Selena и запуска FE/BE-компонентов необходимо перезапустить три компонента Querybook, чтобы IDE получила актуальные параметры подключения к StarRocks, включая impersonator-пароли и адрес активного FE:

```
sudo systemctl restart selena-ide-web.service selena-ide-worker.service
selena-ide-scheduler.service selena-mcp.service
```

ПРИМЕЧАНИЕ

Если этого не сделать, IDE может сохранить неактуальное или пустое подключение к движку, и запросы будут направляться на FE, который на момент запуска IDE ещё не существовал.

После перезапуска проверьте, что все три компонента активны, а веб-интерфейс возвращает статус health:

```
systemctl is-active selena-ide-web selena-ide-worker selena-ide-scheduler  
selena-mcp  
curl -s -o /dev/null -w '%{http_code}\\n' <http://127.0.0.1:10001/health> #  
ожидайте 200
```

4.6 Обновление версий и конфигурации (upgrade.sh)

Стенд приводится к yaml без полной reinstall - рестартуют только изменившиеся сервисы. Указывайте оба флага явно (дефолты используют examples/ репозитория, а не файлы, с которыми ставился стенд):

```
# 1. поменяйте версию в artifacts.yaml и/или конфиг в deploy.yaml  
# 2. посмотрите план (стенд не изменяется; в online-режиме NX_USER/NX_PASS  
обязательны  
# и для --dry-run - их требует загрузка конфигурации):  
sudo NX_USER=... NX_PASS=... ./scripts/upgrade.sh --deploy deploy.yaml --  
artifacts artifacts.yaml --dry-run  
# 3. примените:  
sudo NX_USER=... NX_PASS=... ./scripts/upgrade.sh --deploy deploy.yaml --  
artifacts artifacts.yaml  
  
# после sudo (как в примерах выше):  
sudo NX_USER=... NX_PASS=... ./scripts/upgrade.sh ...  
# или из env-файла с export-строками:  
sudo bash -c 'set -a; . /path/to/nx.env; set +a; ./scripts/upgrade.sh ...'  
# или (если sudoers разрешает): . nx.env && sudo NX_USER=... NX_PASS=...  
./scripts/upgrade.sh ...
```

5 Установка Selena (offline)

Установка выполняется на изолированном хосте (airgapped) без доступа к внешним сетям и без внутреннего реестра артефактов. Все необходимые дистрибутивы поставляются единым бандлом, из которого инсталлятор выполняет развёртывание.

5.1 Бандл артефактов + инсталлятор

Бандл артефактов (bundle-pack) и инсталлятор предоставляются вендором по запросу на info@selena-lakehouse.ru.

Комплект поставки:

- Архив `selena-offline-bundle-<date>.tar.gz`:
 - Артефакты инфраструктуры и control-plane.
 - Артефакты Engine - FE/BE/CN.
 - Файл контрольной суммы (.sha256).
 - Манифест (artifacts-offline.yaml) с описанием артефактов в пакете.
- Инсталлятор:

Файл	Назначение
<code>scripts/install.sh</code>	Установка и переустановка поверх существующей; <code>--deploy</code> , <code>--artifacts</code> , <code>--render-only</code> .
<code>scripts/upgrade.sh</code>	Обновление версий и конфигурации без полной переустановки; <code>--dry-run</code> , <code>--yes</code> .
<code>scripts/uninstall.sh</code>	Остановка и удаление юнитов; <code>--delete-data</code> - вместе с данными и секретами.
<code>scripts/preflight.sh</code>	Проверка хоста до установки; вызывается из <code>install.sh</code> , запускается и отдельно.
<code>scripts/status.sh</code>	Таблица состояния юнитов с HTTP-кодами health-эндпоинтов.
<code>scripts/logs.sh</code>	Обёртка над <code>journalctl</code> для юнита компонента.

5.2 Доставка и распаковка

На airgapped-хосте выполните:

```
sudo mkdir -p /opt/selena-bundle
sudo tar xzf selena-offline-bundle-<date>.tar.gz -C /opt/selena-bundle --
strip-components=1
# итор: /opt/selena-bundle/artifacts/<comp>/<ver>/<comp>-linux-amd64.tar.gz
(+ .sha256)
```

5.3 License + LLM-ключ

License

Скопируйте файл лицензии в `/run/selena-input`:

```
sudo install -d -m0755 /run/selena-input
sudo cp <license>.jwt /run/selena-input/license.jwt
sudo chmod 0600 /run/selena-input/license.jwt
```

LLM-ключ

⚠ Ключ обязателен, если вы используете AI-чат.

Без него сервис `selena-ai-backend` не запустится (юнит уйдёт в цикл перезапусков).

Если AI-чат не нужен - отключите его в конфигурации: `components.ai: false`.

- Основной путь (рекомендуемый):

```
printf '%s' '<LLM_API_KEY>' | sudo tee
/var/lib/selena/runtime/secrets/llm-api-key >/dev/null
sudo chmod 0600 /var/lib/selena/runtime/secrets/llm-api-key
```

- Альтернативный путь (инсталлятор читает его первым и сам копирует значение в `runtime/secrets`):

```
printf '%s' '<LLM_API_KEY>' | sudo tee /run/selena-input/llm-api-key
>/dev/null
sudo chmod 0600 /run/selena-input/llm-api-key
```

Назначение каталога `/run/selena-input`: входной каталог оператора для `license.jwt` и `llm-api-key`. Инсталлятор читает файлы и переносит значения: лицензию - в СМ, LLM-ключ - в `/var/lib/selena/runtime/secrets/llm-api-key`. Ничего постоянного там не хранится, файлы после установки можно удалить.

При смене ключа необходимо повторно запустить `install.sh/upgrade.sh` и перезапустить сервис: `systemctl restart selena-ai-backend`.

5.4 Установка

1. Перейдите в директорию инсталлятора:

```
cd deploy/baremetal
```

2. Скопируйте пример конфигурации (`examples/offline.deploy.yaml` или `examples/cm.deploy.yaml`) в `my.deploy.yaml` и укажите в нём `publicHost` (внешний IP или домен вашего хоста).

Измените прочие настройки по аналогии с разделом 4.2 «Настройка `deploy.yaml`».

3. Запустите установку:

```
sudo -E bash scripts/install.sh --deploy my.deploy.yaml --artifacts  
examples/artifacts-offline.yaml
```

В offline-режиме переменные `NX_USER` и `NX_PASS` не требуются. Инсталлятор копирует бандлы из каталога `bundleDir`, помещает архивы StarRocks в `/opt/selena/_dl/` и устанавливает для Cluster Manager режим `registry.mode=OFFLINE`. В этом режиме CM распаковывает дистрибутивы локально, без обращения к внешнему реестру.

5.5 UI

Сервисы доступны по адресам:

```
http://<publicHost>:8088 (CM), :10001 (IDE), :8412 (AI), :3000 (Grafana).
```

5.6 Обновление дистрибутивов (engine upgrade)

Имена в `/opt/selena/_dl/` версионно-нейтральные (обновление = замена файла):

```
sudo cp engine-fe-<new>.tar.gz /opt/selena/_dl/engine-fe.tar.gz  
# затем штатный engine-upgrade в CM UI
```

5.7 Обновление версий и конфигурации (upgrade.sh)

Для обновления в изолированной среде:

1. Поместите новые бандлы в локальный каталог bundleDir, указанный в artifacts-offline.yaml. Старые версии из bundleDir не удаляйте, если планируется возможный откат.
2. Запустите upgrade.sh с теми же конфигурационными файлами, что использовались при установке. В режиме registry.mode: offline флаги `--dry-run` и `--yes` работают стандартно.

```
sudo bash scripts/upgrade.sh --deploy my.deploy.yaml \  
--artifacts examples/artifacts-offline.yaml --dry-run  
sudo bash scripts/upgrade.sh --deploy my.deploy.yaml \  
--artifacts examples/artifacts-offline.yaml -yes
```

`NX_USER` и `NX_PASS` не требуются — скачивание идёт из локального bundleDir.

6 Управление и обслуживание

Глава актуальна для обоих режимов установки, онлайн и офлайн.

6.1 Управление стеком

```
sudo systemctl start selena.target # поднять всё
sudo systemctl stop selena.target # остановить всё
sudo systemctl restart selena-cm.service # перезапустить один компонент
journalctl -u selena-cm.service -f # логи (или scripts/logs.sh)
```

Стек включён в multi-user.target - запускается сам после ребута.

6.2 Дополнительные переменные окружения (extraEnv)

Любому control-plane сервису возможно передать произвольные переменные окружения через deploy.yaml:

```
extraEnv:
cm:
vars: # инлайн-пары -> рендерятся в
/var/lib/selena/runtime/secrets/cm.extra.env
JAVA_OPTS_APPEND: "-Xmx2g"
files: # готовые env-файлы оператора (абсолютные пути, формат KEY=VALUE)
- /etc/selena/custom/cm-tuning.env
```

Переменные окружения могут быть заданы для следующих сервисов:

postgres, keycloak, minio, lakekeeper, grafana, cm, ide, ai-backend, ai-frontend, mcp.

Для ide окружение является общим для трёх компонентов: web, worker, scheduler.

Настройка Engine выполняется через Cluster Manager, а не через extraEnv.

Порядок применения файлов окружения (приоритет у последнего):

`<svc>.env` → `<svc>.extra.env` → files (по порядку) → impersonator-файлы CM (только для ide/mcp). Impersonator имеет наивысший приоритет и не переопределяется через extraEnv.

extraEnv позволяет добавлять и переопределять переменные (например, KC_HOSTNAME) на усмотрение администратора.

- Значения пишутся как есть (KEY=VALUE, без кавычек в итоговом файле).
- Файлы из files инсталлятор не копирует - только валидирует (проверяет наличие и абсолютный путь) и подключает юниту через EnvironmentFile=.
Права доступа к ним обеспечивает оператор (для секретов рекомендуется 0600).
- extra.env регенерируется каждым запуском install.sh и deploy.yaml:
переменные задаются в конфиге, ручные правки теряются при переустановке.
- Для применения изменений выполните:

```
sudo ./scripts/install.sh --deploy ... && sudo systemctl restart <юнит>.
```

6.3 Удаление

Для удаления компонентов Selena используются команды:

Команда	Действие
<code>sudo bash scripts/uninstall.sh</code>	Остановка и удаление systemd-юнитов
<code>sudo bash scripts/uninstall.sh --delete-data</code>	Остановка и удаление systemd-юнитов + удаление /opt/selena, /var/lib/selena, /etc/selena.
<code>sudo bash scripts/uninstall.sh --delete-data && sudo userdel selena</code>	Чистая переустановка, удаление пользователя selena, иначе при повторной установке будет конфликт UID.

Примечания:

- uninstall.sh без флага `--delete-data` выполняет удаление каталога `/opt/selena`. В данном каталоге размещаются метаданные FE (`/opt/selena/fe/meta`), сторедж BE (`/opt/selena/be/storage`), логи движка и `_dl`. На All-in-One- установке это приводит к уничтожению метаданных StarRocks вместе с данными, восстановление кластера невозможно.
- После выполнения uninstall.sh в системе остаются:
 - Пользователь selena-ansible
 - Файл `/etc/sudoers.d/90-selena-all-in-one-ansible`
 - Каталог `/var/lib/selena-engine`
 - systemd-юниты Engine на Engine-нодах, созданные плейбуками CM.
- uninstall.sh не обрабатывает конфигурационный файл deploy.yaml. Пути для удаления определяются переменной окружения DATA_ROOT (по умолчанию `/var/lib/selena`).

Если в `deploy.yaml` изменены параметры `roots.*`, использование `--delete-data` на All-in-One установке приведёт к удалению не тех каталогов, а фактические каталоги с данными останутся нетронутыми.

6.4 Особенности работы `upgrade.sh`

- Версии: сравнивается маркер `.artifact` в каталоге компонента с `artifacts.yaml`; новый бандл скачивается и проверяется (sha256) до остановки сервиса.
- Конфигурация: изменившиеся `env`-файлы/юниты вычисляются рендером (`install.sh --render-only`) - рестартует только затронутое, для `ide` все три юнита. Смена версии тоже видна как изменение (конфиг `[<comp>]`) - рендер переписывает файлы компонента, `change-set` дедуплицируется, рестарт один.
- Старая инсталляция без `.artifact`-маркеров (поставлена `install.sh` до появления `upgrade.sh`):
текущие версии неизвестны → план покажет `<unknown>` -> `<версия>` для всех компонентов и, после подтверждения, будут загружены и заменены все бандлы, данные в `<dataRoot>` не затрагиваются.

Чтобы получить корректный минимальный дифф, установите маркеры фактических версий перед первым запуском:

```
printf '%s' '<path-из-artifacts.yaml>' | sudo tee  
/opt/selena/<comp>/.artifact.
```

⚠ При пустом маркере `major-guard postgres` не срабатывает (текущий `major` неизвестен) - сверьте версию `postgres` в `artifacts.yaml` с установленной вручную.

- Отката нет: укажите старую версию в `yaml` и снова запустите `upgrade.sh` (для `offline`: старый бандл должен лежать в `bundleDir`).
- Смена `major`-версии PostgreSQL требует `--yes` и не переносит данные (`pg_upgrade` вне скоупа). Engine обновляется через CM, а не с помощью `upgrade.sh`.
- Ошибка на компоненте останавливает процесс, уже обновлённые компоненты не откатываются. Оставшиеся компоненты фиксируются в `pending`-файле (`<runtimeRoot>/upgrade-pending`), повторный запуск подхватывает их как `recovery` [...] и продолжает с места падения (даже если конфиг-дифф уже пуст).
- При отказе от подтверждения (N), прерывании (Ctrl+C), обрыве ввода (закрытие терминала, разрыв SSH, перенаправление `< /dev/null` - трактуется как отказ) или сбое

рендера подготовленные изменения откатываются (restore и daemon-reload). Стенд возвращается к исходному состоянию.

7 Внешние PostgreSQL и Keycloak (consume-only)

Глава описывает подключение baremetal-инсталляции к уже существующим PostgreSQL и/или Keycloak.

Модель: оператор самостоятельно готовит БД/роли и realm/клиентов. Инсталлятор их не создаёт, а инсталляция получает настройки через extraEnv в deploy.yaml.

Готовые примеры: `examples/external-services.deploy.yaml`, `examples/external-secrets.env`, `examples/external-keycloak-realm.json`.

Как работает: `extraEnv.<svc>.vars` рендерится в `<svc>.extra.env` при каждом install/upgrade и подключается юнитом после основного env; файлы из `extraEnv.<svc>.files` подключаются после `extra.env` (последний имеет приоритет).

Поэтому переопределения декларативны, сохраняются при обновлениях, а секреты хранятся в отдельном файле, который инсталлятор не трогает.

Поддерживаемые сценарии: только внешний PG, только внешний Keycloak, оба (пример ниже).

7.1 Подготовка внешнего PostgreSQL

Locale - обязательное требование (lakekeeper требует ICU-collation):

либо кластер инициализирован с `--encoding=UTF8 --locale=C.utf8` (как в initdb бандла),

либо каждая БД создаётся с явными `ENCODING 'UTF8' LC_COLLATE 'C.utf8' LC_CTYPE 'C.utf8' TEMPLATE template0`.

Выполнить под суперьюзером:

```
CREATE ROLE selena_cm LOGIN PASSWORD 'CHANGE_ME';
CREATE ROLE selena_ide LOGIN PASSWORD 'CHANGE_ME'; -- если
components.ide=true
CREATE ROLE selena_ai LOGIN PASSWORD 'CHANGE_ME'; -- если
components.ai=true
CREATE ROLE lakekeeper LOGIN PASSWORD 'CHANGE_ME'; -- если
components.lakekeeper=true
CREATE ROLE keycloak LOGIN PASSWORD 'CHANGE_ME'; -- ТОЛЬКО если Keycloak
остаётся бандленным
-- Если кластер УЖЕ инициализирован с --locale=C.utf8 - обычный CREATE
DATABASE ... OWNER
-- для остальных БД достаточен. Если локаль кластера иная - локаль нужно
указать явно
-- КАЖДОЙ БД (см. lakekeeper ниже), иначе lakekeeper упадёт на ICU-
collation.
CREATE DATABASE selena_cm OWNER selena_cm;
CREATE DATABASE selena_ide OWNER selena_ide;
CREATE DATABASE selena_ai OWNER selena_ai;
CREATE DATABASE lakekeeper OWNER lakekeeper
ENCODING 'UTF8' LC_COLLATE 'C.utf8' LC_CTYPE 'C.utf8' TEMPLATE template0;
CREATE DATABASE keycloak OWNER keycloak; -- только при бандленном Keycloak
```

Расширения: lakekeeper при первой миграции выполняет `CREATE EXTENSION "uuid-ossf"`. Файлы расширения должны быть установлены на сервере PG (пакет `postgresql16-contrib` или аналог). В PG бандла он уже есть, при отсутствии расширения запуск lakekeeper завершится с ошибкой: «extension "uuid-ossf" is not available».

Для ролей, используемых при установке, необходимо настроить сетевой доступ (`pg_hba`) с хоста, где выполняется инсталляция. Создавать схемы не требуется: инсталлятор самостоятельно выполнит миграции IDE и AI через alembic; а CM, Keycloak и lakekeeper применят миграции при первом запуске.

7.2 Подготовка внешнего Keycloak

Вариант А (одной операцией): импортировать `examples/external-keycloak-realm.json`, предварительно заменив `CHANGE_ME_*`-секреты и 203.0.113.10 на ваш publicHost.

Шаблон не тестировался на работающем внешнем Keycloak (импорт service-account-юзеров version-dependent). После импорта проверьте чек-лист:

- У service-account клиента `selen-cluster-manager-api` назначены 4 клиентские роли в realm-management: query-groups, query-users, view-realm, view-users.
- Bootstrap-юзер добавлен в группы `/cm/admins` и `/selena/writers`.
- Группа `cm/security-admins` существует.

Вариант Б (вручную) - завести в realm:

Клиент	Тип	Обязательное
selen-engine	confidential	directAccessGrants; audience-маппер selen-engine; groups-маппер groups (full path); redirect URIs <code>http://<publicHost>:8088/*</code> и <code>http://<publicHost>:10001/*</code>
selen-cluster-manager-api	service account	клиентские роли realm-management: query-groups, query-users, view-realm, view-users
selen-ide	confidential	audience-маппер selen-engine; redirect <code>http://<publicHost>:10001/*</code>
selen-ide-service	service account	audience-маппер selen-engine
selen-ai	confidential	redirect <code>http://<publicHost>:8411/*, :8412/*</code>
selen-ai-service	service account	audience-маппер selen-engine

Браузерным клиентам (selen-engine, selen-ide, selen-ai) необходимо также заполнить webOrigins: в realm бандла - *; в проде - по вашей CORS-политике.

Группы: `cm/{admins, role-admins, security-admins, viewers}`, `selen/{readers, writers}`. Права пользователей Selena определяются членством в этих группах (claim groups, full path). Для автоматического Engine bootstrap требуется УЗ в `/cm/admins` с разрешённым password grant (см. раздел 7.5).

7.3 Deploy.yaml

В качестве основы используется файл `examples/external-services.deploy.yaml`. В нём необходимо отключить установку встроенных PostgreSQL и Keycloak:

```
components:
  postgres: false # не ставить бандленный PG
  keycloak: false # не ставить бандленный KC
```

Параметры подключения, не содержащие секретов (URL, имена БД), передаются через `extraEnv.<svc>.vars` в `deploy.yaml`.

Пароли и токены размещаются в файле `/etc/selena/external/secrets.env`, который подключается к сервисам через `extraEnv.<svc>.files`.

Переменные для сервисов:

Таблица 7.1. PostgreSQL

Сервис	vars (несекретно)	secrets.env
Keycloak (если бандленный)	KC_DB_URL	KC_DB_USERNAME, KC_DB_PASSWORD
cm	CM_DB_URL, CM_DB_USER	CM_DB_PASSWORD
ide	-	DATABASE_CONN (conn-string целиком)
ai-backend	-	AI_BACKEND_DATABASE_URL
lakekeeper	-	LAKEKEEPER_PG_DATABASE_URL_READ, LAKEKEEPER_PG_DATABASE_URL_WRITE

Таблица 7.2. Keycloak (база = `https://<kc-host>/realms/<realm>`)

Сервис	vars (несекретно)	secrets.env
cm	CM_KEYCLOAK_TOKEN_URI, CM_KEYCLOAK_LOGOUT_URI, CM_KEYCLOAK_JWKS_URI, CM_KEYCLOAK_ISSUER_URI, CM_KEYCLOAK_ADMIN_TOKEN_URI, CM_KEYCLOAK_ADMIN_BASE_URI	CM_KEYCLOAK_LOGIN_CLIENT_SECRET, CM_KEYCLOAK_ADMIN_CLIENT_SECRET
ide	IDE_KEYCLOAK_ISSUER_URL, IDE_KEYCLOAK_AUTHORIZATION_URL, IDE_KEYCLOAK_TOKEN_URL, IDE_KEYCLOAK_JWKS_URL, IDE_KEYCLOAK_LOGOUT_URL, IDE_SERVICE_TOKEN_URL	IDE_KEYCLOAK_CLIENT_SECRET, IDE_SERVICE_CLIENT_SECRET
ai-backend	AI_BACKEND_AUTH_ISSUER_URL, AI_BACKEND_AUTH_AUTHORIZATION_URL, AI_BACKEND_AUTH_TOKEN_URL, AI_BACKEND_AUTH_LOGOUT_URL, AI_BACKEND_AUTH_JWKS_URL	AI_BACKEND_AUTH_CLIENT_SECRET
mcp	MCP_KEYCLOAK_JWKS_URL, MCP_KEYCLOAK_ISSUER	-

При внешнем Keycloak внутренний (127.0.0.1) и публичный URL объединяются в один базовый URL.

7.4 Секрет-файл

Создайте каталог и разместите в нём файл с секретами:

```
sudo install -d -m0755 /etc/selena/external
sudo install -m0600 -o root -g root examples/external-secrets.env
/etc/selena/external/secrets.env
sudoedit /etc/selena/external/secrets.env # заменить все CHANGE_ME
```

Формат - systemd EnvironmentFile: KEY=значение до конца строки (без кавычек). Один файл может содержать секреты для всех сервисов (имена переменных имеют префиксы).

Файл должен существовать до запуска install.sh, иначе установка завершится ошибкой (fail fast).

7.5 Установка и Engine bootstrap

Далее следуйте шагам разделов 4 «Установка Selena (online)» или 5 «Установка Selena (offline)».

На этапе preflight инсталлятор выдаст предупреждение о том, что встроенные PostgreSQL и Keycloak отключены. Это ожидаемое поведение, ошибкой не является.

Миграции IDE и AI будут выполнены инсталлятором во внешний PostgreSQL через alembic. Переопределение параметров подключения действует как при установке (install.sh), так и при обновлении (upgrade.sh).

Для автоматической настройки Engine при использовании внешнего Keycloak в файле secrets.env необходимо указать `CM_BOOTSTRAP_KC_USERNAME` и `CM_BOOTSTRAP_KC_PASSWORD`. Учетная запись должна состоять в группе /cm/admins и иметь разрешенный password grant (для клиента selena-engine включена опция Direct Access Grants).

Если password grant запрещен политикой Keycloak, фаза engine_bootstrap пропускается (выдается предупреждение). В этом случае Engine добавляется вручную через Cluster Manager UI с использованием той же учетной записи.

Для сценария «Инфра + Control Plane» (например, examples/online.deploy.yaml) фаза engine_bootstrap также пропускается. Переменные `CM_BOOTSTRAP_KC_USERNAME` и

`CM_BOOTSTRAP_KC_PASSWORD` требуются только в сценариях, предусматривающих установку Engine.

7.6 Ограничения

- TLS внешнего Keycloak. Сертификат должен быть доверенным для системного хранилища хоста. Для самоподписанного сертификата необходимо добавить корневой сертификат (CA) в доверенные до запуска `install.sh`.
- Апгрейды (`upgrade.sh`). Все настройки из `vars` обновляются из `deploy.yaml`. Файл с секретами (`secrets.env`) не изменяется.
- Файл `credentials.txt`. Не содержит данных о внешних сервисах. Пароли от внешних PostgreSQL и Keycloak известны только администратору.
- Vault. Секрет-файл может рендерить Vault Agent:
`template → /etc/selena/external/secrets.env + systemctl try-restart`
затронутых юнитов, - для инсталлятора это прозрачно.
- Ротация значений в `/etc/selena/external/secrets.env` не отслеживается `upgrade.sh`. После правки файла вручную перезапустите затронутые юниты: `systemctl, try-restart, selena-cm, selena-ide-web, selena-ide-worker, selena-ide-scheduler, selena-ai-backend, selena-lakekeeper`.
- Не редактируйте `examples/external-secrets.env` в чекауте репозитория - копируйте в `/etc/selena/external` и правьте копию (риск закоммитить реальные секреты).
- Комбинация «встроенный Keycloak + внешний PostgreSQL» поддерживается, но БД keycloak должна существовать до старта Keycloak.

8 Частые ошибки и решение

Таблица 8.1. Ошибки при установке Selena

Проблема	Причина	Решение
publicHost: localhost	UI недоступен извне	Укажите реальный IP-адрес или DNS-имя и откройте порты в Security Group и firewalld
Ошибка 404 на этапе fetch	Версия в artifacts-конфиге отсутствует в Nexus	Проверьте наличие пути в реестре
BE не запускается (GLIBC_2.35 not found / отсутствие AVX2)	Неподдерживаемая ОС или CPU Engine-нод	См. раздел 3.2.5.1 «ОС Engine-нод (для BE/CN)»
Ошибка Ansible: couldn't resolve module ansible.posix.sysctl	Коллекции установлены без флага -p /usr/share/ansible/collections	Установите коллекции с флагом -p /usr/share/ansible/collections
Ошибка: No space left при распаковке StarRocks	Корневой раздел (root LVM) на Engine-ноде слишком мал	Увеличьте раздел (см. раздел 3.2.5.5 «Root LVM»)
Ошибка SELinux (203/EXEC) на RHEL	Инсталлятор выполняет restorecon автоматически	При ручных изменениях выполните restorecon -RF /opt/selena