



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 В KUBERNETES

Версия программы 3.0

Москва
2026

Оглавление

- 1 Общая информация
- 2 Чеклист подготовки кластера
- 3 Установка внешних зависимостей
- 4 Keycloak, LDAP и OAuth
- 5 Values, Secrets и Images
- 6 Установка базовой платформы Selena
- 7 Engine cluster, bootstrap и storage
- 8 IDE, AI/MCP, RBAC, пользователи и direct SQL
- 9 Приемка, эксплуатация и troubleshooting
- 10 Selena grants, roles и CM API
- 11 Состав поставки (Delivery Kit Contents)
- 12 Values-prod — самодостаточные values для production

1 Общая информация

Руководство предназначено для администраторов, устанавливающих Selena в свой Kubernetes-кластер. Разделы ниже описывают поддерживаемый production flow.

1.1 Порядок установки

1. Подготовить Kubernetes, DNS, registry, storage и внешние зависимости.
2. Установить foundational external dependencies: PostgreSQL, Keycloak runtime и, если нужен demo, OpenLDAP по разделу Установка внешних зависимостей.
3. Настроить Keycloak realm, groups, clients и token claims по разделу Keycloak, LDAP и OAuth.
4. Доустановить external dependencies, которым нужны уже созданные Keycloak clients, например bundled monitoring/Grafana OAuth; также установить MinIO и Lakekeeper, если они нужны в выбранной схеме.
5. Создать application Secrets для CM, IDE и AI, когда уже известны database endpoints/passwords и Keycloak client secrets.
6. Установить только базовую платформу Selena: Selena operator и Selena Cluster Manager.
7. Через CM API установить Selena Engine, дождаться `READY` и выполнить `bootstrap-auth-rbac`.
8. Только после bootstrap установить IDE, AI backend/frontend и MCP.
9. Настроить Selena roles/grants, привязать Keycloak groups к Selena roles, проверить direct SQL и UI. Подробный справочник по grants и object-level privileges: Selena grants, roles и CM API.

IDE, AI и MCP намеренно ставятся после Engine bootstrap. До bootstrap еще не готовы runtime credentials и Engine endpoint, поэтому ранний deploy этих сервисов создает лишние `Pending/Unready` симптомы и усложняет приемку.

1.2 External dependencies

Подробный порядок установки внешних зависимостей вынесен в Установка внешних зависимостей. Его результат - готовые endpoints/credentials для PostgreSQL, Keycloak, S3, Lakekeeper и других зависимостей; эти значения затем используются в Values, Secrets и Images.

1.3 Разделы

1. Preflight — подготовка кластера, DNS, storage и registry access.
2. Установка внешних зависимостей — установка или проверка PostgreSQL, Keycloak, monitoring/Grafana, MinIO, Lakekeeper и demo LDAP.
3. Keycloak, LDAP и OAuth — Keycloak, LDAP/AD federation, groups, clients, claims и Grafana roles.
4. Values, Secrets и Images — минимальные values, application Secrets, images и справочник свойств.
5. Установка базовой платформы Selena — установка Selena operator, CM и загрузка license.
6. Engine cluster, bootstrap и storage — установка Engine через CM API, bootstrap и storage.
7. IDE, AI/MCP, RBAC, пользователи и direct SQL — установка IDE/AI/MCP после bootstrap, RBAC, group sync и direct SQL.
8. Приемка, эксплуатация и troubleshooting — финальная приемка, operations и troubleshooting.
9. Selena grants, roles и CM API — подробный справочник по Selena roles, grants, object types и CM API.
10. Состав поставки — Helm charts и минимальные values-файлы: internal и external.

1.4 Placeholder format

В командах используется формат `{variable_name}`. Это не shell variable. Перед выполнением команды замените placeholder на фактическое значение.

Пример:

```
curl -fsS {cm_url}/actuator/health
```

Если ваш CM API опубликован как `https://cm.example.com/api/v1`, выполняйте:

```
curl -fsS https://cm.example.com/api/v1/actuator/health
```

Значения, которые возвращает API, берите из предыдущего ответа:

Placeholder	Откуда взять
{cm_admin_access_token}	Поле <code>.accessToken</code> из ответа <code>POST /auth/login</code> .
{operation_id}	Поле <code>.operationId</code> из ответа <code>lifecycle operation</code> .
{bootstrap_operation_id}	Поле <code>.operationId</code> из ответа <code>POST /engine/cluster:bootstrap-auth-rbac</code> .
{cli_secret}	Поле <code>.cliSecret</code> из ответа <code>POST /me/cli-secret:reset</code> ; показывается один раз.

Для сложных overrides с URL, JSON или списками используйте отдельный values overlay file и подключайте его через `-f`. Не передавайте такие значения через `helm --set:` символы `:`, `,`, `{}` и nested keys легко интерпретируются Helm CLI как разделители, и в итоге chart получает не то значение.

1.5 Частые placeholders

Placeholder	Значение
{selena_namespace}	Namespace для CM, IDE, AI и shared Secrets, обычно <code>selena</code> .
{engine_namespace}	Namespace для Selena Engine, обычно <code>selena-engine</code> .
{operator_namespace}	Namespace Selena operator, обычно <code>selena-operator</code> .
{monitoring_namespace}	Namespace monitoring/Grafana, обычно <code>selena-monitoring</code> .
{registry}	Registry/repository prefix с Selena images, например <code>registry.example.com/selena</code> .
{registry_host}	Host registry для image pull Secret.
{registry_username}	Username registry service account.
{registry_password}	Password/token registry service account.
{keycloak_public_url}	Browser-facing Keycloak URL, например <code>https://keycloak.example.com</code> .
{keycloak_internal_url}	In-cluster Keycloak URL, например <code>http://selena-keycloak-http.selena-keycloak.svc.cluster.local</code> .
{cm_url}	CM API URL с <code>/api/v1</code> , например <code>https://cm.example.com/api/v1</code> .
{cm_internal_url}	In-cluster CM URL, например <code>http://selena-cm.selena.svc.cluster.local:8088/api/v1</code> .
{ide_public_url}	Browser-facing Selena IDE URL.
{ai_public_url}	Browser-reachable AI iframe URL. Это может быть отдельный host или path под IDE host.
{grafana_public_url}	Browser-facing Grafana URL.
{selena_fe_mysql_host}	Host/IP для direct SQL на port <code>9030</code> .
{storage_class}	Kubernetes StorageClass для PVC.
{engine_tag}	Selena Engine image tag.
{selena_cm_tag}	Selena CM image tag.
{selena_ide_tag}	Selena IDE image tag.

Placeholder	Значение
<code>{selenai_backend_tag}</code>	Selena AI backend image tag.
<code>{selenai_frontend_tag}</code>	Selena AI frontend image tag.
<code>{selenamcp_tag}</code>	Selena MCP image tag.
<code>{postgres_image_tag}</code>	PostgreSQL image tag, если ставите bundled PostgreSQL chart.
<code>{keycloak_image_tag}</code>	Keycloak image tag, если ставите bundled Keycloak chart.
<code>{openldap_image_tag}</code>	OpenLDAP image tag, только для test/demo LDAP.
<code>{minio_image_tag}</code>	MinIO image tag, если ставите bundled MinIO tenant.
<code>{lakekeeper_image_tag}</code>	Lakekeeper catalog image tag.
<code>{vault_image_tag}</code>	Vault image tag, если ставите optional bundled Vault.
<code>{postgres_service_host}</code>	Kubernetes DNS PostgreSQL service, например <code>selenapostgres.selenapostgres.svc.cluster.local</code> .
<code>{grafana_login_url_encoded}</code>	URL-encoded Grafana login URL для Keycloak logout, например <code>https%3A%2F%2Fgrafana.example.com%2Flogin</code> .

1.6 Общие правила установки

- Production secrets не кладутся в ConfigMap или plaintext values.
- Public browser endpoints должны быть стабильны до настройки Keycloak clients.
- Direct SQL endpoint публикуется только approved сетевым способом: LoadBalancer, TCP Gateway/Ingress controller feature, private endpoint или VPN-only address.
- LDAP/AD, если используется, подключается к Keycloak federation.
- Пользователи, группы, роли и доступы настраиваются через шаги этого guide и CM API.

2 Чеклист подготовки кластера

Этот чеклист подготавливает Kubernetes-кластер до установки Selena. Не пропускайте его: большинство проблем установки связано не с Selena, а с неподготовленными DNS, StorageClass, registry access, Keycloak URL или Kubernetes permissions.

2.1 Проверьте доступ к Kubernetes

Чеклист:

- ☐ `kubectl` смотрит в нужный cluster.
- ☐ Есть права создавать namespaces, Secrets, ConfigMaps, Services, Deployments, StatefulSets, PVC и CRD.
- ☐ Доступен Helm 3.
- ☐ В cluster хватает CPU, memory и storage под выбранный размер Engine.

Команды:

```
kubectl config current-context
kubectl get nodes -o wide
kubectl get storageclass
helm version
```

Проверка прав:

```
kubectl auth can-i create namespaces
kubectl auth can-i create secrets -A
kubectl auth can-i create configmaps -A
kubectl auth can-i create persistentvolumeclaims -A
kubectl auth can-i create services -A
kubectl auth can-i create deployments.apps -A
kubectl auth can-i create statefulsets.apps -A
kubectl auth can-i create customresourcedefinitions.apirestensions.k8s.io
```

Ожидаемо: `yes` для всех прав, которые нужны выбранному installation path.

2.2 Выберите namespaces

Рекомендованный baseline:

Namespace	Для чего
<code>sena</code>	CM, IDE, AI backend, AI frontend, MCP и shared application Secrets.
<code>sena-operator</code>	Sena operator, если operator ставится отдельно от application namespace.
<code>sena-engine</code>	Sena Engine FE/CN или FE/BE pods и services.
<code>sena-keycloak</code>	Keycloak, если ставится в этот же cluster.
<code>sena-postgres</code>	PostgreSQL, если ставится в этот же cluster.
<code>sena-monitoring</code>	Prometheus и Grafana, если ставятся в этот же cluster.
<code>sena-minio-operator</code>	MinIO Operator, если bundled MinIO ставится в этот же cluster.
<code>sena-minio</code>	MinIO, если используется S3-compatible storage в Kubernetes.
<code>sena-lakekeeper</code>	Lakekeeper, если используется Iceberg REST catalog.

Создайте namespaces:

```
kubectl create namespace "{sena_namespace}" --dry-run=client -o yaml |
kubectl apply -f -
kubectl create namespace "{operator_namespace}" --dry-run=client -o yaml |
kubectl apply -f -
kubectl create namespace "{engine_namespace}" --dry-run=client -o yaml |
kubectl apply -f -
```

Если зависимости тоже ставятся в этот cluster:

```
kubectl create namespace sena-keycloak --dry-run=client -o yaml | kubectl
apply -f -
kubectl create namespace sena-postgres --dry-run=client -o yaml | kubectl
apply -f -
kubectl create namespace sena-monitoring --dry-run=client -o yaml |
kubectl apply -f -
kubectl create namespace sena-minio-operator --dry-run=client -o yaml |
kubectl apply -f -
kubectl create namespace sena-minio --dry-run=client -o yaml | kubectl
apply -f -
kubectl create namespace sena-lakekeeper --dry-run=client -o yaml |
kubectl apply -f -
```

2.3 Подготовьте DNS и публикацию сервисов

Чеклист:

- ☐ Решено, как пользователи открывают Keycloak, CM, IDE, Grafana и Selena FE MySQL endpoint.
- ☐ Если используется AI Agent, определен browser-reachable URL, который IDE откроет во встроеном iframe. Это может быть отдельный host или path под тем же доменом, что IDE.
- ☐ Для production browser-facing endpoints используется HTTPS.
- ☐ Keycloak public issuer URL стабилен до настройки OIDC clients.
- ☐ PostgreSQL, LDAP/AD, internal Prometheus и service-to-service ports не опубликованы наружу без явного approval security-команды.

Типовые варианты:

Вариант	Когда использовать
Ingress или API gateway	Рекомендованный production-вариант для HTTP(S).
Cloud LoadBalancer	Подходит, если каждому service нужен отдельный внешний адрес.
NodePort	Обычно только controlled/dev environments.
Port-forward	Только временная admin-проверка, не production exposure.

См. Kubernetes documentation:

- Ingress для HTTP/HTTPS routing.
- Services для ClusterIP, NodePort и LoadBalancer.

Минимальные browser-facing HTTP endpoints:

Service	Пример
Keycloak	<code>https://keycloak.example.com</code>
CM API	<code>https://cm.example.com/api/v1</code>
IDE	<code>https://ide.example.com</code>
AI iframe URL	<code>https://ide.example.com/selena/chat</code> или <code>https://ai.example.com</code>
Grafana	<code>https://grafana.example.com</code>

Direct SQL endpoint:

Service	Пример
Selena FE MySQL	<code>selena-sql.example.com:9030</code>

Минимальный Ingress example для HTTP services в `{selena_namespace}`:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: selena-http
  namespace: {selena_namespace}
spec:
  ingressClassName: {ingress_class_name}
  tls:
    - hosts:
        - cm.example.com
        - ide.example.com
      secretName: {tls_secret_name}
  rules:
    - host: cm.example.com
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: selena-cm
                port:
                  number: 8088
    - host: ide.example.com
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: selena-ide
                port:
                  number: 10001
```

Keycloak и Grafana могут быть в других namespaces. В этом случае создайте отдельный Ingress в соответствующем namespace или используйте ваш стандартный API gateway. Для AI iframe URL можно использовать отдельный host `https://ai.example.com` или path под IDE host, если ваш reverse проxy умеет маршрутизировать этот path в AI frontend Service.

MySQL-compatible endpoint `9030` не является HTTP. Не публикуйте его через обычный HTTP Ingress. Используйте `LoadBalancer`, TCP Gateway/Ingress controller feature, private endpoint или VPN-only address.

Проверка DNS:

```
dig +short keycloak.example.com
dig +short cm.example.com
dig +short ide.example.com
dig +short grafana.example.com
```

2.4 Подготовьте persistent storage

Чеклист:

- ☐ Выбран StorageClass для PostgreSQL.
- ☐ Выбран StorageClass для FE metadata/log volumes.
- ☐ Выбран StorageClass для CN или BE data/cache/log volumes.
- ☐ Dynamic provisioning PVC работает.
- ☐ Есть backup/snapshot policy для PostgreSQL и Engine volumes.

Минимальные требования:

Workload	Требования к StorageClass
PostgreSQL	<code>ReadWriteOnce</code> , dynamic provisioning, надежный block storage, поддержка snapshot/backup на уровне платформы. Для production желательно <code>allowVolumeExpansion=true</code> .
Selena FE metadata/log volumes	<code>ReadWriteOnce</code> , persistent disk, стабильная latency. Не используйте ephemeral/local-only storage без отдельного HA/backup решения.
Selena CN/BE data/cache/log volumes	<code>ReadWriteOnce</code> , достаточная throughput/IOPS под compute workload, capacity expansion или понятная процедура расширения.
Multi-zone cluster	StorageClass должен быть topology-aware. Обычно лучше <code>volumeBindingMode: WaitForFirstConsumer</code> , чтобы PVC создавался в зоне, где scheduler разместил pod.

Проверьте выбранный StorageClass:

```
kubectl get storageclass {storage_class} -o yaml
```

На что смотреть:

- `provisioner` соответствует storage driver, approved в вашем cluster;
- `volumeBindingMode` подходит вашей topology;
- `allowVolumeExpansion` включен, если вы планируете online/offline расширение PVC;
- reclaim policy соответствует политике заказчика. Для production данных часто требуется `Retain` или отдельная backup/snapshot процедура.

Проверка StorageClass:

```
kubectl get storageclass
```

Проверка dynamic PVC:

```
cat > /tmp/selena-pvc-test.yaml <<YAML
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: selena-pvc-test
  namespace: {selena_namespace}
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: {storage_class}
  resources:
    requests:
      storage: 1Gi
YAML

kubectl apply -f /tmp/selena-pvc-test.yaml
kubectl -n "{selena_namespace}" get pvc selena-pvc-test
kubectl -n "{selena_namespace}" delete pvc selena-pvc-test
```

Ожидаемо: PVC переходит в `Bound`.

2.5 Подготовьте container registry access

Чеклист:

- ☐ Kubernetes nodes могут pull'ить все Selena images.

- ☐ Для private registry создан image pull Secret в каждом namespace, где будут запускаться pods.
- ☐ Если используется mirror, все image repository/tag в values указывают на mirror.

Что нужно знать:

- image pull Secret имеет type `kubernetes.io/dockerconfigjson`;
- Secret namespace-scoped, поэтому его нужно создать в каждом namespace, где запускаются pods с private images;
- имя Secret должно совпадать с `imagePullSecrets[].name` в values или Engine spec. В этих документах используется имя `selenaregistry`.

Если Selen images находятся в private registry, создайте Secret минимум в:

Namespace	Зачем
<code>{selenamespace}</code>	CM, IDE, AI backend/frontend, MCP.
<code>{engine_namespace}</code>	Selena Engine FE/CN или FE/BE pods.
<code>{operator_namespace}</code>	Selena operator, если operator image тоже private.

Если bundled external images тоже mirror'ятся в private registry, создайте такой же Secret в namespaces этих dependencies.

Создайте pull Secret:

```
kubectl -n {selena_namespace} create secret docker-registry selena-registry \
--docker-server={registry_host} \
--docker-username={registry_username} \
--docker-password={registry_password} \
--docker-email={registry_email}

kubectl -n {engine_namespace} create secret docker-registry selena-registry \
--docker-server={registry_host} \
--docker-username={registry_username} \
--docker-password={registry_password} \
--docker-email={registry_email}

kubectl -n {operator_namespace} create secret docker-registry selena-
registry \
--docker-server={registry_host} \
--docker-username={registry_username} \
--docker-password={registry_password} \
--docker-email={registry_email}
```

Если Secret уже существует и нужно обновить его без ручного удаления:

```
kubectl -n {selena_namespace} create secret docker-registry selena-registry \
--docker-server={registry_host} \
--docker-username={registry_username} \
--docker-password={registry_password} \
--docker-email={registry_email} \
--dry-run=client -o yaml | kubectl apply -f -
```

Повторите эту команду для каждого namespace.

Если у администратора уже есть локальный Docker config с доступом к registry:

```
kubectl -n {selena_namespace} create secret generic selena-registry \
--from-file=.dockerconfigjson={path_to_docker_config_json} \
--type=kubernetes.io/dockerconfigjson
```

Обычно `{path_to_docker_config_json}` равен `~/.docker/config.json` на машине, где выполнен `docker login {registry_host}`.

Пропишите Secret в values:

```
imagePullSecrets:
  - name: selena-registry
```

Для Selena operator ключ находится внутри `selenaOperator`:

```
selenaOperator:
  imagePullSecrets:
    - name: selena-registry
```

Для Engine cluster Secret задается в CM API spec:

```
"imagePullSecrets": [
  { "name": "selena-registry" }
]
```

Проверьте Secret:

```
kubectl -n {selena_namespace} get secret selena-registry
kubectl -n {engine_namespace} get secret selena-registry
kubectl -n {operator_namespace} get secret selena-registry
kubectl -n {selena_namespace} get secret selena-registry \
  -o jsonpath='{.type}{"\n"}'
```

Ожидаемый type: `kubernetes.io/dockerconfigjson`.

Kubernetes documentation:

- Pull an Image from a Private Registry
- Specifying imagePullSecrets on a Pod

2.6 Подготовьте внешние зависимости

Selena может использовать уже существующие customer-managed зависимости или зависимости, установленные в тот же Kubernetes cluster.

Обязательные:

Dependency	Используется для
Keycloak	Web login, token validation, источник group membership.
PostgreSQL	CM DB, Keycloak DB если Keycloak in-cluster, IDE metadata, AI sessions.

Обычно нужны в production:

Dependency	Используется для
S3-compatible object storage	Shared data storage и Iceberg workflows.
Prometheus и Grafana	Monitoring и dashboards.

Optional:

Dependency	Используется для
LDAP/AD	Enterprise users/groups за Keycloak federation.
Lakekeeper	Iceberg REST catalog.
External Secrets или Vault	Доставка secret values в Kubernetes Secrets.

3 Установка внешних зависимостей

Этот документ готовит external dependencies, от которых дальше зависят Selena components: PostgreSQL, Keycloak, OpenLDAP demo, monitoring/Grafana, MinIO и Lakekeeper.

На выходе из этого шага должны быть известны endpoints, users/passwords, Keycloak runtime URL, S3/Lakekeeper параметры и другие значения, которые дальше используются в application values и Secrets Selena.

Если зависимости предоставляет platform team, не ставьте bundled charts. Проверьте доступность endpoints, получите credentials через процесс заказчика и переходите к Keycloak, LDAP и OAuth для настройки realm/clients.

Для bundled Keycloak есть один сознательный переход между документами: сначала в этом документе ставится Keycloak runtime, затем в документе 02 создаются realm/clients, затем сюда можно вернуться для monitoring/Grafana, которым нужен уже созданный Keycloak client `grafana`.

Рабочая директория:

```
cd {selena_docs_repo}/deploy/k8s
```

3.1 Подготовьте values

Скопируйте минимальные values в рабочие customer-specific файлы и замените placeholders. Не редактируйте chart defaults целиком.

```
cp docs-external/values-minimal/postgres-values.yaml {postgres_values}
cp docs-external/values-minimal/keycloak-values.yaml {keycloak_values}
cp docs-external/values-minimal/openldap-values.yaml {openldap_values}
cp docs-external/values-minimal/kube-prometheus-stack-values.yaml
{monitoring_values}
cp docs-external/values-minimal/minio-operator-values.yaml
{minio_operator_values}
cp docs-external/values-minimal/minio-tenant-values.yaml
{minio_tenant_values}
cp docs-external/values-minimal/lakekeeper-values.yaml {lakekeeper_values}
```

Минимально заменяются:

Values file	Обязательные placeholders
{postgres_values}	{storage_class}
{keycloak_values}	{keycloak_public_url}, {postgres_service_host}
{openldap_values}	{ldap_domain}, {storage_class}
{monitoring_values}	{storage_class}, {grafana_public_url}, {keycloak_public_url}, {grafana_login_url_encoded}
{minio_tenant_values}	{storage_class}
{lakekeeper_values}	{postgres_service_host}

{postgres_service_host} для bundled PostgreSQL обычно:

```
selena-postgres.{postgres_namespace}.svc.cluster.local
```

Пароли и keys из команд ниже генерируются или выдаются secret-management процессом заказчика. В документации они намеренно остаются placeholders:

Placeholder	Что это
{postgres_admin_password}	Пароль PostgreSQL admin user.
{postgres_selena_user_password}	Пароль PostgreSQL application user <code>selena</code> .
{postgres_replication_password}	Пароль replication user, если chart его требует.
{postgres_metrics_password}	Пароль metrics user, если chart его требует.
{keycloak_admin_username} / {keycloak_admin_password}	Bootstrap admin Keycloak.
{grafana_admin_user} / {grafana_admin_password}	Local emergency/admin credentials Grafana.
{minio_root_user} / {minio_root_password}	Root credentials MinIO tenant.
{minio_access_key} / {minio_secret_key}	S3 credentials Selena workloads.

Placeholder	Что это
<code>{lakekeeper_secret_encryption_key}</code>	Stable encryption key для Lakekeeper secret backend.

3.2 PostgreSQL

PostgreSQL должен быть готов раньше Keycloak, Lakekeeper, CM, IDE и AI: им всем нужны databases или credentials из этого шага.

Создайте install-time Secret для PostgreSQL chart:

```
kubectl -n {postgres_namespace} create secret generic selena-postgres-auth \
  --from-literal=postgres-password='{postgres_admin_password}' \
  --from-literal=password='{postgres_selena_user_password}' \
  --from-literal=replication-password='{postgres_replication_password}' \
  --from-literal=metrics-password='{postgres_metrics_password}'
```

Установите PostgreSQL:

```
helm lint charts/external/postgres/chart -f {postgres_values}

helm upgrade --install selena-postgres charts/external/postgres/chart \
  -n {postgres_namespace} -f {postgres_values} --wait --timeout 20m
```

Проверьте:

```
kubectl -n {postgres_namespace} get pods,svc,pvc -l
app.kubernetes.io/instance=selena-postgres
kubectl -n {postgres_namespace} rollout status statefulset/selena-postgres
--timeout=10m
```

После этого известны значения для downstream components:

Placeholder	Значение
<code>{postgres_service_host}</code>	<code>selena-postgres.{postgres_namespace}.svc.cluster.local</code> или host customer-managed PostgreSQL.
<code>{cm_db_jdbc_url}</code>	<code>jdbc:postgresql://{postgres_service_host}:5432/selena_cm</code>
<code>{cm_db_user}</code>	Обычно <code>selena</code> , если используется bundled profile.

Placeholder	Значение
{cm_db_password}	{postgres_selena_user_password} или пароль, выданный platform team.
{ide_database_conn}	postgresql://selena:{postgres_selena_user_password}@{postgres_service_host}:5432/selena_ide
{ai_backend_database_url}	postgresql+asyncpg://selena:{postgres_selena_user_password}@{postgres_service_host}:5432/selena_ai

3.3 Demo LDAP, если нужен

OpenLDAP chart нужен только для demo/test federation. Для production обычно используется customer LDAP/AD, подключенный к Keycloak.

Создайте install-time Secret:

```
kubectl -n {openldap_namespace} create secret generic selena-openldap-auth \
  --from-literal=LDAP_ADMIN_USERNAME='{ldap_admin_username}' \
  --from-literal=LDAP_ADMIN_PASSWORD='{ldap_admin_password}' \
  --from-literal=LDAP_CONFIG_ADMIN_PASSWORD='{ldap_config_admin_password}'
```

Установите demo LDAP:

```
helm lint charts/external/openldap/chart -f {openldap_values}

helm upgrade --install selena-openldap charts/external/openldap/chart \
  -n {openldap_namespace} -f {openldap_values} --wait --timeout 10m
```

Проверьте:

```
kubectl -n {openldap_namespace} get pods,svc,pvc
```

3.4 Keycloak

Keycloak ставится после PostgreSQL, потому что хранит state в PostgreSQL. После установки Keycloak еще нужно вручную настроить realm, groups, clients и token claims по Keycloak, LDAP и OAuth.

Создайте install-time Secrets:

```
kubectl -n {keycloak_namespace} create secret generic selen-keycloak-admin \
--from-literal=admin-user='{keycloak_admin_username}' \
--from-literal=admin-password='{keycloak_admin_password}'

kubectl -n {keycloak_namespace} create secret generic selen-keycloak-db \
--from-literal=password='{postgres_selen_user_password}'
```

Установите Keycloak:

```
helm lint charts/external/keycloak/chart -f {keycloak_values}

helm upgrade --install selen-keycloak charts/external/keycloak/chart \
-n {keycloak_namespace} -f {keycloak_values} --wait --timeout 15m
```

Проверьте:

```
kubectl -n {keycloak_namespace} get pods,svc
kubectl -n {keycloak_namespace} rollout status statefulset/selen-keycloak
--timeout=10m
curl -fsS {keycloak_public_url}/realms/master/.well-known/openid-
configuration | jq -r .issuer
```

Эта команда проверяет только Keycloak runtime в `master` realm. Рабочий Selen issuer появится после создания realm `selen` и проверяется в документе 02 как `{keycloak_public_url}/realms/selen`.

Дальше выполните Keycloak, LDAP и OAuth. На этом шаге появятся Keycloak client secrets для:

```
selen-engine
selen-cluster-manager-api
selen-ide
selen-ide-service
selen-ai
grafana
```

Client `selena-ai-service` создавайте только если ваш deployment profile явно использует AI service-to-service calls в CM. В текущем minimal AI chart для него нет application Secret.

3.5 Monitoring и Grafana

Если monitoring/Grafana предоставляет platform team, пропустите этот раздел.

Если используете bundled monitoring с Grafana OAuth, сначала создайте client `grafana` в Keycloak по документу 02. Без client secret chart не сможет создать корректный Grafana OAuth config.

Создайте install-time Secrets для Grafana:

```
kubectl -n {monitoring_namespace} create secret generic selena-grafana-admin \
  --from-literal=admin-user='{grafana_admin_user}' \
  --from-literal=admin-password='{grafana_admin_password}'

kubectl -n {monitoring_namespace} create secret generic selena-grafana-keycloak-client \
  --from-literal=client-secret='{grafana_client_secret}'
```

Установите Prometheus Operator CRD и monitoring chart:

```
kubectl create --save-config=false \
  -f charts/external/kube-prometheus-stack/chart/charts/crds/crds/ \
  || kubectl apply --server-side --force-conflicts \
  -f charts/external/kube-prometheus-stack/chart/charts/crds/crds/

helm lint charts/external/kube-prometheus-stack/chart -f
{monitoring_values}

helm upgrade --install selena-monitoring charts/external/kube-prometheus-stack/chart \
  -n {monitoring_namespace} -f {monitoring_values} --wait --timeout 20m
```

Не используйте обычный `kubectl apply` для этих CRD: большие Prometheus Operator CRD могут превысить Kubernetes annotation limit из-за `kubectl.kubernetes.io/last-applied-configuration`.

Проверьте:

```
kubectl get crd servicemonitors.monitoring.coreos.com
podmonitors.monitoring.coreos.com
kubectl -n {monitoring_namespace} get pods,svc,pvc
```

3.6 MinIO, если нужен bundled S3

Если S3-compatible object storage предоставляет platform team, пропустите этот раздел и используйте выданный endpoint, access key, secret key и bucket names.

Установите MinIO Operator:

```
helm lint charts/external/minio/operator/chart -f {minio_operator_values}

helm upgrade --install minio-operator charts/external/minio/operator/chart \
  -n {minio_operator_namespace} -f {minio_operator_values} --wait --timeout
10m
```

Создайте install-time Secrets для tenant-a:

```
kubectl -n {minio_namespace} create secret generic selena-minio-env-
configuration \
  --from-literal=config.env='export MINIO_ROOT_USER="{minio_root_user}"
export MINIO_ROOT_PASSWORD="{minio_root_password}"'

kubectl -n {minio_namespace} create secret generic selena-minio-user \
  --from-literal=CONSOLE_ACCESS_KEY='{minio_access_key}' \
  --from-literal=CONSOLE_SECRET_KEY='{minio_secret_key}'
```

Установите tenant:

```
helm lint charts/external/minio/tenant/chart -f {minio_tenant_values}

helm upgrade --install selena-minio charts/external/minio/tenant/chart \
  -n {minio_namespace} -f {minio_tenant_values} --wait --timeout 15m
```

Проверьте:

```
kubectl -n {minio_namespace} get pods,svc,pvc
```

3.7 Lakekeeper, если нужен Iceberg REST catalog

Lakekeeper ставится после PostgreSQL. Если Iceberg REST catalog предоставляет platform team, пропустите этот раздел и используйте выданный endpoint.

Создайте install-time Secrets:

```
kubectl -n {lakekeeper_namespace} create secret generic selena-lakekeeper-  
db \  
  --from-literal=postgresql-user='selena' \  
  --from-literal=postgresql-password='{postgres_selena_user_password}'  
  
kubectl -n {lakekeeper_namespace} create secret generic selena-lakekeeper-  
postgres-encryption \  
  --from-literal=encryptionKey='{lakekeeper_secret_encryption_key}'
```

Установите Lakekeeper:

```
helm lint charts/external/lakekeeper/chart -f {lakekeeper_values}  
  
helm upgrade --install selena-lakekeeper charts/external/lakekeeper/chart \  
  -n {lakekeeper_namespace} -f {lakekeeper_values} --wait --timeout 15m
```

Проверьте:

```
kubectl -n {lakekeeper_namespace} get pods,svc
```

3.8 Что должно быть готово дальше

Перед созданием application Secrets и установкой Selena platform должны быть известны:

Значение	Откуда берется
CM_DB_URL, CM_DB_USER, CM_DB_PASSWORD	PostgreSQL из раздела 2 или customer-managed DB.

Значение	Откуда берется
Keycloak public/internal URLs	Keycloak из раздела 4 или customer-managed Keycloak.
Keycloak client secrets	Настройка clients в документе 02.
Grafana client secret	Keycloak client <code>grafana</code> , если ставите bundled Grafana.
S3 endpoint/credentials/buckets	MinIO из раздела 6 или customer-managed S3.
Lakekeeper endpoint	Lakekeeper из раздела 7 или customer-managed catalog.

После этого переходите к Values, Secrets и Images и создавайте application Secrets для CM, IDE и AI.

4 Keycloak, LDAP и OAuth

Этот чеклист настраивает Keycloak для Selena через Keycloak Admin Console. Keycloak может быть внешним customer-managed сервисом или компонентом, установленным в тот же Kubernetes cluster. Selena Helm charts не создают realm, users, groups, clients и LDAP federation.

4.1 Создайте или выберите realm

Чеклист:

- ☐ Realm называется `selena` или все Selena values адаптированы под другое имя realm.
- ☐ Browser-facing issuer стабилен до создания clients.
- ☐ Issuer в tokens совпадает с настройками CM, IDE, AI, MCP и Grafana.

В Keycloak Admin Console:

1. Откройте Keycloak Admin Console.
2. В левом верхнем углу откройте realm selector.
3. Нажмите `Create realm`.
4. В поле `Realm name` введите `selena`.
5. Убедитесь, что `Enabled` включен.
6. Нажмите `Create`.

Проверьте issuer:

1. Перейдите в `Realm settings`.
2. Откройте `General`.
3. Найдите ссылку `OpenID Endpoint Configuration`.
4. Откройте ее и проверьте поле `issuer`.

Ожидаемое значение:

```
{keycloak_public_url}/realms/selena
```

Например:

```
https://keycloak.example.com/realms/selena
```

Это значение должно совпадать с `{keycloak_public_url}/realms/selena` во всех Selenia values.

4.2 Настройте LDAP или Active Directory federation

Пропустите этот раздел только если все users локальные в Keycloak.

Полезные официальные справки:

Документация	Когда открыть
Keycloak Server Administration Guide: LDAP and Active Directory	Основная справка по настройке <code>User federation</code> в Keycloak: LDAP provider, storage/edit mode, sync, LDAPS, connection pool и troubleshooting.
Keycloak Server Administration Guide: LDAP mappers	Когда настраиваете mapper для групп, user attributes или нестандартной LDAP/AD schema.
OpenLDAP Administrator's Guide	Если LDAP server построен на OpenLDAP и нужно уточнить DN, suffix, schema, access rules, replication или диагностику LDAP server-side.
Microsoft: Configure LDAPS for Active Directory Domain Services	Если используется Microsoft AD и Keycloak должен подключаться по LDAPS (<code>ldaps://...:636</code>).

Чеклист:

- ☐ Keycloak может подключиться к LDAP/AD.
- ☐ Bind DN имеет read-only доступ к users и groups.
- ☐ Username в Keycloak совпадает с username, которым пользователь будет входить в Selenia UI.
- ☐ Group membership виден в Keycloak.

В Keycloak Admin Console:

1. Выберите realm `selena`.
2. Перейдите в `User federation`.
3. Нажмите `Add provider`.
4. Выберите `ldap`.
5. Заполните connection и user lookup settings.
6. Сохраните provider.

Типовые настройки LDAP/AD provider:

Setting	Пример	Комментарий
Vendor	Active Directory или Other	Выберите closest match.
Connection URL	ldaps://ad.example.com:636	Для production используйте TLS.
Bind type	simple	Обычно достаточно для service account.
Bind DN	CN=svc-keycloak,OU=Service Accounts,DC=example,DC=com	Read-only service account.
Bind credential	password из secret-management процесса заказчика	Не храните пароль в документации или ConfigMap.
Users DN	OU=Users,DC=example,DC=com	Base DN для user search.
Username LDAP attribute	sAMAccountName или uid	Это станет Keycloak username.
RDN LDAP attribute	cn	Обычно cn, но зависит от directory schema.
UUID LDAP attribute	objectGUID или entryUUID	Для AD обычно objectGUID, для LDAP часто entryUUID.
User object classes	person, organizationalPerson, user или customer schema	Должно соответствовать directory schema.
Edit mode	READ_ONLY	Рекомендуется для enterprise LDAP/AD.
Import users	ON	Рекомендуется для стабильного lookup и membership sync.
Sync registrations	OFF	Обычно users создаются в LDAP/AD, не через Keycloak.

Добавьте group mapper:

1. Откройте созданный LDAP provider.
2. Перейдите во вкладку Mappers.
3. Нажмите Add mapper.
4. Выберите mapper type group-ldap-mapper.
5. Заполните group lookup settings.
6. Сохраните mapper.

Типовые настройки group mapper:

Setting	Пример	Комментарий
Mapper type	group-ldap-mapper	Читает LDAP/AD groups в Keycloak groups.
LDAP Groups DN	OU=Groups,DC=example,DC=com	Base DN для group search.
Group name LDAP attribute	cn	Имя группы в Keycloak.
Group object classes	group или groupOfNames	Для AD обычно group.

Setting	Пример	Комментарий
Membership LDAP attribute	member	Атрибут, где group хранит members.
Membership attribute type	DN	Обычно DN.
User roles retrieve strategy	LOAD_GROUPS_BY_MEMBER_ATTRIBUTE	Читает groups через membership attribute.
Preserve group inheritance	ON	Сохраняет иерархию groups, если она есть.
Mode	READ_ONLY	Рекомендуется для enterprise LDAP/AD.

Синхронизируйте users и groups:

1. Вернитесь в **User federation**.
2. Откройте LDAP provider.
3. В **Action** выберите **Sync all users**.
4. Откройте group mapper.
5. В **Action** выберите **Sync LDAP groups to Keycloak**.

Проверьте в UI:

1. **Users** -> найдите LDAP/AD user -> вкладка **Groups**.
2. **Groups** -> откройте imported group -> вкладка **Members**.

Если groups не появились, проверьте **LDAP Groups DN**, membership attribute и права bind account на чтение groups.

4.3 Создайте группы

Чеклист:

- ☐ Созданы CM admin/viewer groups.
- ☐ Созданы Grafana groups.
- ☐ Созданы business groups, которые будут давать доступ к данным в Selena Engine.
- ☐ В CM values используются full group paths.

В Keycloak Admin Console:

1. Перейдите в **Groups**.
2. Для top-level group нажмите **Create group**.
3. Для nested group откройте parent group и используйте **Subgroups** -> **Create group**.
4. После создания добавляйте users через **Users** -> user -> **Groups** -> **Join group**.

Рекомендованный baseline:

Group path	Для чего
<code>/cm/admins</code>	Полное администрирование Selena Cluster Manager. Users из этой группы могут выполнять административные операции CM.
<code>/cm/role-admins</code>	Управление Selena roles/grants через CM без полного cluster admin-доступа.
<code>/cm/viewers</code>	Read-only доступ к CM для просмотра состояния и диагностики.
<code>/grafana/admins</code>	Grafana Admin. Используется Grafana OAuth role mapping.
<code>/grafana/viewers</code>	Grafana Viewer. Используется Grafana OAuth role mapping.
<code>/ldap/selena-readers</code>	Пример business-группы для LDAP/AD users, которым нужен read-доступ к данным в Selena Engine. В CM values эта group path маппится на external group, а затем на Selena role/grants.
<code>/ldap/selena-writers</code>	Пример business-группы для LDAP/AD users, которым нужен write-доступ к данным в Selena Engine. Обычно этой группе назначают более широкие Selena privileges, чем reader-группе.
<code>/local/selena-readers</code>	Пример business-группы для локальных Keycloak users, если заказчик использует local users для demo, service acceptance или отдельных внутренних учеток. Смысл такой же: membership дает доступ к данным через CM mapping и Selena roles.

Группы `/ldap/...` и `/local/...` в таблице являются примерами. В production используйте реальные customer group paths, например `/data/finance/readers` или `/analytics/sales/writers`. Важно, чтобы path был стабильным и совпадал с mapping в `selena-cm-values.yaml`.

В CM этот mapping задается в `docs-external/values-minimal/selena-cm-values.yaml`, в блоке `extraEnv -> SPRING_APPLICATION_JSON`. Там находятся две разные настройки:

Настройка	Что означает
<code>cm.identity.external-groups.mappings</code>	Keycloak group path -> Selena external group name. Это business/data-access mapping: какие Keycloak groups CM синхронизирует для доступа к данным в Selena Engine.
<code>cm.security.group-role-mappings</code>	Keycloak group path -> CM API role. Это access mapping для самого Selena Cluster Manager: кто будет CM admin, role admin или viewer.

Ключи вида `[/ldap/selena-readers]` должны совпадать с full group path в Keycloak. Значения вроде `ldap_selena_readers` или `cm_admin` - это уже имена, которые понимает CM: external group name для Engine-доступа или CM role для доступа к CM API. Подробный RBAC-чеклист находится в IDE, AI/MCP, RBAC, пользователи и direct SQL.

Эти business-группы управляют именно доступом к данным в Selena Engine. Сама группа в Keycloak еще не выдает privilege автоматически: администратор сначала указывает full group path в CM values, затем через CM API создает Selena roles, назначает им privileges и привязывает roles к mapped external groups. После этого scheduled sync CM читает membership из Keycloak и поддерживает актуальные пользовательские права в Selena.

Разделение `/ldap/...` и `/local/...` не является техническим требованием. Это понятный пример, который помогает не смешивать LDAP/AD-backed users и локальных Keycloak users при приемочном тестировании. Если у заказчика уже есть свои группы в AD/Keycloak, используйте их реальные paths.

Для local Keycloak users:

1. Перейдите в `Users`.
2. Нажмите `Add user`.
3. Заполните `Username`, `First name`, `Last name`, `Email`.
4. Включите `Email verified`, если email уже проверен customer process-ом.
5. Сохраните user.
6. Откройте вкладку `Credentials`.
7. Нажмите `Set password`.
8. Введите password и выключите `Temporary`, если user не должен менять пароль при первом входе.
9. Откройте вкладку `Groups`.
10. Нажмите `Join group` и выберите нужные groups.

Для LDAP/AD users профиль и password приходят из federation. Membership можно вести в LDAP/AD или назначать дополнительные Keycloak-local groups, если это разрешено политикой заказчика.

4.4 Создайте OIDC clients

Чеклист:

- ☐ Созданы все required clients.
- ☐ Client secrets сохранены в Kubernetes Secret или external secret store.
- ☐ Redirect URIs используют финальные public URLs.
- ☐ User-facing clients имеют `groups` mapper.
- ☐ Tokens, которые проверяют Selena services, имеют audience `selena-engine`.

В Keycloak Admin Console:

1. Перейдите в **Clients**.
2. Нажмите **Create client**.
3. **Client type** выберите **OpenID Connect**.
4. Введите **Client ID**.
5. Нажмите **Next**.
6. Включите **Client authentication** для confidential clients.
7. Настройте **Authentication flow**.
8. Нажмите **Next**.
9. Заполните **Valid redirect URIs**, **Valid post logout redirect URIs** и **Web origins**, если они нужны client-у.
10. Нажмите **Save**.
11. Для confidential client откройте вкладку **Credentials** и скопируйте **Client secret** в secret-management процесс.

Required clients:

Client ID	Как настроить в Keycloak UI
<code>selen-engine</code>	Client authentication : ON. Standard flow : ON. Direct access grants : ON. Service accounts : OFF. Valid redirect URIs : <code>{cm_public_url}/*</code> . Web origins : <code>{cm_public_url}</code> .
<code>selen-cluster-manager-api</code>	Client authentication : ON. Standard flow : OFF. Direct access grants : OFF. Service accounts : ON. Browser redirect URIs не нужны.
<code>selen-id</code>	Client authentication : ON. Standard flow : ON. Direct access grants : OFF. Service accounts : OFF. Valid redirect URIs : <code>{ide_public_url}/oauth2callback</code> , <code>{ide_public_url}/*</code> . Web origins : <code>{ide_public_url}</code> .
<code>selen-ide-service</code>	Client authentication : ON. Standard flow : OFF. Direct access grants : OFF. Service accounts : ON. Browser redirect URIs не нужны.
<code>selen-ai</code>	Client authentication : ON. Standard flow : ON. Direct access grants : OFF. Service accounts : OFF. Valid redirect URIs : <code>{ai_public_url}/api/v1/auth/callback</code> , <code>{ai_public_url}/*</code> . Web origins : <code>{ai_public_url}</code> , <code>{ide_public_url}</code> .
<code>grafana</code>	Client authentication : ON. Standard flow : ON. Direct access grants : OFF. Service accounts : OFF. Valid redirect URIs : <code>{grafana_public_url}/login/generic_oauth</code> , <code>{grafana_public_url}/*</code> . Valid post logout redirect URIs : <code>{grafana_public_url}/login</code> . Web origins : <code>{grafana_public_url}</code> .

`{ai_public_url}` не обязан быть отдельным публичным порталом. Это URL, который browser может загрузить внутри IDE iframe. Его можно опубликовать как отдельный host, например `https://ai.example.com`, или как path под IDE/API gateway, если reverse proxy маршрутизирует этот path в AI frontend Service.

Optional client `selenai-service` создавайте только если ваш deployment profile явно использует AI service-to-service calls в CM. В текущем minimal AI chart используется browser client `selenai` и trusted IDE service client `selenai-ide-service`; отдельного Secret для `selenai-service` в minimal values нет.

Сохраните client secrets в Kubernetes Secrets из Values, Secrets и Images:

Client ID	Secret key
<code>selenai-engine</code>	<code>CM_KEYCLOAK_LOGIN_CLIENT_SECRET</code>
<code>selenai-cluster-manager-api</code>	<code>CM_KEYCLOAK_ADMIN_CLIENT_SECRET</code>
<code>selenai-ide</code>	<code>IDE_KEYCLOAK_CLIENT_SECRET</code>
<code>selenai-ide-service</code>	<code>IDE_SERVICE_CLIENT_SECRET</code>
<code>selenai-ai</code>	<code>AI_BACKEND_AUTH_CLIENT_SECRET</code>
<code>grafana</code>	<code>selenai-grafana-keycloak-client/client-secret</code>

4.5 Добавьте token mappers

Token mappers нужны, чтобы приложения Selena получали из Keycloak token те claims, по которым они узнают пользователя, группы и назначение token. Keycloak хранит эту информацию в своем user/group model, но приложение видит только то, что попало в issued token. Поэтому mapper здесь означает "положить конкретное поле из Keycloak в конкретный claim access token".

Что именно нужно получить в access token:

Claim	Что маппим	Зачем нужно
<code>preferred_username</code>	Keycloak user username.	Selena components используют это как стабильное имя пользователя. Обычно claim уже приходит через built-in <code>profile</code> scope, но его надо проверить.
<code>groups</code>	Полный список Keycloak groups пользователя в виде full paths, например <code>/cm/admins</code> , <code>/ldap/selena-readers</code> или <code>/data/finance/readers</code> .	CM использует group paths для CM roles и для mapping business groups на Selena roles/grants. Full path нужен, чтобы не перепутать одинаковые leaf names в разных ветках.
<code>aud</code>	Audience <code>selenai-engine</code> .	Selena services отклоняют token, который не предназначен для Selena audience. Это защита от использования token, выпущенного для чужого client-a.

Preferred username mapper

Сначала проверьте, что client получает built-in client scope `profile`:

1. Откройте **Clients**.
2. Выберите user-facing client, например **selen-engine**.
3. Откройте вкладку **Client scopes**.
4. Убедитесь, что **profile** назначен как default scope.

Если **preferred_username** не появляется в access token, добавьте mapper:

1. Откройте **Client scopes**.
2. Создайте или выберите scope, который назначен Selen clients.
3. Откройте **Mappers**.
4. Нажмите **Add mapper**.
5. Выберите mapper type **User Property**.

Settings:

Field	Value
Name	preferred_username
Property	username
Token claim name	preferred_username
Claim JSON Type	String
Add to access token	ON
Add to ID token	ON
Add to userinfo	ON

Groups mapper

Рекомендуемый вариант: создать общий client scope, например **selen-groups**, и назначить его всем user-facing clients.

В Keycloak Admin Console:

1. Перейдите в **Client scopes**.
2. Нажмите **Create client scope**.
3. Name: **selen-groups**.
4. Type: **Default** или **Optional**, в зависимости от политики заказчика. Для Selena проще использовать **Default**, чтобы claim всегда был в token.
5. Сохраните scope.
6. Откройте scope **selen-groups**.
7. Перейдите в **Mappers**.

8. Нажмите `Add mapper`.
9. Выберите mapper type `Group Membership`.

Settings:

Field	Value
Name	groups
Token claim name	groups
Full group path	ON
Add to access token	ON
Add to ID token	ON, если UI читает ID token
Add to userinfo	ON, если приложение вызывает userinfo

Назначьте scope `selena-groups` user-facing clients:

1. Откройте `Clients`.
2. Выберите client.
3. Откройте `Client scopes`.
4. Нажмите `Add client scope`.
5. Выберите `selena-groups`.
6. Назначьте как `Default`, если claim должен приходить всегда.

User-facing clients:

```
selena-engine  
selena-ide  
selena-ai  
grafana
```

Audience mapper

Audience mapper добавляет `selena-engine` в claim `aud`. Это нужно для tokens, которые проверяются Selena services.

Рекомендуемый вариант: создать client scope `selena-audience`.

В Keycloak Admin Console:

1. Перейдите в `Client scopes`.

2. Нажмите `Create client scope`.
3. Name: `selenia-audience`.
4. Type: `Default`.
5. Сохраните scope.
6. Откройте scope `selenia-audience`.
7. Перейдите в `Mappers`.
8. Нажмите `Add mapper`.
9. Выберите mapper type `Audience`.

Settings:

Field	Value
Name	<code>selenia-engine-audience</code>
Included client audience	<code>selenia-engine</code>
Add to access token	ON
Add to ID token	OFF, если нет отдельного требования

Назначьте scope `selenia-audience` этим clients:

```
selenia-engine  
selenia-ide  
selenia-ai  
selenia-ide-service
```

Если в вашем deployment profile используется optional `selenia-ai-service`, тоже назначьте ему `selenia-audience`.

Проверка mappers в Keycloak UI

1. Откройте `Clients`.
2. Выберите client, например `selenia-engine`.
3. Откройте вкладку `Client scopes`.
4. Откройте `Evaluate`.
5. Выберите test user, который состоит в Selenia groups.
6. Нажмите generate/evaluate token.
7. Откройте generated access token.

Проверьте:

Claim	Ожидаемо
preferred_username	Keycloak username test user-a.
groups	Содержит full paths, например /cm/admins или /ldap/selena-readers.
aud	Содержит selena-engine.

Если groups содержит только selena-readers без parent path, включите Full group path в Group Membership mapper. Если aud не содержит selena-engine, проверьте Audience mapper и назначение client scope на client.

4.6 Дайте CM read-доступ к Keycloak Admin API

Чеклист:

- ☐ selena-cluster-manager-api имеет service account.
- ☐ Service account может читать realm users и groups.
- ☐ Service account не имеет broad write-admin ролей без approval.

В Keycloak Admin Console:

1. Перейдите в Clients.
2. Откройте selena-cluster-manager-api.
3. Убедитесь, что Service accounts включен.
4. Откройте вкладку Service account roles.
5. Нажмите Assign role.
6. В фильтре выберите roles клиента realm-management.
7. Назначьте только read/query roles из таблицы ниже.

Required roles:

Role	Зачем
view-realm	Читать metadata realm.
view-users	Читать users.
query-users	Искать users по username.
query-groups	Искать groups и membership.

Если в CM на экране Группы и SQL роли виден warning, что CM не видит группы Keycloak, сначала проверьте именно эти service account roles у selena-cluster-manager-api. Пользовательская роль cm_admin даёт права внутри CM, но не заменяет технический read-доступ backend-а к Keycloak Admin API.

Не назначайте `manage-users`, `manage-realm`, `realm-admin` или другие write roles без отдельного security approval.

4.7 Настройте Grafana roles

Чеклист:

- ☐ Созданы realm roles `grafana_admin` и `grafana_viewer`.
- ☐ `/grafana/admins` получает `grafana_admin`.
- ☐ `/grafana/viewers` получает `grafana_viewer`.
- ☐ Grafana OAuth config использует strict role mapping.
- ☐ Пользователь без Grafana role не может войти в Grafana.

Создайте realm roles:

1. В Keycloak Admin Console откройте `Realm roles`.
2. Нажмите `Create role`.
3. Создайте role `grafana_admin`.
4. Повторите для `grafana_viewer`.

Назначьте roles группам:

1. Откройте `Groups`.
2. Выберите `/grafana/admins`.
3. Откройте `Role mapping`.
4. Нажмите `Assign role`.
5. Выберите realm role `grafana_admin`.
6. Повторите для `/grafana/viewers` и role `grafana_viewer`.

Проверка:

1. Добавьте test user в `/grafana/viewers`.
2. Зайдите в Grafana и проверьте Viewer.
3. Переместите user в `/grafana/admins`.
4. Выйдите и войдите снова; проверьте Admin.
5. Уберите обе Grafana groups.
6. Выйдите и войдите снова; доступ должен быть запрещен.

4.8 Добавьте первого CM admin

Чеклист:

- ☐ Хотя бы один human user состоит в `/cm/admins`.
- ☐ Этот user может войти в CM после установки CM.
- ☐ Token содержит group claim.

В Keycloak Admin Console:

1. Перейдите в `Users`.
2. Найдите нужного user.
3. Откройте вкладку `Groups`.
4. Нажмите `Join group`.
5. Выберите `/cm/admins`.
6. Сохраните membership.

После установки CM выполните login в CM под этим user и проверьте, что user получил роль `cm_admin`.

5 Values, Secrets и Images

Внешний deployment должен переопределять только параметры окружения. Не копируйте весь chart `values.yaml`: chart defaults уже описывают обычное поведение. Минимальные override-файлы лежат в `docs-external/values-minimal/`.

Этот файл не является линейным install script. Используйте его разделы в той фазе установки, где они нужны:

1. Сначала установите или проверьте PostgreSQL, Keycloak и остальные external dependencies по Установка внешних зависимостей. Для bundled charts этот документ отдельно показывает install-time Secrets, которые нужны самим dependency charts.
2. Настройте Keycloak realm и OIDC clients по Keycloak, LDAP и OAuth.
3. Только после этого выполняйте разделы с application Secrets для CM, IDE и AI: они требуют уже известных database endpoints/passwords и Keycloak client secrets. IDE/AI Secrets создаются до Engine bootstrap, потому что bootstrap записывает туда generated impersonator passwords.

5.1 Image pull Secret

Image pull Secret нужен только для private registry. Если registry public, `imagePullSecrets` можно убрать из values.

Не создавайте пустой `kubernetes.io/dockerconfigjson` Secret как замену registry credentials, если values или Engine spec все еще ссылаются на `imagePullSecrets`. Такой Secret проходит type-check, но private registry все равно вернет `ImagePullBackOff`. Либо подтвердите, что конкретные Selena images/tags действительно доступны anonymous pull и уберите `imagePullSecrets`, либо создайте настоящий `docker-registry` Secret.

Secret создается отдельно в каждом namespace: Kubernetes Secret не работает между namespaces. В этих документах используется имя `selena-registry`.

Минимально создайте Secret в:

Namespace	Какие pods используют
<code>{selena_namespace}</code>	CM, IDE, AI backend/frontend, MCP.
<code>{engine_namespace}</code>	Selena Engine pods.
<code>{operator_namespace}</code>	Selena operator, если operator image private.

Создайте Secret:

```
kubectl -n {selena_namespace} create secret docker-registry selena-registry \
  --docker-server={registry_host} \
  --docker-username={registry_username} \
  --docker-password={registry_password} \
  --docker-email={registry_email}

kubectl -n {engine_namespace} create secret docker-registry selena-registry \
  --docker-server={registry_host} \
  --docker-username={registry_username} \
  --docker-password={registry_password} \
  --docker-email={registry_email}

kubectl -n {operator_namespace} create secret docker-registry selena-registry \
  --docker-server={registry_host} \
  --docker-username={registry_username} \
  --docker-password={registry_password} \
  --docker-email={registry_email}
```

Если Secret уже существует, обновляйте его через `--dry-run=client -o yaml | kubectl apply -f -`:

```
kubectl -n {selena_namespace} create secret docker-registry selena-registry \
  --docker-server={registry_host} \
  --docker-username={registry_username} \
  --docker-password={registry_password} \
  --docker-email={registry_email} \
  --dry-run=client -o yaml | kubectl apply -f -
```

Проверьте Secret type:

```
kubectl -n {selena_namespace} get secret selena-registry \
  -o jsonpath='{.type}'
```

Ожидаемо: `kubernetes.io/dockerconfigjson`.

Пропишите Secret в values:

```
imagePullSecrets:  
  - name: selena-registry
```

Для Selena operator ключ вложен в `selenaOperator`:

```
selenaOperator:  
  imagePullSecrets:  
    - name: selena-registry
```

Для Engine cluster Secret указывается в JSON спец, который вы отправляете в CM API:

```
"imagePullSecrets": [  
  { "name": "selena-registry" }  
]
```

Подробности в Kubernetes documentation:

- Pull an Image from a Private Registry
- Specifying imagePullSecrets

5.2 Install-time Secrets для bundled external charts

Эти Secrets нужны только если вы ставите bundled external charts. Если PostgreSQL, Keycloak, LDAP/AD, MinIO, Lakekeeper или Vault предоставлены платформой заказчика, используйте их native secret-management процесс.

Это не application Secrets Selena. Это inputs для самих dependency charts, поэтому они создаются до `helm upgrade --install` соответствующей внешней зависимости. Если external dependency уже установлена platform team, этот раздел пропускается: возьмите выданные endpoint/credentials и переходите к application Secrets после настройки Keycloak.

PostgreSQL:

```
kubectl -n {postgres_namespace} create secret generic selena-postgres-auth \
  --from-literal=postgres-password='{postgres_admin_password}' \
  --from-literal=password='{postgres_selena_user_password}' \
  --from-literal=replication-password='{postgres_replication_password}' \
  --from-literal=metrics-password='{postgres_metrics_password}'
```

Keycloak:

```
kubectl -n {keycloak_namespace} create secret generic selena-keycloak-admin \
  --from-literal=admin-user='{keycloak_admin_username}' \
  --from-literal=admin-password='{keycloak_admin_password}'

kubectl -n {keycloak_namespace} create secret generic selena-keycloak-db \
  --from-literal=password='{postgres_selena_user_password}'
```

OpenLDAP test/demo:

```
kubectl -n {openldap_namespace} create secret generic selena-openldap-auth \
  --from-literal=LDAP_ADMIN_USERNAME='{ldap_admin_username}' \
  --from-literal=LDAP_ADMIN_PASSWORD='{ldap_admin_password}' \
  --from-literal=LDAP_CONFIG_ADMIN_PASSWORD='{ldap_config_admin_password}'
```

MinIO tenant:

```
kubectl -n {minio_namespace} create secret generic selena-minio-env-configuration \
  --from-literal=config.env='export MINIO_ROOT_USER="{minio_root_user}"
export MINIO_ROOT_PASSWORD="{minio_root_password}"'

kubectl -n {minio_namespace} create secret generic selena-minio-user \
  --from-literal=CONSOLE_ACCESS_KEY='{minio_access_key}' \
  --from-literal=CONSOLE_SECRET_KEY='{minio_secret_key}'
```

`config.env` с `export MINIO_ROOT_USER=...` - это формат, который ожидает MinIO Tenant chart. Это не shell-переменные для выполнения команд.

Lakekeeper:

```
kubectl -n {lakekeeper_namespace} create secret generic selenalakekeeperdb \
  --from-literal=postgresql-user='selenalakekeeper' \
  --from-literal=postgresql-password='{postgres_selenalakekeeper_password}'

kubectl -n {lakekeeper_namespace} create secret generic selenalakekeeperpostgresencryption \
  --from-literal=encryptionKey='{lakekeeper_secret_encryption_key}'
```

5.3 CM Secret

CM Secret содержит только секретные значения. Несекретные URLs и mappings лежат в CM values.

Не создавайте `selenalakekeeper-secrets` до подготовки CM database и Keycloak clients:

`CM_DB_URL`, `CM_DB_USER`, `CM_DB_PASSWORD`, `CM_KEYCLOAK_LOGIN_CLIENT_SECRET` и

`CM_KEYCLOAK_ADMIN_CLIENT_SECRET` должны быть уже известны. Если используете

bundled PostgreSQL, сначала создайте `selenalakekeeper-postgres-auth`, установите PostgreSQL chart, подтвердите service DNS и database credentials, затем создавайте CM Secret.

Сгенерируйте 32-byte encryption key:

```
openssl rand -base64 32
```

Скопируйте результат в `{cm_encryption_key_base64}`.

Создайте Secret:

```
kubectl -n {selena_namespace} create secret generic selena-cm-secrets \
  --from-literal=CM_DB_URL='{cm_db_jdbc_url}' \
  --from-literal=CM_DB_USER='{cm_db_user}' \
  --from-literal=CM_DB_PASSWORD='{cm_db_password}' \
  --from-
literal=CM_KEYCLOAK_LOGIN_CLIENT_SECRET='{selena_engine_client_secret}' \
  --from-
literal=CM_KEYCLOAK_ADMIN_CLIENT_SECRET='{selena_cm_admin_client_secret}' \
  --from-literal=CM_ENCRYPTION_KEY_BASE64='{cm_encryption_key_base64}' \
  --from-literal=CM_ENCRYPTION_KEY_ID='main'
```

Key	Откуда взять
CM_DB_URL	JDBC URL PostgreSQL database CM.
CM_DB_USER / CM_DB_PASSWORD	CM database credentials.
CM_KEYCLOAK_LOGIN_CLIENT_SECRET	Secret Keycloak client <code>selena-engine</code> .
CM_KEYCLOAK_ADMIN_CLIENT_SECRET	Secret Keycloak client <code>selena-cluster-manager-api</code> .
CM_ENCRYPTION_KEY_BASE64	Результат <code>openssl rand -base64 32</code> .
CM_ENCRYPTION_KEY_ID	Stable key id, например <code>main</code> .

5.4 IDE Secret

Создайте Secret до Engine `POST /engine/cluster:bootstrap-auth-rbac`, даже если IDE chart будет установлен позже. Поле `IDE_SELENA_IMPERSONATOR_PASSWORD` сначала пустое: CM заполнит его во время Engine bootstrap.

```
kubectl -n {selena_namespace} create secret generic selena-ide-secrets \
  --from-literal=FLASK_SECRET_KEY='{ide_flask_secret_key}' \
  --from-literal=DATABASE_CONN='{ide_database_conn}' \
  --from-literal=IDE_KEYCLOAK_CLIENT_SECRET='{selena_ide_client_secret}' \
  --from-literal=IDE_TOKEN_ENCRYPTION_KEY='{ide_token_encryption_key}' \
  --from-
literal=IDE_SERVICE_CLIENT_SECRET='{selena_ide_service_client_secret}' \
  --from-literal=IDE_SELENA_IMPERSONATOR_PASSWORD='' \
  --from-literal=OPENAI_API_KEY='{optional_ide_openai_api_key}'
```

`IDE_TOKEN_ENCRYPTION_KEY` должен быть стабильным между pod restarts. Для генерации можно использовать `openssl rand -base64 32`.

5.5 AI Secret

Создайте Secret до Engine `POST /engine/cluster:bootstrap-auth-rbac`, даже если AI chart будет установлен позже. Поле `MCP_SELENA_IMPERSONATOR_PASSWORD` сначала пустое: CM заполнит его во время Engine bootstrap.

```
kubectl -n {selena_namespace} create secret generic selena-ai-secrets \
  --from-literal=AI_BACKEND_DATABASE_URL='{ai_backend_database_url}' \
  --from-literal=AI_BACKEND_LLM_API_KEY='{llm_api_key}' \
  --from-literal=AI_BACKEND_AUTH_CLIENT_SECRET='{selena_ai_client_secret}' \
  \
  --from-
literal=AI_BACKEND_AUTH_TOKEN_ENCRYPTION_KEY='{ai_token_encryption_key}' \
  --from-literal=MCP_SELENA_IMPERSONATOR_PASSWORD=' '
```

`AI_BACKEND_LLM_API_KEY` должен быть реальным OpenAI/OpenRouter-compatible key. Placeholder даст зеленый Kubernetes rollout и даже health, но live chat вернет LLM authorization error.

5.6 Grafana OAuth Secret

Если Grafana ставится вместе с Selena:

```
kubectl -n {monitoring_namespace} create secret generic selena-grafana-
admin \
  --from-literal=admin-user='{grafana_admin_user}' \
  --from-literal=admin-password='{grafana_admin_password}'

kubectl -n {monitoring_namespace} create secret generic selena-grafana-
keycloak-client \
  --from-literal=client-secret='{grafana_client_secret}'
```

5.7 Минимальные values

Используйте файлы:

```
docs-external/values-minimal/selena-operator-values.yaml
docs-external/values-minimal/selena-cm-values.yaml
docs-external/values-minimal/selena-ide-values.yaml
docs-external/values-minimal/selena-ai-values.yaml
```

Для bundled external dependencies рядом лежат такие же короткие override-файлы:

```
docs-external/values-minimal/postgres-values.yaml
docs-external/values-minimal/keycloak-values.yaml
docs-external/values-minimal/openldap-values.yaml
docs-external/values-minimal/kube-prometheus-stack-values.yaml
docs-external/values-minimal/minio-operator-values.yaml
docs-external/values-minimal/minio-tenant-values.yaml
docs-external/values-minimal/lakekeeper-values.yaml
docs-external/values-minimal/vault-values.yaml
```

Скопируйте их в customer deployment repo и замените `{...}` placeholders.

5.8 Свойства Selena operator values

В minimal values оставлены только image settings. RBAC, ServiceAccount, imagePullPolicy и metrics имеют product defaults в chart.

Свойство	Значение	Зачем нужно
<code>selenaOperator.image.repository</code>	<code>hub.selena.sh/selena/operator</code> или совместимый mirror	Repository Selena operator image.
<code>selenaOperator.image.tag</code>	<code>1.11.4</code> или совместимый tag	Версия operator. Должна быть согласована с Selena Engine CRD support matrix.
<code>selenaOperator.imagePullSecrets[0].name</code>	<code>selena-registry</code>	Secret для private registry. Если registry public, уберите этот список.

5.9 Свойства Selena CM values

В minimal values оставлены URL-ы стенда, images, secrets и business mappings. HA replicas, ports, RBAC, refresh schedule, lock timings, client IDs, audience, group claim и metrics имеют product defaults в CM/chart. В chart `values.yaml` обязательные стендовые Keycloak URL-ы перечислены с пустыми defaults и должны переопределяться в customer-specific values.

Свойство	Значение	Зачем нужно
<code>image.repository</code>	<code>{registry}/selena-cm</code>	Repository CM image. Для air-gapped окружений укажите внутренний registry.
<code>image.tag</code>	<code>{selena_cm_tag}</code>	Версия CM image. Фиксируйте конкретный release tag.
<code>imagePullSecrets[0].name</code>	<code>selena-registry</code>	Secret для private registry. Если registry public, уберите этот список.
<code>config.env.CM_KEYCLOAK_TOKEN_URI</code>	<code>{keycloak_internal_url}/realms/selena/protocol/openid-connect/token</code>	Internal token endpoint для password login и service calls.
<code>config.env.CM_KEYCLOAK_LOGOUT_URI</code>	<code>{keycloak_public_url}/realms/selena/protocol/openid-connect/logout</code>	Browser logout endpoint.
<code>config.env.CM_KEYCLOAK_JWKS_URI</code>	<code>{keycloak_internal_url}/realms/selena/protocol/openid-connect/certs</code>	JWKS для проверки JWT.
<code>config.env.CM_KEYCLOAK_ISSUER_URI</code>	<code>{keycloak_public_url}/realms/selena</code>	Должен точно совпадать с token claim <code>iss</code> .
<code>config.env.CM_KEYCLOAK_ADMIN_TOKEN_URI</code>	<code>{keycloak_internal_url}/realms/selena/protocol/openid-connect/token</code>	Internal token endpoint для service account CM Admin API client.
<code>config.env.CM_KEYCLOAK_ADMIN_BASE_URI</code>	<code>{keycloak_internal_url}/admin/realms/selena</code>	Internal Keycloak Admin API base URL, откуда CM читает users/groups.
<code>extraEnv[0].name</code>	<code>SPRING_APPLICATION_JSON</code>	Передаёт structured config без сложного env escaping.
<code>config.env.CM_SELENA_EXTERNAL_GROUP_FALLBACK_STRATEGY</code>	<code>FULL_PATH</code>	Как CM строит Selena external group для Keycloak groups без explicit mapping. <code>FULL_PATH</code> делает <code>/cm/admins -> cm_admins</code> ; <code>NONE</code> включает strict allow-list только через mappings.
<code>extraEnv[0].value.cm.identity.external-groups.mappings</code>	Keycloak path -> external group	Optional override для generated external group names. Используйте, когда нужно сохранить legacy short names или избежать collision.
<code>extraEnv[0].value.cm.security.group-role-mappings</code>	Keycloak path -> CM role	Какие Keycloak groups дают доступ к CM API.
<code>secrets.existingSecret</code>	<code>selena-cm-secrets</code>	Откуда CM берёт secret env vars.
<code>secrets.envKeys.CMDB_URL</code>	Secret key	JDBC URL CM PostgreSQL.
<code>secrets.envKeys.CMDB_USER</code>	Secret key	DB user CM PostgreSQL.
<code>secrets.envKeys.CMDB_PASSWORD</code>	Secret key	DB password CM PostgreSQL.

Свойство	Значение	Зачем нужно
<code>secrets.envKeys.CM_KEYCLOAK_LOGIN_CLIENT_SECRET</code>	Secret key	Secret Keycloak client <code>selena-engine</code> .
<code>secrets.envKeys.CM_KEYCLOAK_ADMIN_CLIENT_SECRET</code>	Secret key	Secret Keycloak client <code>selena-cluster-manager-api</code> .
<code>secrets.envKeys.CM_ENCRYPTION_KEY_BASE64</code>	Secret key	Stable 32-byte base64 key для шифрования CM secrets в DB.
<code>secrets.envKeys.CM_ENCRYPTION_KEY_ID</code>	Secret key	Идентификатор активного encryption key, например <code>main</code> .

5.10 Свойства Selena IDE values

В `minimal values` оставлены только `public/internal URL`-ы, `Engine FE host`, `images` и `secrets`. `Replicas`, `ports`, `auth backend`, `client IDs`, `service-to-service CM URL`, `Querybook defaults` и `metrics` имеют `product defaults` в `chart`. В `chart values.yaml` обязательные стендовые URL-ы и `Engine FE host` перечислены с пустыми `defaults` и должны переопределяться в `customer-specific values`.

Свойство	Значение	Зачем нужно
<code>image.repository</code>	<code>{registry}/selena-ide</code>	Repository IDE image.
<code>image.tag</code>	<code>{selena_ide_tag}</code>	Версия IDE image.
<code>imagePullSecrets[0].name</code>	<code>selena-registry</code>	Secret для private registry. Если registry public, уберите этот список.
<code>config.env.PUBLIC_URL</code>	<code>{ide_public_url}</code>	Должен совпадать с Keycloak redirect URI.
<code>config.env.IDE_KEYCLOAK_ISSUER_URL</code>	<code>{keycloak_public_url}/realms/selena</code>	Issuer, который IDE ожидает в tokens.
<code>config.env.IDE_KEYCLOAK_AUTHORIZATION_URL</code>	<code>{keycloak_public_url}/realms/selena/protocol/openid-connect/auth</code>	Browser authorization endpoint.
<code>config.env.IDE_KEYCLOAK_TOKEN_URL</code>	<code>{keycloak_internal_url}/realms/selena/protocol/openid-connect/token</code>	Internal token endpoint для IDE.
<code>config.env.IDE_KEYCLOAK_JWKS_URL</code>	<code>{keycloak_internal_url}/realms/selena/protocol/openid-connect/certs</code>	JWKS endpoint для проверки tokens.
<code>config.env.IDE_KEYCLOAK_LOGOUT_URL</code>	<code>{keycloak_public_url}/realms/selena/protocol/openid-connect/logout</code>	Keycloak logout endpoint для browser logout.

Свойство	Значение	Зачем нужно
	<code>connect/logout</code>	
<code>config.env.IDE_SERVICE_TOKEN_URL</code>	<code>{keycloak_internal_url}/realms/selena/protocol/openid-connect/token</code>	Token endpoint для service client <code>selena-ide-service</code> .
<code>config.env.IDE_SERVICE_HOST</code>	<code>selena-prod-fe-service.{engine_namespace}.svc.cluster.local</code>	Internal FE service DNS для SQL execution через IDE. Должен совпадать с Engine <code>spec.name</code> .
<code>config.env.AI_AGENT_URL</code>	<code>{ai_public_url}</code>	Browser-reachable URL, который IDE открывает во встроенном AI iframe. Это не обязательно отдельный host; можно использовать path под IDE domain.
<code>secrets.existingSecret</code>	<code>selena-ide-secrets</code>	IDE secrets, включая password, который заполнит Engine bootstrap.
<code>secrets.envKeys.FLASK_SECRET_KEY</code>	Secret key	Flask/session signing key IDE.
<code>secrets.envKeys.DATABASE_CONN</code>	Secret key	PostgreSQL connection string IDE.
<code>secrets.envKeys.IDE_KEYCLOAK_CLIENT_SECRET</code>	Secret key	Secret Keycloak browser client <code>selena-ide</code> .
<code>secrets.envKeys.IDE_TOKEN_ENCRYPTION_KEY</code>	Secret key	Stable key для шифрования IDE tokens/sessions.
<code>secrets.envKeys.IDE_SELENA_IMPERSONATOR_PASSWORD</code>	Secret key	Password <code>ide_impersonator_all</code> ; CM заполняет после Engine bootstrap.
<code>secrets.envKeys.IDE_SERVICE_CLIENT_SECRET</code>	Secret key	Secret service client <code>selena-ide-service</code> .
<code>secrets.envKeys.OPENAI_API_KEY</code>	Secret key	Optional key для встроенного IDE assistant. Если не используется, оставьте customer-approved dummy или уберите env по политике chart.

5.11 Свойства Selena AI values

В `minimal values` оставлены LLM provider, browser-facing URLs, Keycloak URLs, Engine FE host, images и secrets. Replicas, ports, service types, sticky MCP affinity, token claims, client IDs, audience, metrics и default backend/MCP ports имеют product defaults в chart. В `chart values.yaml` обязательные стендовые URL-ы тоже перечислены, но оставлены пустыми: их надо задать в `customer-specific values`, потому что public issuer/browser URL и internal service URL отличаются для каждого Kubernetes стенда.

Свойство	Значение	Зачем нужно
<code>imagePullSecrets[0].name</code>	<code>selenia-registry</code>	Secret для private registry. Если registry public, уберите этот список.
<code>components.backend.image.repository</code>	<code>{registry}/selenia-ai-backend</code>	Repository AI backend image.
<code>components.backend.image.tag</code>	<code>{selenia_ai_backend_tag}</code>	Версия AI backend image.
<code>components.backend.env.AI_BACKEND_LLM_BASE_URL</code>	<code>{llm_base_url}</code>	Provider endpoint, например OpenAI или OpenRouter compatible URL.
<code>components.backend.env.AI_BACKEND_LLM_MODEL</code>	<code>{llm_model}</code>	Модель для AI Agent.
<code>components.backend.env.AI_BACKEND_AUTH_ISSUER_URL</code>	<code>{keycloak_public_url}/realms/selenia</code>	Issuer, который AI backend ожидает в tokens.
<code>components.backend.env.AI_BACKEND_AUTH_AUTHORIZATION_URL</code>	<code>{keycloak_public_url}/realms/selenia/protocol/openid-connect/auth</code>	Browser authorization endpoint для AI auth flow.
<code>components.backend.env.AI_BACKEND_AUTH_TOKEN_URL</code>	<code>{keycloak_internal_url}/realms/selenia/protocol/openid-connect/token</code>	Internal token endpoint.
<code>components.backend.env.AI_BACKEND_AUTH_JWKS_URL</code>	<code>{keycloak_internal_url}/realms/selenia/protocol/openid-connect/certs</code>	JWKS endpoint для token validation.
<code>components.backend.env.AI_BACKEND_AUTH_LOGOUT_URL</code>	<code>{keycloak_public_url}/realms/selenia/protocol/openid-connect/logout</code>	Logout endpoint Keycloak.
<code>components.backend.env.AI_BACKEND_AUTH_REDIRECT_URI</code>	<code>{ai_public_url}/api/v1/auth/callback</code>	Redirect URI, который должен быть указан в Keycloak client. <code>{ai_public_url}</code> должен быть доступен browser-y, даже если AI открывается только внутри IDE iframe.
<code>components.backend.env.AI_BACKEND_AUTH_POST_LOGIN_REDIRECT_URI</code>	<code>{ai_public_url}/</code>	Куда вернуть browser после successful login.
<code>components.backend.env.AI_BACKEND_AUTH_POST_LOGOUT_REDIRECT_URI</code>	<code>{ai_public_url}/</code>	Куда вернуть browser после logout.
<code>components.backend.env.AI_BACKEND_AUTH_FRONTEND_URL</code>	<code>{ai_public_url}</code>	Browser-reachable AI iframe URL для links/cookies.
<code>components.backend.env.AI_BACKEND_CORS_ORIGINS</code>	<code>{ide_public_url}</code>	Browser origin IDE, которому разрешены AI backend calls.
<code>components.backend.secrets.existingSecret</code>	<code>selenia-ai-secrets</code>	Secret backend env vars.

Свойство	Значение	Зачем нужно
<code>components.backend.secret</code> <code>s.envKeys.AI_BACKEND_DATAB</code> <code>ASE_URL</code>	Secret key	PostgreSQL URL AI backend sessions/storage.
<code>components.backend.secret</code> <code>s.envKeys.AI_BACKEND_LLM_A</code> <code>PI_KEY</code>	Secret key	LLM provider API key.
<code>components.backend.secret</code> <code>s.envKeys.AI_BACKEND_AUTH_</code> <code>CLIENT_SECRET</code>	Secret key	Secret Keycloak client <code>selenai</code> .
<code>components.backend.secret</code> <code>s.envKeys.AI_BACKEND_AUTH_</code> <code>TOKEN_ENCRYPTION_KEY</code>	Secret key	Stable key для шифрования AI tokens/sessions.
<code>components.frontend.image.</code> <code>repository</code>	<code>{registry}/selena-ai-fro</code> <code>ntend</code>	Repository AI frontend image.
<code>components.frontend.image.</code> <code>tag</code>	<code>{selena_ai_frontend_tag}</code>	Версия AI frontend image.
<code>components.mcp.image.repos</code> <code>itory</code>	<code>{registry}/selena-mcp</code>	Repository MCP image.
<code>components.mcp.image.tag</code>	<code>{selena_mcp_tag}</code>	Версия MCP image.
<code>components.mcp.env.MCP_SEL</code> <code>ENA_HOST</code>	<code>selenai-prod-fe-service.</code> <code>{engine_namespace}.svc.cl</code> <code>uster.local</code>	Internal FE service DNS. Должен совпадать с Engine <code>spec.n</code> <code>ame</code> .
<code>components.mcp.env.MCP_KEY</code> <code>CLOAK_JWKS_URL</code>	<code>{keycloak_internal_url}/</code> <code>realms/selena/protocol/op</code> <code>enid-connect/certs</code>	JWKS endpoint для проверки tokens, которые приходят в MCP.
<code>components.mcp.env.MCP_KEY</code> <code>CLOAK_ISSUER</code>	<code>{keycloak_public_url}/re</code> <code>alms/selena</code>	Expected issuer.
<code>components.mcp.secrets.exi</code> <code>stingSecret</code>	<code>selenai-secrets</code>	Secret MCP env vars.
<code>components.mcp.secrets.env</code> <code>Keys.MCP_SELENA_IMPERSONAT</code> <code>OR_PASSWORD</code>	Secret key	Password <code>ai_impersonator_all</code> ; CM заполняет после Engine bootstrap.

Перед выбором `{llm_model}` проверьте compatibility конкретного `{selena_ai_backend_tag}` с provider API. Старые AI backend builds используют legacy OpenAI parameter `max_tokens`; модели, которым нужен `max_completion_tokens` (например часть gpt-5/o-series), требуют backend build с обновленным provider adapter. Если такой build еще не выбран, используйте модель, совместимую с `max_tokens`, например `gpt-4o-mini` или approved provider-specific аналог.

5.12 Свойства external chart values

Эти charts являются удобным bundled вариантом. Если у заказчика уже есть production PostgreSQL, Keycloak, LDAP/AD, S3-compatible storage, monitoring или Vault, используйте customer-managed сервис и не ставьте соответствующий chart.

Минимальные external values оставляют только то, что реально нужно задать для работающего bundled стенда. Chart defaults не копируются в файл.

PostgreSQL

Свойство	Значение	Зачем нужно
<code>fullnameOverride</code>	<code>selen-postgres</code>	Стабильное имя Service/StatefulSet для команд и downstream URLs.
<code>auth.username</code>	<code>selen</code>	Application DB user, которому init script выдает доступ к Selen databases.
<code>auth.existingSecret</code>	<code>selen-postgres-auth</code>	Пароли берутся из Secret, а не из values.
<code>primary.persistence.storageClass</code>	<code>{storage_class}</code>	StorageClass для PostgreSQL PVC.
<code>primary.persistence.size</code>	<code>64Gi</code>	Размер bundled PostgreSQL volume.
<code>primary.initdb.scripts</code>	create Selen databases	Создает <code>keycloak</code> , <code>lakekeeper</code> , <code>selen_cm</code> , <code>selen_id</code> , <code>selen_ai</code> и выдает grants user <code>selen</code> .

Keycloak

Свойство	Значение	Зачем нужно
<code>fullnameOverride</code>	<code>selen-keycloak</code>	Стабильное имя Keycloak resources.
<code>args --hostname</code>	<code>{keycloak_public_url}</code>	Public URL, который попадет в issuer, redirect и logout flows.
<code>http.relativePath</code>	<code>/</code>	Realm URLs имеют вид <code>{keycloak_public_url}/realms/selen</code> , без legacy <code>/auth</code> .
<code>proxy.mode</code> / <code>proxy.http.enabled</code>	<code>xforwarded</code> / <code>true</code>	Keycloak корректно работает за Ingress/LB, а внутри кластера слушает HTTP.
<code>service.httpsPort</code>	<code>null</code>	Не создает лишний HTTPS service port в minimal profile.
<code>extraEnv.KC_BOOTSTRAP_ADMIN_*</code>	Secret <code>selen-keycloak-admin</code>	Bootstrap admin берется из Secret.
<code>database.*</code>	PostgreSQL connection	Keycloak использует PostgreSQL, а не dev-file/dev-mem storage.

Если меняете Keycloak hostname после первичной установки, меняйте именно `args[]` с `-hostname=...`, а не container `command`. `command` должен оставаться `entrypoint chart/image`; подмена `command` на `start` приведет к ошибке container runtime `executable file not found`.

OpenLDAP

OpenLDAP приведен в качестве примера для test/demo стенда. В production обычно используйте существующий LDAP/AD заказчика и подключайте его к Keycloak federation.

Свойство	Значение	Зачем нужно
<code>fullnameOverride</code>	<code>selena-openldap</code>	Стабильное имя LDAP Service.
<code>global.ldapDomain</code>	<code>{ldap_domain}</code>	LDAP domain для demo directory.
<code>global.existingSecret</code>	<code>selena-openldap</code> <code>-auth</code>	Admin/config passwords берутся из Secret.
<code>replicaCount</code>	<code>1</code>	Минимальный demo LDAP.
<code>replication.enabled</code>	<code>false</code>	Упрощает test/demo установку.
<code>service.enableSslLdapPort</code>	<code>false</code>	LDAPS не публикуется в demo profile.
<code>env.LDAP_ENABLE_TLS</code>	<code>no</code>	Demo LDAP работает только внутри cluster; production LDAP/AD должен иметь свой TLS.
<code>persistence.storageClass</code> / <code>size</code>	<code>{storage_class}</code> , <code>16Gi</code>	PVC для demo directory.
<code>ltb-passwd.enabled</code> / <code>phpldapa</code> <code>admin.enabled</code>	<code>false</code>	Не ставим лишние LDAP UI/tools.

kube-prometheus-stack и Grafana

Свойство	Значение	Зачем нужно
<code>fullnameOverride</code>	<code>selena-monitoring</code>	Стабильные names monitoring resources.
<code>crds.enabled</code>	<code>true</code>	Устанавливает Prometheus Operator CRD до Selena charts с ServiceMonitor.
<code>prometheus.prometheusSpec.*SelectorNilUsesHelmValues</code>	<code>false</code>	Prometheus видит ServiceMonitor/PodMonitor/Rules из Selena namespaces.
<code>prometheus.prometheusSpec.*NamespaceSelector</code>	<code>{}</code>	Разрешает discovery во всех namespaces.
<code>prometheus.prometheusSpec.retention</code>	<code>15d</code>	Сколько хранить metrics history.
<code>prometheus.prometheusSpec.storageSpec</code>	<code>{storage_class}</code> , <code>64Gi</code>	Persistent storage Prometheus.

Свойство	Значение	Зачем нужно
<code>alertmanager.alertmanagerSpec.storage</code>	<code>{storage_class}, 8Gi</code>	Persistent storage Alertmanager.
<code>grafana.admin.existingSecret</code>	<code>selenium-grafana-admin</code>	Grafana admin credentials берутся из Secret.
<code>grafana.envValueFrom.GF_AUTH_GENERIC_OAUTH_CLIENT_SECRET</code>	<code>selenium-grafana-keycloak-client</code>	Grafana OAuth client secret берется из Secret.
<code>grafana.persistence</code>	<code>{storage_class}, 16Gi</code>	Сохраняет Grafana state между restart-ами.
<code>grafana.grafana.ini.server.root_url</code>	<code>{grafana_public_url}/</code>	Public URL Grafana, совпадающий с Keycloak redirects/logout.
<code>grafana.grafana.ini.auth.generic_oauth.*</code>	Keycloak endpoints	Включает login через Keycloak client <code>grafana</code> .
<code>grafana.grafana.ini.auth.generic_oauth.role_attribute_path</code>	<code>grafana_admin / grafana_viewer</code>	Grafana role берется из Keycloak realm roles.
<code>grafana.grafana.ini.auth.generic_oauth.role_attribute_strict</code>	<code>true</code>	Пользователь без Grafana role не входит в Grafana.
<code>grafana.grafana.ini.auth.generic_oauth.signout_redirect_url</code>	Keycloak logout URL with <code>client_id=grafana</code>	Logout завершает Keycloak session и не падает на <code>id_token_hint</code> .
<code>seleniumDashboards.items</code>	Selena dashboards	Chart создает dashboard ConfigMaps для Grafana sidecar.

`$_env{GF_AUTH_GENERIC_OAUTH_CLIENT_SECRET}` в Grafana config - это синтаксис Grafana, не shell variable. Его не заменяйте на plaintext secret.

MinIO

Свойство	Значение	Зачем нужно
<code>minio-operator-values.yaml</code>	comment-only	MinIO Operator работает на upstream defaults. Overrides нужны только для private registry/mirror.
<code>tenant.name</code>	<code>selenium-minio</code>	Имя MinIO Tenant.
<code>tenant.configSecret.existingSecret</code>	<code>true</code>	Root credentials берутся из заранее созданного Secret.
<code>tenant.pools[0].servers / volumesPerServer</code>	<code>1 / 1</code>	Минимальный single-server tenant.
<code>tenant.pools[0].size</code>	<code>64Gi</code>	Размер MinIO PVC.
<code>tenant.pools[0].storageClassName</code>	<code>{storage_class}</code>	StorageClass MinIO PVC.
<code>tenant.buckets</code>	<code>selenium, warehouse</code>	Buckets, которые нужны lakehouse сценариям.

Свойство	Значение	Зачем нужно
<code>tenant.users</code>	<code>selena-minio</code> <code>-user</code>	Secret с access key/secret key для application user.

Lakekeeper

Свойство	Значение	Зачем нужно
<code>fullnameOverride</code>	<code>selena-lakekeeper</code>	Стабильное имя Lakekeeper Service.
<code>postgresql.enabled</code>	<code>false</code>	Embedded DB выключена, используется общий PostgreSQL.
<code>externalDatabase.*</code>	PostgreSQL connection + Secrets	Lakekeeper database credentials не лежат в values.
<code>secretBackend.type</code>	<code>Postgres</code>	Lakekeeper secrets хранит в PostgreSQL.
<code>secretBackend.postgres.encryption</code> <code>KeySecret</code>	<code>selena-lakekeeper-postgres-en</code> <code>ryption</code>	Stable encryption key хранится в Secret.

Vault

Bundled Vault values - optional bootstrap/example. Для production чаще используется customer-managed Vault или External Secrets.

Свойство	Значение	Зачем нужно
<code>fullnameOverride</code>	<code>selena-vault</code>	Стабильное имя Vault resources.
<code>injector.enabled</code>	<code>false</code>	Minimal profile не включает Vault Agent Injector.
<code>server.dataStorage.storageClass</code>	<code>{storage_class}</code>	StorageClass Vault PVC.
<code>server.dataStorage.size</code>	<code>20Gi</code>	Размер Vault PVC.
<code>ui.enabled</code>	<code>true</code>	Создает Service для Vault UI/API.

5.13 Проверка values до установки

Команды ниже проверяют customer-specific values files, которые вы получили копированием из `docs-external/values-minimal/` и заменой placeholders. Не устанавливайте chart-ы напрямую с `values-minimal`: это шаблоны, а не готовый deployment profile.

```
helm lint charts/internal/selena-operator/chart \
  -f {selena_operator_values}

helm lint charts/internal/selena-cm/chart \
  -f {selena_cm_values}

helm lint charts/internal/selena-ide/chart \
  -f {selena_ide_values}

helm lint charts/internal/selena-ai/chart \
  -f {selena_ai_values}

helm lint charts/external/postgres/chart \
  -f {postgres_values}

helm lint charts/external/keycloak/chart \
  -f {keycloak_values}

helm lint charts/external/kube-prometheus-stack/chart \
  -f {monitoring_values}

helm lint charts/external/minio/operator/chart \
  -f {minio_operator_values}

helm lint charts/external/minio/tenant/chart \
  -f {minio_tenant_values}

helm lint charts/external/lakekeeper/chart \
  -f {lakekeeper_values}

helm lint charts/external/openldap/chart \
  -f {openldap_values}

helm lint charts/external/vault/chart \
  -f {vault_values}
```

Render-проверка:

```
helm template selenacm charts/internal/selenacm/chart \
  -n {selenamespace} \
  -f {selenacm_values} \
  >/tmp/selenacm-rendered.yaml

rg -n "password|secret|token|client-secret" /tmp/selenacm-rendered.yaml
```

В rendered manifests не должно быть plaintext secret values. Допустимы только имена Secrets и env var names.

6 Установка базовой платформы Selena

В этом документе устанавливаются только компоненты Selena, которые нужны до Engine: Selena operator и Selena Cluster Manager. External dependencies, включая monitoring/Grafana, устанавливаются или подтверждаются раньше по Установка внешних зависимостей. IDE, AI backend/frontend и MCP устанавливаются позже, после Engine READY и POST /engine/cluster:bootstrap-auth-rbac.

Рабочая директория для команд:

```
cd {selena_docs_repo}/deploy/k8s
```

6.1 Проверка входных данных

Перед Helm install должны быть готовы:

- namespaces {selena_namespace}, {engine_namespace} и {operator_namespace};
- image pull Secret в {selena_namespace} и {engine_namespace};
- CM database уже создана и доступна из cluster;
- external dependencies установлены или подтверждены по Установка внешних зависимостей;
- Keycloak realm, clients и admin user из Keycloak, LDAP и OAuth;
- Kubernetes Secret selena-cm-secrets, созданный после того, как стали известны CM DB credentials и Keycloak client secrets;
- customer-specific values, созданные из минимальных шаблонов в Values, Secrets и Images.

В командах ниже:

Placeholder	Что это
{selena_operator_values}	Ваш values file для operator, скопированный из docs-external/values-minimal/selena-operator-values.yaml и заполненный под стенд.
{selena_cm_values}	Ваш values file для CM, скопированный из docs-external/values-minimal/selena-cm-values.yaml и заполненный под стенд.

Проверка:

```
kubectl get namespace {selena_namespace} {engine_namespace}
{operator_namespace}
kubectl -n {selena_namespace} get secret selena-registry selena-cm-secrets
kubectl -n {engine_namespace} get secret selena-registry

helm lint charts/internal/selena-operator/chart \
  -f {selena_operator_values}

helm lint charts/internal/selena-cm/chart \
  -f {selena_cm_values}
```

Если в values включены ServiceMonitor/PodMonitor, до установки operator/CM должны существовать Prometheus Operator CRD:

```
kubectl get crd servicemonitors.monitoring.coreos.com
podmonitors.monitoring.coreos.com
```

Если CRD нет, выключите ServiceMonitor/PodMonitor в values или сначала установите monitoring.

Что проверить в values перед установкой без bundled monitoring:

Chart	Values, которые должны оставаться выключенными, если CRD нет
selena-cm	metrics.serviceMonitor.enabled: false, metrics.podMonitor.enabled: false
selena-operator	metrics.serviceMonitor.enabled: false

В `docs-external/values-minimal/` эти параметры не переопределены, поэтому используются безопасные defaults chart-ов: ServiceMonitor/PodMonitor не создаются. Metrics endpoint при этом не выключается; platform monitoring может scrape'ить его своим способом, если это требуется.

6.2 Selena operator

Operator нужен CM для создания и управления Selena Engine CR. Установите CRD до Helm release:

```
kubectl apply -f charts/internal/selena-operator/chart/crds/
```

Установка:

```
helm upgrade --install selenator \
  charts/internal/selenator/chart \
  -n {operator_namespace} \
  --create-namespace \
  -f {selenator_values} \
  --wait --timeout 10m
```

Проверка:

```
kubectl -n {operator_namespace} rollout status deploy/selenator
kubectl -n {operator_namespace} get pods -o wide
helm status selenator -n {operator_namespace}
```

Критерий готовности: operator pod `Running`, rollout complete, CRD применены.

6.3 Selenator Cluster Manager

СМ должен стартовать до Engine, потому что Engine устанавливается только через СМ API.

Установка:

```
helm upgrade --install selenator \
  charts/internal/selenator/chart \
  -n {selenator_namespace} \
  --create-namespace \
  -f {selenator_values} \
  --wait --timeout 10m
```

Проверка Kubernetes:

```
kubectl -n {selenator_namespace} rollout status deploy/selenator
kubectl -n {selenator_namespace} get pods -l
app.kubernetes.io/instance=selenator -o wide
kubectl -n {selenator_namespace} get svc selenator -o wide
helm status selenator -n {selenator_namespace}
```

Проверка HTTP:

```
curl -fsS {cm_url}/actuator/health | jq
```

Ожидаемый ответ содержит:

```
{  
  "status": "UP"  
}
```

6.4 Первый login в CM

Выполните login под Keycloak user, который состоит в группе, mapped на `cm_admin`, например `/cm/admins`.

```
curl -fsS -X POST {cm_url}/auth/login \  
  -H 'Content-Type: application/json' \  
  -d '{"username":"{cm_admin_username}","password":"' \  
  {cm_admin_web_password}"}' \  
  | jq
```

Скопируйте поле `.accessToken` из ответа. Дальше в документации оно называется `{cm_admin_access_token}`.

Проверьте, что CM видит роль admin:

```
curl -fsS {cm_url}/auth/me \  
  -H 'Authorization: Bearer {cm_admin_access_token}' \  
  | jq
```

Ожидаемо:

- `username` равен `{cm_admin_username}`;
- `cmRoles` содержит `cm_admin`.

Если login не работает, проверьте:

Что проверить	Где искать
Keycloak issuer	<code>CM_KEYCLOAK_ISSUER_URI</code> в CM values должен совпадать с token claim <code>iss</code> .
Login client secret	Kubernetes Secret <code>selena-cm-secrets</code> , key <code>CM_KEYCLOAK_LOGIN_CLIENT_SECRET</code> .
Username claim	В token должен быть <code>preferred_username</code> .
Groups claim	В token должен быть <code>groups</code> с full group paths.
Audience	В token должен быть <code>selena-engine</code> , а CM должен ждать тот же audience.

6.5 License

License gate защищает рабочие CM endpoints. Health, login, OpenAPI и license endpoints остаются доступны.

Проверить текущий статус:

```
curl -fsS {cm_url}/license \
  -H 'Authorization: Bearer {cm_admin_access_token}' \
  | jq
```

Загрузить license:

```
curl -fsS -X POST {cm_url}/license \
  -H 'Authorization: Bearer {cm_admin_access_token}' \
  -F 'file=@{license_file_path}' \
  | jq
```

Проверить после загрузки:

```
curl -fsS {cm_url}/license \
  -H 'Authorization: Bearer {cm_admin_access_token}' \
  | jq
```

Критерий готовности: `status` равен `ACTIVE`, а `cpu0veruse` равен `false`. Любые `NO_LICENSE`, `CORRUPTED`, `NOT_ACTIVE` или `EXPIRED` блокируют рабочие CM endpoints.

6.6 Что не устанавливается на этом шаге

На этом этапе не устанавливайте:

- `selenide`;
- `selenide-ai`;
- `selenide-ai-mcp`.

Причина: Engine еще не создан, а runtime credentials для IDE/AI появятся только после `POST /engine/cluster:bootstrap-auth-rbac`. Ранний deploy обычно дает ожидаемые `Pending/Unready` состояния и усложняет проверку.

Следующий документ: Engine cluster, bootstrap и storage.

7 Engine cluster, bootstrap и storage

Selena Engine устанавливается только через Selena Cluster Manager API. Не создавайте Engine StatefulSet/CR вручную и не масштабируйте их напрямую в штатном режиме.

Порядок:

1. Сохранить Engine spec в CM.
2. Запустить install operation.
3. Дождаться `READY`.
4. Запустить auth/RBAC bootstrap.
5. Проверить bootstrap credentials и FE MySQL endpoint.
6. Настроить storage volume, если deployment использует shared-data/S3.

7.1 Проверка перед Engine install

Перед продолжением должны быть готовы:

- CM health `UP`;
- license загружена;
- Selena operator установлен;
- `{engine_namespace}` создан;
- image pull Secret `selena-registry` создан в `{engine_namespace}`;
- `selena-ide-secrets` и `selena-ai-secrets` созданы в `{selena_namespace}`, если Engine bootstrap должен записать туда IDE/AI impersonator passwords;
- CM admin access token получен из `POST {cm_url}/auth/login`.

Проверки:

```
curl -fsS {cm_url}/actuator/health | jq

curl -fsS {cm_url}/license \
  -H 'Authorization: Bearer {cm_admin_access_token}' \
  | jq

kubectl -n {operator_namespace} get pods -o wide
kubectl -n {engine_namespace} get secret selena-registry
kubectl -n {selena_namespace} get secret selena-ide-secrets selena-ai-secrets
```

7.2 Минимальный Engine spec

Пример ниже описывает `SHARED_DATA` cluster: 3 FE + 3 CN. Это рекомендуемый вариант для deployment, где данные хранятся во внешнем shared storage/S3, а CN используются как compute nodes.

Перед выполнением команды замените placeholders:

Placeholder	Откуда взять
<code>{engine_namespace}</code>	Namespace, где operator будет создавать Engine pods, services и PVC.
<code>{selena_namespace}</code>	Namespace, где установлены Selena application services: CM, IDE, AI. Обычно это namespace CM/IDE/AI chart-ов.
<code>{registry}</code>	Registry/mirror из согласованного deployment profile.
<code>{engine_tag}</code>	Версия Selena Engine image из release/deployment profile.
<code>{storage_class}</code>	Kubernetes StorageClass, выбранный для Engine PVC.

Если нужен `SHARED_NOTHING` cluster с BE, не добавляйте `be` в этот JSON поверх `cn`. Используйте отдельный BE-пример ниже: в одном spec должен быть либо `fe` + `cn`, либо `fe` + `be`.

Важно про `config`: значения из `fe.config`, `cn.config` и `be.config` CM передает в Engine component как есть, построчно в config files нод: `fe.config` -> `fe.conf`, `cn.config` -> `cn.conf`, `be.config` -> `be.conf`. Каждая JSON-пара превращается в строку вида `key = value`. Поэтому указывайте здесь только реальные Selena Engine config keys для соответствующего компонента. Если key написан с ошибкой или value несовместим с Engine, pod может не стартовать или operation завершится ошибкой.

```
cat > /tmp/selena-engine-spec.json <<'JSON'
{
  "spec": {
    "name": "selena-prod",
    "namespace": "{engine_namespace}",
    "deploymentType": "KUBERNETES",
    "clusterMode": "SHARED_DATA",
    "fe": {
      "replicas": 3,
      "image": "{registry}/engine-frontend:{engine_tag}",
      "imagePullPolicy": "IfNotPresent",
      "command": ["/opt/selena/fe_entrypoint.sh"],
      "args": ["${FE_SERVICE_NAME}"],
      "imagePullSecrets": [
        {"name": "selena-registry"}
      ],
      "requests": {"cpu": "4", "memory": "16Gi"},
      "limits": {"cpu": "8", "memory": "32Gi"},
      "config": {
        "run_mode": "shared_data",
        "authentication_chain": "native",
        "meta_dir": "/opt/selena/fe/meta",
        "sys_log_dir": "/opt/selena/fe/log",
        "sys_log_level": "INFO"
      },
    },
    "volumes": [
      {
        "name": "fe-meta",
        "mountPath": "/opt/selena/fe/meta",
        "size": "100Gi",
        "storageClassName": "{storage_class}"
      },
      {
        "name": "fe-log",
        "mountPath": "/opt/selena/fe/log",
        "size": "100Gi",
        "storageClassName": "{storage_class}"
      }
    ],
    "env": [
      {
        "name": "MYSQL_PWD",
        "valueFrom": {
          "secretKeyRef": {
            "name": "selena-engine-admin-sql",
```

```
        "key": "SELENA_ADMIN_SQL_PASSWORD",
        "optional": true
    }
}
},
"service": {
    "type": "ClusterIP"
}
},
"cn": {
    "replicas": 3,
    "image": "{registry}/engine-compute:{engine_tag}",
    "imagePullPolicy": "IfNotPresent",
    "command": ["/opt/selena/cn_entrpoint.sh"],
    "args": ["$(FE_SERVICE_NAME)"],
    "imagePullSecrets": [
        {"name": "selena-registry"}
    ],
    "requests": {"cpu": "8", "memory": "32Gi"},
    "limits": {"cpu": "16", "memory": "64Gi"},
    "config": {
        "storage_root_path": "/data/cn1;/data/cn2",
        "datacache_enable": "true",
        "datacache_mem_size": "10G",
        "datacache_disk_size": "80%",
        "sys_log_dir": "/opt/selena/cn/log",
        "sys_log_level": "INFO"
    },
    "volumes": [
        {
            "name": "cn-cache-1",
            "mountPath": "/data/cn1",
            "size": "500Gi",
            "storageClassName": "{storage_class}"
        },
        {
            "name": "cn-cache-2",
            "mountPath": "/data/cn2",
            "size": "500Gi",
            "storageClassName": "{storage_class}"
        },
        {
            "name": "cn-log",
            "mountPath": "/opt/selena/cn/log",
            "size": "100Gi",
```

```
        "storageClassName": "{storage_class}"
      }
    ],
    "env": [
      {
        "name": "MYSQL_PWD",
        "valueFrom": {
          "secretKeyRef": {
            "name": "selena-engine-admin-sql",
            "key": "SELENA_ADMIN_SQL_PASSWORD",
            "optional": true
          }
        }
      }
    ]
  },
  "bootstrap": {
    "impersonators": [
      {
        "purpose": "CM_IMPERSONATOR",
        "username": "cm_impersonator_all"
      },
      {
        "purpose": "IDE_IMPERSONATOR",
        "username": "ide_impersonator_all",
        "k8sNamespace": "{selena_namespace}",
        "k8sSecretName": "selena-ide-secrets",
        "k8sSecretKey": "IDE_SELENA_IMPERSONATOR_PASSWORD"
      },
      {
        "purpose": "AI_IMPERSONATOR",
        "username": "ai_impersonator_all",
        "k8sNamespace": "{selena_namespace}",
        "k8sSecretName": "selena-ai-secrets",
        "k8sSecretKey": "MCP_SELENA_IMPERSONATOR_PASSWORD"
      }
    ]
  }
}
JSON
```

Как читать и менять этот спес:

JSON path	Значение в примере	Можно менять?	Что означает
<code>spec.name</code>	<code>selena-prod</code>	Да, до install.	Имя Engine cluster. Оно входит в имена Kubernetes resources и Service DNS names. После install меняйте только через delete/recreate.
<code>spec.namespaces</code>	<code>{engine_namespace}</code>	Да, до install.	Namespace, где operator создаст FE/CN или FE/BE pods, services и PVC.
<code>spec.deploymentType</code>	KUBERNETES	Обычно нет.	Сейчас CM поддерживает Kubernetes/operator deployment backend.
<code>spec.clusterMode</code>	SHARED_DATA	Выбирается до install.	SHARED_DATA означает fe + cn. Для SHARED_NOTHING используйте fe + be. После install mode не меняют на месте.
<code>spec.fe</code>	object	Обязательно.	FE component: metadata, SQL frontend, MySQL-compatible endpoint и cluster coordination.
<code>spec.fe.replicas</code>	3	Да, через CM API.	Количество FE pods. Для HA используйте 3. Не масштабируйте FE до 0; CM API защищает от такого сценария.
<code>spec.fe.image</code>	<code>{registry}/engine-frontend:{engine_tag}</code>	Да, для upgrade.	Полный image FE. Registry и tag должны соответствовать release profile.
<code>spec.fe.imagePullPolicy</code>	IfNotPresent	Редко.	Kubernetes pull policy. Для immutable release tags обычно достаточно IfNotPresent.
<code>spec.fe.command / spec.fe.args</code>	<code>/opt/selena/fe_entrypoint.sh, \${FE_SERVICE_NAME}</code>	Не меняйте для Selena images.	Selena images используют <code>/opt/selena</code> , а не upstream <code>/opt/starrock</code> . Явный command защищает от operator defaults, рассчитанных на StarRocks paths. CM может обернуть FE/BE entrypoint в <code>/bin/bash -c ... sleep 5 ...</code> для graceful stop; это ожидаемо.
<code>spec.fe.imagePullSecrets</code>	<code>selena-registry</code>	Да.	Secret для private registry. Secret должен быть создан в <code>{engine_namespace}</code> до install.
<code>spec.fe.requests</code>	4 CPU, 16Gi	Да.	Минимально запрашиваемые ресурсы pod-a. Kubernetes scheduler использует их для размещения.
<code>spec.fe.limits</code>	8 CPU, 32Gi	Да.	Верхняя граница ресурсов container-a. Подбирайте под sizing и policy заказчика.
<code>spec.fe.config.run_mode</code>	shared_data	Менять только вместе с <code>clusterMode</code> .	Engine runtime mode. Для SHARED_DATA должен быть shared_data, для SHARED_NOTHING - shared_nothing.
<code>spec.fe.config.auth</code>	native	Не меняйте.	Selena Engine не подключается напрямую к LDAP/AD. Web login идет через Keycloak, direct SQL использует CM-generated CLI secret.

JSON path	Значение в примере	Можно менять?	Что означает
authentication_chain			
spec.fe.config.metadata_dir	/opt/selena/fe/metadata	Только вместе с volume mount.	Каталог FE metadata. Должен совпадать с fe-meta.mountPath.
spec.fe.config.sys_log_dir	/opt/selena/fe/log	Только вместе с volume mount.	Каталог FE logs. Должен совпадать с fe-log.mountPath.
spec.fe.config.sys_log_level	INFO	Да.	Уровень логирования FE. Для диагностики временно можно повысить, но не оставляйте verbose logging без необходимости.
spec.fe.volumes[].name	fe-meta, fe-log	Планируйте до install.	Имя volume в Engine component. Должно быть уникальным внутри component.
spec.fe.volumes[].mountPath	/opt/selena/fe/.../...	Планируйте до install.	Куда PVC монтируется в container. Config paths должны ссылаться на эти же каталоги.
spec.fe.volumes[].size	100Gi	Зависит от StorageClass.	Размер PVC. Увеличение после install возможно только если StorageClass разрешает expansion. Уменьшение PVC обычно невозможно.
spec.fe.volumes[].storageClassName	{storage_class}	Планируйте до install.	StorageClass для FE PVC. Выбирайте класс с нужной надежностью, IOPS/throughput и backup policy.
spec.fe.env[0].MYSQL_PWD	secret selena-engine-admin-sql/SELENA_ADMIN_SQL_PASSWORD	Да, оставьте в spec.	Нужен FE/CN endpoint после /engine/cluster:bootstrap-auth-rbac, когда root/admin SQL password уже установлен. Без этого post-bootstrap scale-up/restart может не зарегистрировать новый pod в FE. Secret создает CM bootstrap.
spec.fe.service.type	ClusterIP	Да.	Kubernetes Service type для FE. ClusterIP оставляет FE внутри cluster; LoadBalancer публикует endpoint наружу, если это разрешено network/security policy.
spec.cn	object	Только для SHARED_DATA.	CN component: compute nodes и локальный cache для shared-data mode. Не используйте одновременно с be.
spec.cn.replicas	3	Да, через CM API.	Количество CN pods. Scale up/down делайте только через CM API, не через kubectl scale.
spec.cn.image	{registry}/engine-compute:{engine_tag}	Да, для upgrade.	Полный image CN. Tag должен быть совместим с FE tag.

JSON path	Значение в примере	Можно менять?	Что означает
	<code>g}</code>		
<code>spec.cn.command / spec.cn.args</code>	<code>/opt/selena/cn_entrypoint.sh, \$(FE_SERVICE_NAME)</code>	Не меняйте для Selena images.	Явно запускает Selena CN endpoint из <code>/opt/selena</code> , чтобы не зависеть от operator defaults.
<code>spec.cn.requests / spec.cn.limits</code>	<code>8/16 CPU, 32/64Gi</code>	Да.	Ресурсы compute pods. Подбирайте под workload и node size.
<code>spec.cn.config.storage_root_path</code>	<code>/data/cn1;/data/cn2</code>	Только вместе с volumes.	Каталоги локального CN cache/storage. Каждый путь должен иметь matching volume mount.
<code>spec.cn.config.datacache_enabled</code>	<code>true</code>	Да.	Включает local data cache на CN. Для shared-data deployments обычно нужен.
<code>spec.cn.config.datacache_memory_size</code>	<code>10G</code>	Да.	Объем memory cache на CN. Должен соответствовать memory limit.
<code>spec.cn.config.datacache_disk_size</code>	<code>80%</code>	Да.	Доля cache volumes, которую может использовать disk cache.
<code>spec.cn.config.sys_log_dir</code>	<code>/opt/selena/cn/log</code>	Только вместе с volume mount.	Каталог CN logs.
<code>spec.cn.volumes[]</code>	<code>cn-cache-1, cn-cache-2, cn-log</code>	Планируйте до install.	PVC для CN cache и logs. Cache volumes должны соответствовать <code>storage_root_path</code> .
<code>spec.cn.environment[] .MYSQL_PWD</code>	<code>secret selena-engine-admin-sql/SELENA_ADMIN_SQL_PASSWORD</code>	Да, оставьте в spec.	Нужен CN endpoint для регистрации compute node в FE после auth/RBAC bootstrap и для последующих scale-up/restart.
<code>spec.bootstrap.impersonators[]</code>	<code>cm/ide/ai_impersonator_all</code>	Обычно не меняйте.	Service users, которые CM создаст в Selena после install. Они нужны CM, IDE и AI/MCP для выполнения запросов от имени пользователя.

JSON path	Значение в примере	Можно менять?	Что означает
<code>impersonators[].purpose</code>	<code>CM_IMPERSONATOR</code> , <code>ID_IMPERSONATOR</code> , <code>AI_IMPERSONATOR</code>	Нет.	Тип service credential. CM понимает фиксированные purpose names.
<code>impersonators[].username</code>	<code>*_impersonator_al</code>	Обычно нет.	Имя service user в Selena Engine. Если меняете, синхронно обновляйте dependent service config.
<code>impersonators[].kubernetesNamespace</code>	<code>{selena_namespace}</code>	Да.	Namespace Kubernetes Secret, куда CM запишет generated password для IDE/AI после bootstrap.
<code>impersonators[].kubernetesSecretName</code>	<code>selena-ide-secret</code> , <code>selena-ai-secrets</code>	Да, если меняете charts.	Existing Secret, который читают IDE/AI/MCP pods. Имя должно совпадать с values этих chart-ов.
<code>impersonators[].kubernetesSecretKey</code>	<code>IDE_SELENA_IMPERSONATOR_PASSWORD</code> , <code>MCP_SELENA_IMPERSONATOR_PASSWORD</code>	Да, если меняете charts.	Key внутри Secret. Должен совпадать с env var, которую читает соответствующий service.

Пример для `SHARED_NOTHING` cluster: 3 FE + 3 BE. Используйте его вместо `SHARED_DATA` примера, если deployment profile выбран без shared storage/S3.

```
cat > /tmp/selena-engine-spec-shared-nothing.json <<'JSON'
{
  "spec": {
    "name": "selena-prod",
    "namespace": "{engine_namespace}",
    "deploymentType": "KUBERNETES",
    "clusterMode": "SHARED_NOTHING",
    "fe": {
      "replicas": 3,
      "image": "{registry}/engine-frontend:{engine_tag}",
      "imagePullPolicy": "IfNotPresent",
      "command": ["/opt/selena/fe_entrypoint.sh"],
      "args": ["$(FE_SERVICE_NAME)"],
      "imagePullSecrets": [
        {"name": "selena-registry"}
      ],
      "requests": {"cpu": "4", "memory": "16Gi"},
      "limits": {"cpu": "8", "memory": "32Gi"},
      "config": {
        "run_mode": "shared_nothing",
        "authentication_chain": "native",
        "meta_dir": "/opt/selena/fe/meta",
        "sys_log_dir": "/opt/selena/fe/log",
        "sys_log_level": "INFO"
      },
    },
    "volumes": [
      {
        "name": "fe-meta",
        "mountPath": "/opt/selena/fe/meta",
        "size": "100Gi",
        "storageClassName": "{storage_class}"
      },
      {
        "name": "fe-log",
        "mountPath": "/opt/selena/fe/log",
        "size": "100Gi",
        "storageClassName": "{storage_class}"
      }
    ],
    "service": {
      "type": "ClusterIP"
    }
  },
  "be": {
    "replicas": 3,
```

```
"image": "{registry}/engine-backend:{engine_tag}",
"imagePullPolicy": "IfNotPresent",
"command": ["/opt/selena/be_entrypoint.sh"],
"args": ["$(FE_SERVICE_NAME)"],
"imagePullSecrets": [
  {"name": "selena-registry"}
],
"requests": {"cpu": "8", "memory": "32Gi"},
"limits": {"cpu": "16", "memory": "64Gi"},
"config": {
  "storage_root_path": "/data/be1;/data/be2",
  "sys_log_dir": "/opt/selena/be/log",
  "sys_log_level": "INFO"
},
"volumes": [
  {
    "name": "be-storage-1",
    "mountPath": "/data/be1",
    "size": "500Gi",
    "storageClassName": "{storage_class}"
  },
  {
    "name": "be-storage-2",
    "mountPath": "/data/be2",
    "size": "500Gi",
    "storageClassName": "{storage_class}"
  },
  {
    "name": "be-log",
    "mountPath": "/opt/selena/be/log",
    "size": "100Gi",
    "storageClassName": "{storage_class}"
  }
]
},
"bootstrap": {
  "impersonators": [
    {
      "purpose": "CM_IMPERSONATOR",
      "username": "cm_impersonator_all"
    },
    {
      "purpose": "IDE_IMPERSONATOR",
      "username": "ide_impersonator_all",
      "k8sNamespace": "{selena_namespace}",
      "k8sSecretName": "selena-ide-secrets",

```

```
    "k8sSecretKey": "IDE_SELENA_IMPERSONATOR_PASSWORD"
  },
  {
    "purpose": "AI_IMPERSONATOR",
    "username": "ai_impersonator_all",
    "k8sNamespace": "{selena_namespace}",
    "k8sSecretName": "selena-ai-secrets",
    "k8sSecretKey": "MCP_SELENA_IMPERSONATOR_PASSWORD"
  }
]
}
}
JSON
```

Поля BE:

JSON path	Значение в примере	Что означает
<code>spec.be</code>	object	BE component используется только в <code>SHARED_NOTHING</code> . В таком спес не должно быть <code>cn</code> .
<code>spec.be.image</code>	<code>{registry}/engine-backend:{engine_tag}</code>	Полный image BE. Tag должен быть совместим с FE tag.
<code>spec.be.command</code> / <code>spec.be.args</code>	<code>/opt/selena/be_entrypoint.sh, \$(FE_SERVICE_NAME)</code>	Явно запускает Selena BE entrypoint из <code>/opt/selena</code> , чтобы не зависеть от operator defaults.
<code>spec.be.config.storage_root_path</code>	<code>/data/be1;/data/be2</code>	Каталоги постоянного BE storage. Каждый путь должен иметь отдельный matching volume mount.
<code>spec.be.volumes[].mountPath</code>	<code>/data/be1, /data/be2, /opt/selena/be/log</code>	Persistent volumes для BE данных и logs.

Если нужен внешний direct SQL endpoint сразу через FE Service, задайте в `fe.service` тип `LoadBalancer` или используйте approved TCP Ingress/Gateway. Для production ограничьте source ranges firewall/network policy.

7.3 Сохранить спес в CM

Если вы выбрали `SHARED_DATA`, дальше используйте `/tmp/selena-engine-spec.json`. Если выбрали `SHARED_NOTHING`, дальше используйте `/tmp/selena-engine-spec-shared-nothing.json`.

В командах ниже `{engine_spec_file}` - это выбранный файл спес. Перед отправкой проверьте, что это именно тот файл, который вы хотели установить:

```
jq empty {engine_spec_file}
rg -n '\{[a-zA-Z0-9_]+\}' {engine_spec_file}
```

`jq empty` должен завершиться без ошибки. `rg` не должен найти незамененные placeholders.

```
curl -fsS -X PUT {cm_url}/engine/cluster/config \
-H 'Authorization: Bearer {cm_admin_access_token}' \
-H 'Content-Type: application/json' \
--data-binary @{engine_spec_file} \
| jq
```

Проверить сохраненный spec:

```
curl -fsS {cm_url}/engine/cluster/config \
-H 'Authorization: Bearer {cm_admin_access_token}' \
| jq
```

До установки ожидаемый status: `CONFIG_SAVED`.

7.4 Запустить install operation

```
curl -fsS -X POST {cm_url}/engine/cluster:install \
-H 'Authorization: Bearer {cm_admin_access_token}' \
| jq
```

Скопируйте `.operationId` из ответа. Дальше это `{operation_id}`.

Проверить operation:

```
curl -fsS {cm_url}/engine/cluster/operations/{operation_id} \
-H 'Authorization: Bearer {cm_admin_access_token}' \
| jq

curl -fsS {cm_url}/engine/cluster/operations/{operation_id}/events \
-H 'Authorization: Bearer {cm_admin_access_token}' \
| jq
```

Проверить Kubernetes:

```
kubectl -n {engine_namespace} get pods,svc,pvc -o wide  
kubectl -n {engine_namespace} get events --sort-by=.lastTimestamp
```

7.5 Дождаться реальной готовности Engine

Operation status **APPLIED** означает, что CM отправил desired state в Kubernetes/operator. Это не то же самое, что готовые FE/CN pods.

Проверяйте CM status до **READY**:

```
curl -fsS '{cm_url}/engine/cluster/status?refresh=true' \  
-H 'Authorization: Bearer {cm_admin_access_token}' \  
| jq
```

И Kubernetes:

```
kubectl -n {engine_namespace} get statefulsets,pods -o wide  
kubectl get pods -A --field-selector=status.phase!=Running,status.phase!=Succeeded
```

Критерий готовности:

- CM cluster status **READY**;
- FE StatefulSet ready;
- CN или BE StatefulSet ready;
- нет неожиданных failed/pending pods.

7.6 Выполнить auth/RBAC bootstrap

Install не создает runtime credentials для приложений. Bootstrap запускается отдельно, только после **READY**.

Если в Engine spec есть **IDE_IMPERSONATOR** или **AI_IMPERSONATOR**, Secrets для IDE/AI должны уже существовать, потому что CM записывает generated passwords в эти existing Secrets:

```
kubectl -n {selena_namespace} get secret selena-ide-secrets selena-ai-secrets
```

```
curl -fsS -X POST {cm_url}/engine/cluster:bootstrap-auth-rbac \
  -H 'Authorization: Bearer {cm_admin_access_token}' \
  | jq
```

Скопируйте `.operationId` из ответа. Дальше это `{bootstrap_operation_id}`.

Проверить bootstrap operation:

```
curl -fsS {cm_url}/engine/cluster/operations/{bootstrap_operation_id} \
  -H 'Authorization: Bearer {cm_admin_access_token}' \
  | jq

curl -fsS
{cm_url}/engine/cluster/operations/{bootstrap_operation_id}/events \
  -H 'Authorization: Bearer {cm_admin_access_token}' \
  | jq
```

Проверить, что CM записал service passwords в Secrets:

```
curl -fsS {cm_url}/engine/cluster/impersonators \
  -H 'Authorization: Bearer {cm_admin_access_token}' \
  | jq

kubectl -n {selena_namespace} get secret selena-ide-secrets \
  -o jsonpath='{.data.IDE_SELENA_IMPERSONATOR_PASSWORD}' | wc -c

kubectl -n {selena_namespace} get secret selena-ai-secrets \
  -o jsonpath='{.data.MCP_SELENA_IMPERSONATOR_PASSWORD}' | wc -c
```

Ожидаемо: `wc -c` больше `0`.

Теперь можно устанавливать IDE, AI backend/frontend и MCP. Это описано в IDE, AI/MCP, RBAC, пользователи и direct SQL.

7.7 FE MySQL endpoint для direct SQL

Если пользователи будут подключаться через DBeaver, IntelliJ IDEA Database Tools или MySQL CLI, опубликуйте FE port `9030` approved способом: LoadBalancer, TCP Ingress/Gateway, VPN-only address или другой customer network pattern.

Проверить Service:

```
kubectl -n {engine_namespace} get svc -o wide
```

Admin smoke test:

```
mysql --protocol=TCP \  
-h {selena_fe_mysql_host} \  
-P 9030 \  
-u root \  
-e "SHOW USERS; SHOW ROLES;"
```

Пользовательские direct SQL подключения не выполняются под `root`; для них создается отдельный CLI secret через CM API. Это сделано намеренно: LDAP/Keycloak password остается web/identity credential, а MySQL/JDBC protocol получает отдельный пароль для database-доступа, который можно независимо сбросить, отозвать и не хранить в корпоративном password flow. С точки зрения безопасности не рекомендуется использовать LDAP/Keycloak password как пароль для MySQL/JDBC clients: такие clients часто сохраняют password локально, не понимают SSO/MFA policies и не должны получать основной identity password пользователя.

Пользователь получает access token через обычный CM login:

```
curl -fsS -X POST {cm_url}/auth/login \  
-H 'Content-Type: application/json' \  
-d '{"username":"{username}","password":"{web_password}"}' \  
| jq
```

Скопируйте `.accessToken` из ответа. Дальше это `{user_access_token}`. Затем пользователь сбрасывает CLI secret:

```
curl -fsS -X POST {cm_url}/me/cli-secret:reset \
  -H 'Authorization: Bearer {user_access_token}' \
  | jq
```

Скопируйте `.cliSecret` из ответа. Значение показывается только один раз; если secret потерян, выполните reset снова. Подробный direct SQL checklist описан в IDE, AI/MCP, RBAC, пользователи и direct SQL.

7.8 Storage volume для shared-data

Для shared-data clusters обычный `CREATE TABLE` требует default storage volume. Если используется external S3, создайте bucket и access key вне Selena. Если используется MinIO в Kubernetes, убедитесь, что application user создан в MinIO, а не только записан в Kubernetes Secret.

Пример создания MinIO user:

```
kubectl -n {minio_namespace} exec {minio_pod_name} -c minio -- \
  mc alias set local http://localhost:9000 {minio_root_user}
{minio_root_password}

kubectl -n {minio_namespace} exec {minio_pod_name} -c minio -- \
  mc admin user add local {s3_access_key} {s3_secret_key}

kubectl -n {minio_namespace} exec {minio_pod_name} -c minio -- \
  mc admin policy attach local readwrite --user {s3_access_key}

kubectl -n {minio_namespace} exec {minio_pod_name} -c minio -- \
  mc mb -p local/{bucket_name}
```

Создать default storage volume:

```
mysql --protocol=TCP \
-h {selena_fe_mysql_host} \
-P 9030 \
-u root <<'SQL'
CREATE STORAGE VOLUME IF NOT EXISTS {storage_volume_name}
TYPE = S3
LOCATIONS = ("s3://{bucket_name}/{bucket_prefix}/")
PROPERTIES
(
  "enabled" = "true",
  "aws.s3.region" = "{s3_region}",
  "aws.s3.endpoint" = "{s3_endpoint}",
  "aws.s3.use_aws_sdk_default_behavior" = "false",
  "aws.s3.use_instance_profile" = "false",
  "aws.s3.access_key" = "{s3_access_key}",
  "aws.s3.secret_key" = "{s3_secret_key}",
  "aws.s3.enable_ssl" = "false",
  "aws.s3.enable_path_style_access" = "true"
);

SET {storage_volume_name} AS DEFAULT STORAGE VOLUME;
SHOW STORAGE VOLUMES;
SQL
```

Как читать и менять SQL:

Строка / параметр	Что означает	Что менять
<code>mysql --protocol=TCP</code>	Подключение к FE по MySQL-compatible protocol.	Оставьте <code>TCP</code> , чтобы client не пытался использовать local socket.
<code>-h {selena_fe_mysql_host}</code>	Host/IP опубликованного FE MySQL endpoint.	Возьмите из FE Service, LoadBalancer, Ingress/Gateway или DNS-записи.
<code>-P 9030</code>	MySQL port FE.	Меняйте только если в Engine/Service явно выбран другой external port.
<code>-u root</code>	Admin-подключение для первичной настройки storage volume.	Используйте только для admin setup. Пользовательские подключения идут через CLI secret, не под <code>root</code> .
<code>CREATE STORAGE VOLUME IF NOT EXISTS {storage_volume_name}</code>	Создает named storage volume в Selena, если его еще нет.	Замените <code>{storage_volume_name}</code> на понятное имя, например <code>default_s3_volume</code> . Используйте lowercase letters, digits и underscores.
<code>TYPE = S3</code>	Тип backend storage. Для S3-compatible storage, включая MinIO, используется <code>S3</code> .	Оставьте <code>S3</code> , если storage совместим с S3 API.

Строка / параметр	Что означает	Что менять
<code>LOCATIONS = ("s3://{bucket_name}/{bucket_prefix}/")</code>	Bucket и prefix, куда Selena будет писать данные.	<code>{bucket_name}</code> должен существовать. <code>{bucket_prefix}</code> можно выбрать под окружение, например <code>prod/warehouse</code> ; не начинайте prefix с <code>/</code> .
<code>"enabled" = "true"</code>	Включает volume для использования.	Обычно оставьте <code>true</code> .
<code>"aws.s3.region" = "{s3_region}"</code>	Region S3 endpoint.	Для AWS укажите реальный region, например <code>eu-central-1</code> . Для MinIO часто используют условное значение вроде <code>us-east-1</code> , если другого требования нет.
<code>"aws.s3.endpoint" = "{s3_endpoint}"</code>	Endpoint S3 API.	Для internal MinIO это обычно <code>http://minio.{minio_namespace}.svc.cluster.local:80</code> . Для external S3 используйте endpoint, который доступен FE/CN pods.
<code>"aws.s3.use_aws_sdk_default_behavior" = "false"</code>	Selena не берет credentials из AWS SDK default chain, а использует явно заданные properties ниже.	Оставьте <code>false</code> , если задаете <code>access_key</code> и <code>secret_key</code> в SQL.
<code>"aws.s3.use_instance_profile" = "false"</code>	Selena не использует cloud instance profile/managed identity.	Оставьте <code>false</code> , если не согласован отдельный cloud-native identity flow.
<code>"aws.s3.access_key" = "{s3_access_key}"</code>	Access key S3/MinIO user-a.	Используйте application user с минимально нужными правами на bucket/prefix, не root/admin MinIO user.
<code>"aws.s3.secret_key" = "{s3_secret_key}"</code>	Secret key S3/MinIO user-a.	Не сохраняйте это значение в Git. Передавайте его только в защищенном admin runbook/secret process.
<code>"aws.s3.enable_ssl" = "false"</code>	Использовать ли HTTPS/TLS для S3 endpoint.	Для production external S3 обычно <code>true</code> . Для internal demo MinIO по HTTP может быть <code>false</code> .
<code>"aws.s3.enable_path_style_access" = "true"</code>	Использовать path-style URL, где bucket идет в path, а не в hostname.	Для MinIO обычно <code>true</code> . Для AWS S3 можно использовать customer/platform recommendation.
<code>SET {storage_volume_name} AS DEFAULT STORAGE VOLUME</code>	Делает volume default для обычного <code>CREATE TABLE</code> , если table не указала storage явно.	Имя должно совпадать с volume, созданным выше.
<code>SHOW STORAGE VOLUME S</code>	Проверка, что volume создан и виден Selena.	Оставьте как verification step.

Для internal MinIO endpoint обычно используется:

```
http://minio.{minio_namespace}.svc.cluster.local:80
```

Проверка таблицы:

```
mysql --protocol=TCP \
  -h {selena_fe_mysql_host} \
  -P 9030 \
  -u root \
  -e "CREATE DATABASE IF NOT EXISTS demo_storage; CREATE TABLE IF NOT
EXISTS demo_storage.t1 (id INT); INSERT INTO demo_storage.t1 VALUES (1);
SELECT * FROM demo_storage.t1;"
```

7.9 Engine operations через CM API

CM API endpoints для статуса и event log:

Endpoint	Когда вызывать	Что смотреть в ответе
<code>GET {cm_url}/engine/cluster/status?refresh=true</code>	После install/apply/bootstrap/scale/restart и при любой диагностике.	<code>status</code> , <code>lastApplyError</code> , FE/BE/CN component readiness, FE service, <code>lastOperationId</code> .
<code>GET {cm_url}/engine/cluster/operations/{operation_id}</code>	Когда operation только что создана или status вернул <code>lastOperationId</code> .	Operation <code>status</code> , <code>operationType</code> , timestamps, affected components, error message.
<code>GET {cm_url}/engine/cluster/operations/{operation_id}/events</code>	Для подробного event log операции.	Append-only events: step name, level, message, timestamp. Это основной лог прогресса CM operation.

Проверить текущий cluster status:

```
curl -fsS '{cm_url}/engine/cluster/status?refresh=true' \
  -H 'Authorization: Bearer {cm_admin_access_token}' \
  | jq
```

Проверить operation status:

```
curl -fsS {cm_url}/engine/cluster/operations/{operation_id} \
  -H 'Authorization: Bearer {cm_admin_access_token}' \
  | jq
```

Получить operation event log:

```
curl -fsS {cm_url}/engine/cluster/operations/{operation_id}/events \
-H 'Authorization: Bearer {cm_admin_access_token}' \
| jq
```

Scale CN:

```
curl -fsS -X POST {cm_url}/engine/cluster/components/cn:scale \
-H 'Authorization: Bearer {cm_admin_access_token}' \
-H 'Content-Type: application/json' \
-d '{"replicas":2}' \
| jq
```

Scale FE:

```
curl -fsS -X POST {cm_url}/engine/cluster/components/fe:scale \
-H 'Authorization: Bearer {cm_admin_access_token}' \
-H 'Content-Type: application/json' \
-d '{"replicas":3}' \
| jq
```

Restart:

```
curl -fsS -X POST {cm_url}/engine/cluster:restart \
-H 'Authorization: Bearer {cm_admin_access_token}' \
| jq
```

После каждой operation проверяйте не только operation status, но и реальный cluster status:

```
curl -fsS '{cm_url}/engine/cluster/status?refresh=true' \
-H 'Authorization: Bearer {cm_admin_access_token}' \
| jq
```

8 IDE, AI/MCP, RBAC, пользователи и direct SQL

Этот документ выполняется после Engine `READY` и успешного `POST /engine/cluster:bootstrap-auth-rbac`. К этому моменту CM уже создал runtime credentials и записал passwords в `selenai-secrets` и `selenai-ai-secrets`.

В Helm командах ниже используйте customer-specific values files, созданные из `docs-external/values-minimal/` и заполненные под стэнд: `{selenai_ide_values}`, `{selenai_ai_values}`, `{selenai_cm_values}`. Не устанавливайте production/test стэнд напрямую с `values-minimal`.

8.1 Установить Selena IDE

Перед установкой IDE проверьте:

- Keycloak client `selenai-ide` создан;
- Keycloak client `selenai-ide-service` создан;
- `selenai-ide-secrets` содержит `IDE_SELENA_IMPERSONATOR_PASSWORD` после Engine bootstrap;
- `PUBLIC_URL` в IDE values совпадает с `{ide_public_url}`;
- `{ide_public_url}` добавлен в Keycloak redirect URIs.

Установка:

```
helm upgrade --install selenai-ide \
  charts/internal/selenai-ide/chart \
  -n {selenai_namespace} \
  -f {selenai_ide_values} \
  --wait --timeout 10m
```

Проверка:

```
kubectl -n {selena_namespace} rollout status deploy/selena-ide-web
kubectl -n {selena_namespace} get pods -l
app.kubernetes.io/instance=selena-ide -o wide
curl -fsS {ide_public_url}/health
```

Browser check:

1. Откройте `{ide_public_url}`.
2. Войдите через Keycloak.
3. Откройте SQL console.
4. Выполните простой query, который разрешен ролями пользователя.

8.2 Установить Selena AI backend/frontend/MCP

Перед установкой AI проверьте:

- Keycloak client `selena-ai` создан;
- `selena-ai-secrets` содержит `MCP_SELENA_IMPERSONATOR_PASSWORD` после Engine bootstrap;
- `AI_BACKEND_LLM_API_KEY` является реальным provider key, не placeholder;
- AI values содержат `{ai_public_url}`, `{ide_public_url}` и Keycloak URLs;
- при `components.mcp.replicaCount > 1` включена sticky session affinity для MCP Service.

Установка:

```
helm upgrade --install selena-ai \
  charts/internal/selena-ai/chart \
  -n {selena_namespace} \
  -f {selena_ai_values} \
  --wait --timeout 10m
```

Проверка:

```
kubectl -n {selena_namespace} rollout status deploy/selena-ai-backend
kubectl -n {selena_namespace} rollout status deploy/selena-ai-frontend
kubectl -n {selena_namespace} rollout status deploy/selena-ai-mcp
kubectl -n {selena_namespace} get deploy selena-ai-backend selena-ai-frontend selena-ai-mcp -o wide
curl -fsS {ai_public_url}/api/v1/health | jq
```

Ожидаемый health:

```
{
  "status": "ok",
  "components": {
    "agent": "ok",
    "mcp": "ok",
    "storage": "ok"
  }
}
```

Browser check:

1. Откройте IDE.
2. Войдите пользователем, у которого есть Selena data role.
3. Откройте AI Agent из IDE.
4. Попросите выполнить `SELECT 1 AS ok`.
5. Убедитесь, что AI вызывает query tool и показывает результат ¹.

8.3 Как работает синхронизация Keycloak groups -> Selena users/roles

CM не читает LDAP/AD напрямую. Источник правды для пользователей и групп - Keycloak.

Текущая схема:

```
Keycloak group membership
-> CM scheduled membership refresh
-> CM membership snapshot в PostgreSQL
-> mapped external groups
-> Selena roles bound to external groups
-> native Selena SQL users and role grants
```

Расписание membership refresh включено по умолчанию в CM. В minimal values его не нужно задавать: меняйте эти свойства только если customer runbook требует другой SLA или другой lock window.

Свойство	Рекомендуемое значение	Что делает
CM_GROUP_FILE_REFRESH_ENABLED	"true"	Product default. Включает cron refresh из Keycloak.
CM_GROUP_FILE_REFRESH_INITIAL_DELAY	10s	Через сколько после старта CM выполнить первый refresh.
CM_GROUP_FILE_REFRESH_INTERVAL	30s	Как часто CM перечитывает Keycloak groups и users.
CM_GROUP_FILE_FULL_RECONCILE_INTERVAL	PT6H	Как часто делать полный reconcile даже если snapshot hash не изменился.
CM_GROUP_FILE_REFRESH_LOCK_AT_MOST_FOR	PT5M	Distributed lock через CM DB, чтобы две CM replicas не выполняли refresh одновременно.
CM_GROUP_FILE_REFRESH_LOCK_AT_LEAST_FOR	PT1S	Минимальное время удержания lock, защищает от слишком частых повторов.

Что происходит каждые {refresh_interval}:

1. Одна CM replica берет ShedLock `cm-group-file-refresh`.
2. CM через Keycloak Admin API читает effective members только для groups, которые указаны в external group mappings.
3. CM генерирует membership snapshot и сравнивает hash с последним applied snapshot.
4. Если hash изменился, CM reconcile'ит только пользователей, membership которых изменился: добавляет missing roles, revoke'ит лишние roles, создает новых Selena users и удаляет stale Selena users/metadata.
5. Если hash не изменился, CM не трогает Selena до следующего full reconcile.
6. Раз в {full_reconcile_interval} CM делает полный safety reconcile всех пользователей из snapshot.

Практический SLA: при `CM_GROUP_FILE_REFRESH_INTERVAL=30s` изменение группы в Keycloak обычно доезжает до Selena за один refresh cycle плюс время Keycloak Admin API и Selena SQL DDL. Для production runbook закладывайте 1-2 минуты.

Защита от опасного пустого snapshot: если предыдущий applied snapshot имел members, а новый snapshot внезапно пустой, CM откажется reconcile'ить его, чтобы не снять роли массово из-за сбоя Keycloak/LDAP.

Если external group mappings еще не настроены, scheduled refresh безопасно пропускается и не применяет пустой snapshot. Доступ к данным появится только после явного mapping Keycloak groups в Selena external groups.

Проверить текущий snapshot:

```
curl -fsS {cm_url}/selena/membership-snapshot/status \
-H 'Authorization: Bearer {cm_admin_access_token}' \
| jq

curl -fsS {cm_url}/selena/membership-snapshot/groups \
-H 'Authorization: Bearer {cm_admin_access_token}' \
| jq
```

8.4 CM roles и protected API

CM API roles выдаются через Keycloak group paths в `group-role-mappings`. Эти roles нужны только для доступа в личный кабинет/административный UI Selena Cluster Manager и к защищенным CM API. Они не дают пользователю прав на чтение или запись данных в Selena Engine. Доступ к данным настраивается отдельно через Selena roles/grants и external group mappings в следующем разделе.

Минимальная схема:

```
{
  "cm": {
    "security": {
      "group-role-mappings": {
        "[/cm/admins]": "cm_admin",
        "[/cm/role-admins]": "cm_role_admin",
        "[/cm/viewers]": "cm_viewer"
      }
    }
  }
}
```

Проверить текущего пользователя:

```
curl -fsS {cm_url}/auth/me \  
-H 'Authorization: Bearer {cm_admin_access_token}' \  
| jq
```

Negative check для пользователя без CM roles:

```
curl -i -sS {cm_url}/engine/cluster/status?refresh=true \  
-H 'Authorization: Bearer {no_cm_role_access_token}'
```

Ожидаемо: HTTP 403.

8.5 External group mappings

External group mapping переводит Keycloak group path в короткое имя external group для Selena grants.

Эти mappings нужны для бизнесового доступа к данным в Selena Engine. Их может быть сколько угодно: заказчик сам решает, какие группы отражают его команды, департаменты, проекты и уровни доступа. Например, можно завести groups `/data/analytics/readers`, `/data/analytics/writers`, `/data/developers`, `/data/finance/admins` или любые другие paths, которые соответствуют принятой у заказчика модели доступа.

External group name справа - это короткое техническое имя, к которому потом привязываются Selena roles. Например, Keycloak group `/data/analytics/readers` можно mapped в external group `analytics_readers`, а затем дать этой external group Selena role с read-only privileges на нужные catalogs/databases/tables.

По умолчанию CM также включает `fallback-strategy: FULL_PATH`: каждая Keycloak group автоматически получает Selena-safe external group name по full path. Например `/cm/admins` становится `cm_admins`, а `/local/selena-readers` становится `local_selena_readers`. Explicit mappings ниже остаются override для случаев, где нужно сохранить legacy short names.

Пример:

```
{
  "cm": {
    "identity": {
      "external-groups": {
        "fallback-strategy": "FULL_PATH",
        "mappings": {
          "[/ldap/selena-readers]": "ldap_selena_readers",
          "[/ldap/selena-writers]": "ldap_selena_writers",
          "[/local/selena-readers]": "local_selena_readers",
          "[/local/selena-writers]": "local_selena_writers"
        }
      }
    }
  }
}
```

Правила:

- любая Keycloak group может получить Selena SQL role через derived external group;
- используйте explicit mappings, если нужно переименовать derived external group;
- external group names используйте lowercase letters, digits и underscores;
- новая Keycloak group не дает доступа, пока ее derived external group не bound к Selena role.

После изменения mappings обновите CM:

```
helm upgrade --install selena-cm \
  charts/internal/selena-cm/chart \
  -n {selena_namespace} \
  -f {selena_cm_values} \
  --wait --timeout 10m
```

8.6 Создать Selenia roles

```
curl -fsS -X POST {cm_url}/selena/roles \  
-H 'Authorization: Bearer {cm_admin_access_token}' \  
-H 'Content-Type: application/json' \  
-d '{"name":"selena_readonly"}' \  
| jq  
  
curl -fsS -X POST {cm_url}/selena/roles \  
-H 'Authorization: Bearer {cm_admin_access_token}' \  
-H 'Content-Type: application/json' \  
-d '{"name":"selena_writer"}' \  
| jq
```

Проверить:

```
curl -fsS {cm_url}/selena/roles \  
-H 'Authorization: Bearer {cm_admin_access_token}' \  
| jq
```

8.7 Выдать privileges roles

Этот раздел содержит минимальный пример. Подробная матрица privileges, object types, scopes и типовых ошибок описана в Selenia grants, roles и CM API.

USAGE на catalog:

```
curl -fsS -X POST {cm_url}/selena/roles/selena_readonly/grants \  
-H 'Authorization: Bearer {cm_admin_access_token}' \  
-H 'Content-Type: application/json' \  
-d '{  
  "privileges": ["USAGE"],  
  "objectType": "CATALOG",  
  "objectQualifier": "default_catalog",  
  "scope": "OBJECT",  
  "withGrantOption": false  
}' \  
| jq
```

Read-only access на database:

```
curl -fsS -X POST {cm_url}/selena/roles/selena_readonly/grants \
-H 'Authorization: Bearer {cm_admin_access_token}' \
-H 'Content-Type: application/json' \
-d '{
  "privileges": ["SELECT"],
  "objectType": "TABLE",
  "objectQualifier": "{database_name}",
  "scope": "ALL_TABLES_IN_DATABASE",
  "withGrantOption": false
}' \
| jq
```

Write access to table:

```
curl -fsS -X POST {cm_url}/selena/roles/selena_writer/grants \
-H 'Authorization: Bearer {cm_admin_access_token}' \
-H 'Content-Type: application/json' \
-d '{
  "privileges": ["SELECT", "INSERT"],
  "objectType": "TABLE",
  "objectQualifier": "{database_name}.{table_name}",
  "scope": "OBJECT",
  "withGrantOption": false
}' \
| jq
```

Проверить grants:

```
curl -fsS {cm_url}/selena/roles/selena_readonly/grants \
-H 'Authorization: Bearer {cm_admin_access_token}' \
| jq
```

8.8 Привязать Selenia roles к external groups

```
curl -fsS -X POST {cm_url}/selena/external-groups/ldap_selena_readers/roles \
-H 'Authorization: Bearer {cm_admin_access_token}' \
-H 'Content-Type: application/json' \
-d '{"roleName":"selena_readonly"}' \
| jq

curl -fsS -X POST {cm_url}/selena/external-
groups/local_selena_readers/roles \
-H 'Authorization: Bearer {cm_admin_access_token}' \
-H 'Content-Type: application/json' \
-d '{"roleName":"selena_readonly"}' \
| jq

curl -fsS -X POST {cm_url}/selena/external-groups/ldap_selena_writers/roles \
-H 'Authorization: Bearer {cm_admin_access_token}' \
-H 'Content-Type: application/json' \
-d '{"roleName":"selena_writer"}' \
| jq

curl -fsS -X POST {cm_url}/selena/external-
groups/local_selena_writers/roles \
-H 'Authorization: Bearer {cm_admin_access_token}' \
-H 'Content-Type: application/json' \
-d '{"roleName":"selena_writer"}' \
| jq
```

Проверить external group:

```
curl -fsS {cm_url}/selena/external-groups/ldap_selena_readers/roles \
-H 'Authorization: Bearer {cm_admin_access_token}' \
| jq
```

После следующего membership refresh CM материализует роли на users.

8.9 Direct SQL CLI secret

Web password Keycloak/LDAP не используется для direct SQL. Это отдельная граница безопасности: Keycloak/LDAP password нужен для web login и identity policies, а

MySQL/JDBC clients используют CM-generated CLI secret. Так основной identity password не попадает в database tools, может оставаться под SSO/MFA/password-policy контролем, а direct SQL secret можно независимо сбросить при утечке или увольнении пользователя.

Пользователь логинится в CM и сбрасывает CLI secret:

```
curl -fsS -X POST {cm_url}/auth/login \
  -H 'Content-Type: application/json' \
  -d '{"username":"{username}","password":"{web_password}"}' \
  | jq
```

Скопируйте `.accessToken` из ответа. Дальше это `{user_access_token}`.

```
curl -fsS -X POST {cm_url}/me/cli-secret:reset \
  -H 'Authorization: Bearer {user_access_token}' \
  | jq
```

Скопируйте `.cliSecret` из ответа. Дальше это `{cli_secret}`. Secret больше нельзя прочитать; если он потерян, выполните reset снова.

Подключение:

```
mysql --protocol=TCP \
  -h {selena_fe_mysql_host} \
  -P 9030 \
  -u {username} \
  -p{cli_secret} \
  -e "SELECT CURRENT_USER(); SHOW GRANTS;"
```

DBeaver и IntelliJ IDEA Database Tools используют обычный MySQL driver:

Поле	Значение
Host	<code>{selena_fe_mysql_host}</code>
Port	<code>9030</code>
User	<code>{username}</code>
Password	<code>{cli_secret}</code>

MySQL clear-password plugin не нужен, потому что Selena Engine не настроен как LDAP client.

8.10 Проверка изменения membership

Проверка добавления пользователя в group:

1. Добавьте `{username}` в Keycloak group, mapped на external group.
2. Подождите один refresh interval, обычно `30s`.
3. Проверьте snapshot:

```
curl -fsS {cm_url}/selena/membership-  
snapshot/groups/{external_group}/members \  
-H 'Authorization: Bearer {cm_admin_access_token}' \  
| jq
```

4. Проверьте роли пользователя:

```
curl -fsS {cm_url}/me/direct-sql \  
-H 'Authorization: Bearer {user_access_token}' \  
| jq
```

5. Проверьте direct SQL:

```
mysql --protocol=TCP \  
-h {selena_fe_mysql_host} \  
-P 9030 \  
-u {username} \  
-p{cli_secret} \  
-e "SHOW GRANTS;"
```

Проверка удаления пользователя из group:

1. Удалите `{username}` из Keycloak group.
2. Подождите один refresh interval.
3. Убедитесь, что user исчез из members external group.
4. Убедитесь, что роль удалена из `/me/direct-sql` и `SHOW GRANTS`.

Проверка новой group:

1. Создайте Keycloak group.
2. Добавьте пользователя в Keycloak group.
3. Дождитесь refresh, чтобы CM показал derived external group.
4. Создайте или выберите Selenia role.
5. Привяжите role к derived external group через CM UI или API.
6. Дождитесь materialization и проверьте role.

Проверка удаления group:

1. Уберите пользователей из Keycloak group или удалите group.
2. Дождитесь refresh.
3. Проверьте, что CM revoked roles, выданные через эту external group.
4. Только после этого удаляйте mapping из CM values.

9 Приемка, эксплуатация и troubleshooting

Этот документ используют после первичной установки и после upgrade. Цель - проверить, что Kubernetes, Keycloak, CM, Engine, IDE, AI/MCP, Grafana, RBAC и direct SQL работают как одна система.

Во всех командах значения в фигурных скобках заменяются вручную. Например, `{cm_url}` - это public URL CM API из шага service exposure, а `{cm_admin_access_token}` - поле `.accessToken` из ответа `POST /auth/login`.

9.1 Проверьте Kubernetes baseline

Выполните:

```
kubectl get nodes -o wide
kubectl get pods -A
kubectl get pods -A --field-selector=status.phase!=Running,status.phase!=Succeeded
helm list -A
```

Ожидаемо:

- все nodes в состоянии `Ready`;
- нет неожиданных pods в `Pending`, `CrashLoopBackOff`, `ImagePullBackOff` или долгом `Terminating`;
- Helm releases установлены в ожидаемых namespaces.

Если есть non-running pods, сначала смотрите events и logs:

```
kubectl -n {namespace} describe pod {pod_name}
kubectl -n {namespace} logs {pod_name} --tail=200
kubectl -n {namespace} get events --sort-by=.lastTimestamp
```

9.2 Проверьте CM и лицензию

CM должен отвечать до установки IDE/AI/MCP, потому что Engine lifecycle и bootstrap выполняются через CM API.

```
curl -fsS {cm_url}/actuator/health | jq
```

Ожидаемо: `status` равен `UP`.

Получите admin token:

```
curl -fsS -X POST {cm_url}/auth/login \
  -H 'Content-Type: application/json' \
  -d '{"username":"{cm_admin_username}","password":"{cm_admin_web_password}"}' \
  | jq
```

Скопируйте `.accessToken`; дальше это `{cm_admin_access_token}`.

Проверьте identity и license:

```
curl -fsS {cm_url}/auth/me \
  -H 'Authorization: Bearer {cm_admin_access_token}' \
  | jq

curl -fsS {cm_url}/license \
  -H 'Authorization: Bearer {cm_admin_access_token}' \
  | jq
```

Ожидаемо:

- `/auth/me` возвращает вашего Keycloak username;
- `cmRoles` содержит `cm_admin`;
- `license status` позволяет работать с защищенными endpoint-ами.

9.3 Проверьте Engine через CM API

Engine устанавливается и управляется только через CM API. Не используйте `kubectl scale` для штатных FE/CN операций: Kubernetes должен видеть изменения из Selena

Custom Resource, а не ручной scale StatefulSet.

Проверьте статус:

```
curl -fsS '{cm_url}/engine/cluster/status?refresh=true' \  
  -H 'Authorization: Bearer {cm_admin_access_token}' \  
  | jq
```

Ожидаемо:

- `status` равен `READY`;
- `lastApplyError` пустой или `null`;
- FE и CN components показывают ожидаемое число replicas;
- FE service имеет endpoint для SQL.

Проверьте operation events, если предыдущая команда вернула `lastOperationId` или вы только что запускали install/scale/restart:

```
curl -fsS {cm_url}/engine/cluster/operations/{operation_id}/events \  
  -H 'Authorization: Bearer {cm_admin_access_token}' \  
  | jq
```

Проверьте Kubernetes ресурсы Engine только как read-only диагностики:

```
kubectl -n {engine_namespace} get pods,svc,pvc -o wide
```

9.4 Проверьте CM RBAC

Admin должен видеть diagnostic и mutating endpoints:

```
curl -fsS {cm_url}/selena/status \  
  -H 'Authorization: Bearer {cm_admin_access_token}' \  
  | jq  
  
curl -fsS {cm_url}/selena/roles \  
  -H 'Authorization: Bearer {cm_admin_access_token}' \  
  | jq
```

Получите viewer token так же через `POST /auth/login`, но под пользователем из `/cm/viewers`. Скопируйте `.accessToken`; дальше это `{cm_viewer_access_token}`.

Viewer должен читать разрешенные status endpoints, но не должен менять Engine:

```
curl -i -sS -X POST {cm_url}/engine/cluster/components/cn:scale \
  -H 'Authorization: Bearer {cm_viewer_access_token}' \
  -H 'Content-Type: application/json' \
  -d '{"replicas":3}'
```

Ожидаемо: HTTP 403.

Пользователь без `/cm/admins`, `/cm/role-admins` и `/cm/viewers` не должен получать доступ к protected CM API. Для проверки получите token такого пользователя и выполните:

```
curl -i -sS {cm_url}/selena/status \
  -H 'Authorization: Bearer {cm_plain_user_access_token}'
```

Ожидаемо: HTTP 403.

9.5 Проверьте Keycloak token claims

Проверьте, что token содержит стабильный username, full group paths и нужный audience. Сначала получите access token через Keycloak:

```
curl -fsS -X POST {keycloak_public_url}/realms/selena/protocol/openid-connect/token \
  -H 'Content-Type: application/x-www-form-urlencoded' \
  --data-urlencode 'grant_type=password' \
  --data-urlencode 'client_id=selena-engine' \
  --data-urlencode 'client_secret={selena_engine_client_secret}' \
  --data-urlencode 'username={keycloak_test_username}' \
  --data-urlencode 'password={keycloak_test_password}' \
  | jq
```

Скопируйте `.access_token`; дальше это `{keycloak_test_access_token}`.

Расшифруйте payload:

```
python3 -c 'import base64,json; token="{keycloak_test_access_token}";
payload=token.split(".")[1]; payload += "=" * (-len(payload) % 4);
print(json.dumps(json.loads(base64.urlsafe_b64decode(payload)), indent=2,
ensure_ascii=False))'
```

Ожидаемо:

- `preferred_username` совпадает с canonical Selena username;
- `groups` содержит full paths, например `/cm/admins`;
- `aud` содержит `selen-engine`.

9.6 Проверьте membership refresh и RBAC

CM не читает LDAP напрямую. Цепочка такая:

```
LDAP/AD -> Keycloak federation -> Keycloak groups -> CM scheduled refresh
-> CM PostgreSQL snapshot -> Selena native users and role grants
```

Membership refresh включен по умолчанию в CM. Эти values обычно не задают в minimal profile; переопределяйте их только для изменения SLA или lock window.

Property	Рекомендуемый default	Что означает
<code>CM_GROUP_FILE_REFRESH_ENABLED</code>	<code>true</code>	Product default. Включает scheduled refresh Keycloak membership.
<code>CM_GROUP_FILE_REFRESH_INITIAL_DELAY</code>	<code>10s</code>	Сколько CM ждет после старта перед первым refresh.
<code>CM_GROUP_FILE_REFRESH_INTERVAL</code>	<code>30s</code>	Как часто CM проверяет изменения membership.
<code>CM_GROUP_FILE_FULL_RECONCILE_INTERVAL</code>	<code>PT6H</code>	Как часто CM делает полный safety reconcile даже без hash changes.

В HA только одна CM replica выполняет refresh благодаря ShedLock. На обычное добавление или удаление пользователя из группы закладывайте 1-2 минуты: это покрывает interval, Keycloak Admin API и SQL reconcile в Selena.

Если external group mappings пустые, CM покажет skipped refresh и не будет reconcile'ить пустой snapshot. Это ожидаемо до настройки group mappings.

Проверьте snapshot status:

```
curl -fsS {cm_url}/selena/membership-snapshot/status \
-H 'Authorization: Bearer {cm_admin_access_token}' \
| jq

curl -fsS {cm_url}/selena/membership-snapshot/groups \
-H 'Authorization: Bearer {cm_admin_access_token}' \
| jq
```

Regression сценарии:

Сценарий	Действие	Ожидаемо
Добавить пользователя в mapped group	В Keycloak добавьте {rbac_test_username} в /lda p/selena-readers или другую mapped group.	После refresh у пользователя появляется mapped Selena role.
Удалить пользователя из mapped group	Уберите того же пользователя из группы.	После refresh mapped role исчезает, остальные роли не трогаются.
Создать новую business group	Создайте группу в Keycloak, добавьте mapping в CM values и сделайте Helm upgrade CM.	После refresh members получают роль из mapping.
Удалить group mapping	Уберите mapping из CM values и сделайте Helm upgrade CM.	После full reconcile stale access исчезает.
Удалить группу в Keycloak	Удалите тестовую группу после удаления mapping.	CM refresh не должен ломать остальные groups.

Проверьте drift для пользователя:

```
curl -fsS {cm_url}/me/drift \
-H 'Authorization: Bearer {rbac_test_user_access_token}' \
| jq
```

Проверьте grants напрямую:

```
mysql --protocol=TCP \
-h {selena_fe_mysql_host} \
-P 9030 \
-u {rbac_test_username} \
-p'{rbac_test_cli_secret}' \
-e "SHOW GRANTS;"
```

9.7 Проверьте direct SQL CLI secret

И LDAP/AD-backed users, и Keycloak-local users используют Keycloak для web login, но direct SQL password всегда генерирует CM.

Получите user token через CM login:

```
curl -fsS -X POST {cm_url}/auth/login \  
  -H 'Content-Type: application/json' \  
  -d '{"username":"{sql_test_username}","password":"  
{sql_test_web_password}"}' \  
  | jq
```

Скопируйте `.accessToken`; дальше это `{sql_test_access_token}`.

Сгенерируйте CLI secret:

```
curl -fsS -X POST {cm_url}/me/cli-secret:reset \  
  -H 'Authorization: Bearer {sql_test_access_token}' \  
  | jq
```

Скопируйте `.cliSecret`; дальше это `{sql_test_cli_secret}`. Это значение показывается только один раз.

Подключитесь стандартным MySQL-compatible client:

```
mysql --protocol=TCP \  
  -h {selena_fe_mysql_host} \  
  -P 9030 \  
  -u {sql_test_username} \  
  -p'{sql_test_cli_secret}' \  
  -e "SELECT CURRENT_USER(); SHOW GRANTS;"
```

DBeaver и IntelliJ IDEA Database Tools должны использовать обычный MySQL driver: host `{selena_fe_mysql_host}`, port `9030`, user `{sql_test_username}`, password `{sql_test_cli_secret}`. `mysql_clear_password` для LDAP password flow не нужен, потому что Selena Engine не подключается к LDAP напрямую.

9.8 Проверьте IDE

Откройте `{ide_public_url}` в браузере.

Проверьте:

- login переводит в Keycloak realm `selen`;
- после login отображается IDE;
- user видит только каталоги и объекты, разрешенные его Selen roles;
- простой query выполняется от имени user, а не service account;
- logout возвращает пользователя в expected public URL, без redirect loop.

Если IDE не открывается:

```
kubectl -n {selen_namespace} get pods -l
app.kubernetes.io/instance=selen-ide -o wide
kubectl -n {selen_namespace} logs deploy/selen-ide-web --tail=200
kubectl -n {selen_namespace} logs deploy/selen-ide-worker --tail=200
```

9.9 Проверьте AI frontend, AI backend и MCP

AI/MCP ставятся только после Engine `READY` и `POST /engine/cluster:bootstrap-auth-rbac`, потому что MCP должен знать FE host и runtime secret.

Проверьте pods:

```
kubectl -n {selen_namespace} get deploy selen-ai-backend selen-ai-frontend selen-ai-mcp -o wide
kubectl -n {selen_namespace} get pods -l
app.kubernetes.io/instance=selen-ai -o wide
kubectl -n {selen_namespace} get svc selen-ai-mcp -o
jsonpath='{.spec.sessionAffinity}{"\n"}'
```

Ожидаемо:

- backend, frontend и MCP имеют нужное число replicas;
- pods `Ready`;
- MCP service использует `ClientIP` session affinity, если это включено в values.

Проверьте backend health:

```
curl -fsS {ai_public_url}/api/v1/health | jq
```

В IDE откройте AI panel и отправьте короткий вопрос, например:

```
show my current catalogs
```

Ожидаемо: AI отвечает, а запросы к Selena идут через MCP и impersonation.

Если AI не отвечает:

```
kubectl -n {selena_namespace} logs deploy/selena-ai-backend --tail=200
kubectl -n {selena_namespace} logs deploy/selena-ai-mcp --tail=200
```

9.10 Проверьте Grafana

Откройте `{grafana_public_url}`.

Проверьте три пользователя:

Пользователь	Keycloak membership	Ожидаемый результат
Grafana admin	<code>/grafana/admins</code>	Входит как Grafana Admin.
Grafana viewer	<code>/grafana/viewers</code>	Входит как Grafana Viewer.
Без Grafana role	Нет <code>/grafana/admins</code> и <code>/grafana/viewers</code>	Доступ запрещен.

После изменения membership выйдите из Grafana и войдите снова, чтобы Grafana получила свежий OAuth token.

9.11 Проверьте Engine lifecycle операции

Все операции запускайте через CM API. После каждой операции смотрите operation events и финальный cluster status.

Эти проверки меняют live Engine replicas или перезапускают cluster. На production выполняйте их только если это входит в acceptance scope и есть approved maintenance window. Для обычной read-only приемки достаточно проверить status/events из раздела 3.

Scale CN down:

```
curl -fsS -X POST {cm_url}/engine/cluster/components/cn:scale \
-H 'Authorization: Bearer {cm_admin_access_token}' \
-H 'Content-Type: application/json' \
-d '{"replicas":2}' \
| jq
```

Скопируйте `.operationId`; дальше это `{scale_cn_down_operation_id}`.

```
curl -fsS
{cm_url}/engine/cluster/operations/{scale_cn_down_operation_id}/events \
-H 'Authorization: Bearer {cm_admin_access_token}' \
| jq

curl -fsS '{cm_url}/engine/cluster/status?refresh=true' \
-H 'Authorization: Bearer {cm_admin_access_token}' \
| jq
```

Scale CN back:

```
curl -fsS -X POST {cm_url}/engine/cluster/components/cn:scale \
-H 'Authorization: Bearer {cm_admin_access_token}' \
-H 'Content-Type: application/json' \
-d '{"replicas":3}' \
| jq
```

Scale FE follower down и back выполняется тем же endpoint-ом с component `fe`. Не scale-ите FE до 0: production cluster потеряет SQL endpoint.

```
curl -fsS -X POST {cm_url}/engine/cluster/components/fe:scale \  
-H 'Authorization: Bearer {cm_admin_access_token}' \  
-H 'Content-Type: application/json' \  
-d '{"replicas":2}' \  
| jq
```

```
curl -fsS -X POST {cm_url}/engine/cluster/components/fe:scale \  
-H 'Authorization: Bearer {cm_admin_access_token}' \  
-H 'Content-Type: application/json' \  
-d '{"replicas":3}' \  
| jq
```

Restart через CM:

```
curl -fsS -X POST {cm_url}/engine/cluster:restart \  
-H 'Authorization: Bearer {cm_admin_access_token}' \  
| jq
```

Ожидаемо:

- operation status становится `APPLIED`;
- cluster возвращается в `READY`;
- FE service и direct SQL продолжают работать после восстановления replicas.

9.12 Проверьте Helm values перед upgrade

External docs используют customer-specific values files, созданные из шаблонов `deploy/k8s/docs-external/values-minimal`. В командах ниже `{selena_cm_values}` - это ваш заполненный CM values file, а не шаблон `values-minimal`.

```
helm lint deploy/k8s/charts/internal/selena-cm/chart \  
-f {selena_cm_values}
```

```
helm template selena-cm deploy/k8s/charts/internal/selena-cm/chart \  
-n {selena_namespace} \  
-f {selena_cm_values}
```

Перед production upgrade сделайте:

```
helm upgrade --install selena-cm deploy/k8s/charts/internal/selena-cm/chart \
  -n {selena_namespace} \
  -f {selena_cm_values} \
  --wait \
  --timeout 15m

kubectl -n {selena_namespace} rollout status deploy/selena-cm
curl -fsS {cm_url}/actuator/health | jq
```

Если `helm --wait` упал только из-за долгого LoadBalancer provisioning, сначала проверьте реальные services и pods. Иногда cloud provider создает LB дольше Helm timeout; повторный `helm upgrade --install` должен завершиться clean.

9.13 Частые проблемы

Симптом	Что проверить
CM login возвращает 401	Keycloak issuer, client secret <code>selena-engine</code> , user password, <code>preferred_username</code> .
User вошел в CM, но не получил role	Full group path в token, CM group-role mapping, membership snapshot status, refresh interval.
Direct SQL password не работает	Используется ли <code>.cliSecret</code> из последнего reset; подключение идет к FE MySQL port <code>9030</code> ; username совпадает с Keycloak username.
IDE открывается, но query permission denied	Проверьте grants пользователя через <code>SHOW GRANTS</code> и mapped Selena roles.
AI panel не отвечает	Проверьте AI backend health, MCP pods, LLM secret и MCP access к FE service.
Grafana после logout возвращает на старую страницу	Проверьте Grafana OAuth logout URL и Keycloak client <code>grafana</code> redirect URIs.
Engine status не <code>READY</code>	Смотрите operation events, <code>kubectl -n {engine_namespace} get pods,events</code> , operator logs.
LoadBalancer services долго остаются <code>pending</code>	Проверьте cloud-provider events и <code>metadata.annotations</code> у всех <code>Service</code> типа <code>LoadBalancer</code> . Если старый overlay задавал static IP/resource-group annotations, Helm/operator могут не удалить map keys при переходе на <code>null/{};</code> ; удалите stale annotations явно командой <code>kubectl annotate svc <name> <annotation-key>-</code> .

Диагностика operation events:

```
curl -fsS {cm_url}/engine/cluster/operations/{operation_id}/events \  
-H 'Authorization: Bearer {cm_admin_access_token}' \  
| jq
```

10 Selena grants, roles и CM API

Этот документ объясняет, как настраивать доступ к данным в Selena Engine через Selena Cluster Manager API. Он написан для администраторов, которые раньше не работали со StarRocks/Selena RBAC и не знают, чем отличается privilege на catalog, database, table, view, materialized view, resource и другие объекты.

В Selena используется StarRocks-compatible privilege model. Для справки при подготовке этого документа сверялись официальные StarRocks 4.0 pages:

- Overview of privileges
- Privileges supported by StarRocks
- GRANT
- Privilege FAQ
- SHOW TABLES
- SHOW GRANTS

Важно: документация StarRocks - upstream reference. Для конкретного внедрения проверяйте поведение на используемом Selena Engine image.

10.1 Модель доступа Selena

Внешняя production-модель Selena не раздает права напрямую каждому пользователю вручную. Основной путь такой:

```
Keycloak group
-> CM external group mapping
-> Selena external group
-> Selena role
-> Selena privileges/grants
-> CM materializes roles on canonical SQL users
```

Пример:

```

Keycloak group:      /data/analytics/readers
CM external group:   analytics_readers
Selena role:         analytics_readonly
Selena grants:       USAGE ON CATALOG lakekeeper_iceberg
                     SELECT ON ALL TABLES IN DATABASE mart
SQL user:            alice

```

Пользователь `alice` получает роль не потому, что роль напрямую выдали пользователю, а потому что:

1. `alice` состоит в Keycloak group `/data/analytics/readers`;
2. CM values map'ят эту group в external group `analytics_readers`;
3. external group `analytics_readers` bound к роли `analytics_readonly`;
4. CM membership refresh materializes expected role на Selena SQL user `alice`.

CM API roles (`cm_admin`, `cm_role_admin`, `cm_viewer`) не дают доступа к данным. Они только разрешают вызывать protected CM API. Доступ к данным задают Selena roles/grants.

10.2 Кто может менять grants

Для операций ниже нужен CM access token пользователя с `cm_admin` или `cm_role_admin`.

В примерах используется:

Placeholder	Что это
<code>{cm_url}</code>	Base URL CM API, например <code>https://cm.example.com/api/v1</code> .
<code>{cm_admin_access_token}</code>	Access token пользователя с <code>cm_admin</code> или <code>cm_role_admin</code> .
<code>{role_name}</code>	Selena custom role, например <code>analytics_readonly</code> .
<code>{external_group}</code>	Selena external group из CM mapping, например <code>analytics_readers</code> .

Если token еще не получен:

```
curl -fsS -X POST {cm_url}/auth/login \  
-H 'Content-Type: application/json' \  
-d '{  
  "username": "{cm_admin_username}",  
  "password": "{cm_admin_password}"  
}' \  
| jq
```

Скопируйте `.accessToken` в `{cm_admin_access_token}`.

10.3 Быстрый happy path

Перед этим примером убедитесь, что:

- Keycloak group уже добавлена в CM `external-groups.mappings`;
- external group name из mapping совпадает с `analytics_readers`;
- catalog `lakekeeper_iceberg`, database `mart` и нужные tables уже существуют в Selena.

Создайте custom role:

```
curl -fsS -X POST {cm_url}/selena/roles \  
-H 'Authorization: Bearer {cm_admin_access_token}' \  
-H 'Content-Type: application/json' \  
-d '{"name":"analytics_readonly"}' \  
| jq
```

Дайте роли доступ к catalog:

```
curl -fsS -X POST {cm_url}/selena/roles/analytics_readonly/grants \  
-H 'Authorization: Bearer {cm_admin_access_token}' \  
-H 'Content-Type: application/json' \  
-d '{  
  "privileges": ["USAGE"],  
  "objectType": "CATALOG",  
  "objectQualifier": "lakekeeper_iceberg",  
  "scope": "OBJECT",  
  "withGrantOption": false  
}' \  
| jq
```

Дайте роли read access ко всем таблицам database `mart` в этом catalog:

```
curl -fsS -X POST {cm_url}/selena/roles/analytics_readonly/grants \
-H 'Authorization: Bearer {cm_admin_access_token}' \
-H 'Content-Type: application/json' \
-d '{
  "privileges": ["SELECT"],
  "objectType": "TABLE",
  "objectQualifier": "mart",
  "catalog": "lakekeeper_iceberg",
  "scope": "ALL_TABLES_IN_DATABASE",
  "withGrantOption": false
}' \
| jq
```

Привяжите role к external group:

```
curl -fsS -X POST {cm_url}/selena/external-groups/analytics_readers/roles \
-H 'Authorization: Bearer {cm_admin_access_token}' \
-H 'Content-Type: application/json' \
-d '{"roleName": "analytics_readonly"}' \
| jq
```

Если external group еще не появляется в membership snapshot, проверьте mapping и Keycloak membership до привязки role:

```
curl -fsS {cm_url}/selena/membership-
snapshot/groups/analytics_readers/members \
-H 'Authorization: Bearer {cm_admin_access_token}' \
| jq
```

Проверьте grants роли:

```
curl -fsS {cm_url}/selena/roles/analytics_readonly/grants \
-H 'Authorization: Bearer {cm_admin_access_token}' \
| jq
```

Проверьте binding external group:

```
curl -fsS {cm_url}/selena/external-groups/analytics_readers/roles \
  -H 'Authorization: Bearer {cm_admin_access_token}' \
  | jq
```

После следующего membership refresh CM выдаст роль текущим members external group. Если role binding меняется через CM API, CM сразу materializes role delta для current members этой external group.

10.4 Как читать CM API

Этот раздел не заменяет Swagger. Полный справочник по полям request/response, кодам ошибок и актуальным examples находится в CM OpenAPI:

```
selena-cm/src/main/openapi/cm-api.yaml
{cm_url}/swagger-ui/index.html
{cm_url}/v3/api-docs
```

Здесь важно понять, какие операции нужны администратору и в каком порядке их обычно вызывать.

Что сделать	CM endpoint	Что делает в Selena
Посмотреть roles	GET /selena/roles	Читает live list roles из Engine.
Создать custom role	POST /selena/roles	Выполняет CREATE ROLE <roleName>.
Удалить custom role	DELETE /selena/roles/{roleName}	Выполняет DROP ROLE <roleName>.
Посмотреть grants role	GET /selena/roles/{roleName}/grants	Возвращает live SHOW GRANTS для role.
Выдать privilege role	POST /selena/roles/{roleName}/grants	Выполняет GRANT ... TO ROLE <roleName>.
Отозвать privilege у role	POST /selena/roles/{roleName}/grants:revoke	Выполняет REVOKE ... FROM ROLE <roleName>.
Посмотреть roles external group	GET /selena/external-groups/{externalGroup}/roles	Читает roles, выданные external group.

Что сделать	CM endpoint	Что делает в Selena
Привязать role к external group	<code>POST /selena/external-groups/{externalGroup}/roles</code>	Выполняет <code>GRANT <roleName> TO EXTERNAL GROUP <externalGroup></code> и сразу применяет delta к текущим users из snapshot.
Отвязать role от external group	<code>DELETE /selena/external-groups/{externalGroup}/roles/{roleName}</code>	Выполняет <code>REVOKE <roleName> FROM EXTERNAL GROUP <externalGroup></code> и сразу применяет delta к текущим users из snapshot.
Проверить snapshot групп	<code>GET /selena/membership-snapshot/status</code> , <code>GET /selena/membership-snapshot/groups</code> , <code>GET /selena/membership-snapshot/groups/{externalGroup}/members</code>	Показывает, какие Keycloak groups CM сейчас видит и в какие external groups они превращены.
Проверить себя как пользователя	<code>GET /me/direct-sql</code> , <code>GET /me/drift</code>	Показывает direct-SQL user state, expected roles и расхождения между expected/actual.

Обычный порядок настройки такой:

1. Создать role.
2. Выдать role нужные privileges на catalog/database/table/view/etc.
3. Проверить grants role.
4. Привязать role к external group.
5. Проверить members external group в membership snapshot.
6. Проверить пользователя через `/me/direct-sql`, `/me/drift` и direct SQL.

CM здесь не является отдельной моделью прав поверх Selena. Source of truth для roles/grants - Selena Engine: CM собирает безопасный SQL (`CREATE ROLE`, `GRANT`, `REVOKE`, `SHOW GRANTS`), выполняет его через admin connection и возвращает live result. Поэтому смысл privileges ровно тот же, что у прямого Selena `GRANT`.

Если прямой SQL grant работает, но эквивалентный CM request не проходит, это не "другая authorization semantics", а bug или gap в покрытии CM API. Типичные причины такого gap: CM еще не умеет выразить конкретный `scope`, object type не добавлен в enum, request не проходит проверку identifier safety или нужен `catalog` context для non-default catalog.

10.5 Request body для grant/revoke

`POST /selena/roles/{roleName}/grants` и `POST /selena/roles/{roleName}/grants:revoke` используют один формат:

```
{
  "privileges": ["SELECT"],
  "objectType": "TABLE",
  "objectQualifier": "mart.orders",
  "catalog": "lakekeeper_iceberg",
  "scope": "OBJECT",
  "withGrantOption": false
}
```

Поле	Нужно	Что значит
privileges	Да	Список Selena privilege names: <code>SELECT</code> , <code>INSERT</code> , <code>USAGE</code> , <code>CREATE TABLE</code> и т.д. СМ нормализует casing и собирает SQL, а совместимость privilege с object type проверяется тем же Selena Engine, что и при прямом <code>GRANT</code> .
objectType	Да	Тип объекта Selena: <code>SYSTEM</code> , <code>CATALOG</code> , <code>DATABASE</code> , <code>TABLE</code> , <code>VIEW</code> , <code>MATERIALIZED_VIEW</code> , <code>FUNCTION</code> , <code>GLOBAL _FUNCTION</code> , <code>RESOURCE_GROUP</code> , <code>RESOURCE</code> , <code>STORAGE_VOLUME</code> , <code>USER</code> .
objectQualifier	Обычно да	Имя объекта: catalog name, database name, <code>db.table</code> , storage volume name, username. Для <code>SYSTEM</code> не используется.
catalog	Нет	Catalog context для <code>DATABASE</code> , <code>TABLE</code> , <code>VIEW</code> , <code>MATERIALIZED_VIEW</code> , <code>FUNCTION</code> grants в non-default catalog. СМ перед <code>GRANT/REVOKE</code> выполняет <code>SET CATALOG <catalog></code> в admin session.
scope	Нет	По умолчанию <code>OBJECT</code> . Используется для grants вида "all tables in database".
withGrantOption	Нет	Разрешает получателю дальше выдавать такой privilege другим. Для обычных data roles почти всегда <code>false</code> .

Supported `scope` values:

Scope	Object type	objectQualifier
<code>OBJECT</code>	Все object types, кроме <code>SYSTEM</code>	Конкретный object: <code>default_catalog</code> , <code>mart</code> , <code>mart.orders</code> , <code>selena_s3_volume</code> , <code>alice</code> .
<code>ALL_DATABASES</code>	<code>DATABASE</code>	Не нужен.
<code>ALL_TABLES_IN_DATABASE</code>	<code>TABLE</code>	Database name.
<code>ALL_TABLES_IN_ALL_DATABASES</code>	<code>TABLE</code>	Не нужен.
<code>ALL_VIEWS_IN_DATABASE</code>	<code>VIEW</code>	Database name.
<code>ALL MATERIALIZED_VIEWS_IN_DATABASE</code>	<code>MATERIALIZED_VIEW</code>	Database name.
<code>ALL_FUNCTIONS_IN_DATABASE</code>	<code>FUNCTION</code>	Database name.

Если scope нельзя однозначно превратить в Selena SQL для выбранного object type, CM вернет 400. Если собранный GRANT syntactically valid для CM, но privilege не подходит object type или target object не существует, Selena Engine вернет такой же SQL error, как при прямом GRANT; CM только отдаст его в unified error response.

10.6 Object types и privileges

Ниже - практическая матрица для Selena/StarRocks 4.0-compatible RBAC. ALL означает "все privileges на объекте этого типа", а не "все в кластере".

SYSTEM

SYSTEM privileges управляют cluster-level возможностями, а не данными в таблицах.

Privilege	За что отвечает
CREATE RESOURCE GROUP	Создание resource groups.
CREATE RESOURCE	Создание resources для Spark Load/external tables.
CREATE EXTERNAL CATALOG	Создание external catalogs.
PLUGIN	Установка и удаление plugins.
REPOSITORY	Backup/restore repositories.
BLACKLIST	SQL blacklist и BE blacklist.
FILE	File objects.
OPERATE	Операционные действия: replicas, configs, variables, transactions, kill/query operations.
CREATE GLOBAL FUNCTION	Создание global UDF.
CREATE STORAGE VOLUME	Создание storage volumes.
SECURITY	Security integrations и group providers.

NODE и GRANT существуют в StarRocks privilege model, но не должны выдаваться напрямую custom roles: они принадлежат built-in administrative roles cluster_admin и user_admin.

CM request:

```
{
  "privileges": ["CREATE EXTERNAL CATALOG"],
  "objectType": "SYSTEM",
  "scope": "OBJECT"
}
```

CATALOG

Catalog - верхний уровень namespace. `default_catalog` - internal Selena catalog. External catalogs обычно указывают на Iceberg/Hive/Hudi/Delta и т.п.

Privilege	Internal catalog	External catalog	За что отвечает
<code>USAGE</code>	Да	Да	Позволяет использовать catalog как namespace. Для external catalog это базовый grant почти для любого data access.
<code>CREATE DATABASE</code>	Да	Ограниченно, зависит от catalog type	Создание databases в catalog.
<code>DROP</code>	Нет для <code>default_catalog</code>	Да	Удаление external catalog.
<code>ALL</code>	Да	Да	Все supported privileges на catalog.

Read-only external catalog почти всегда начинается с:

```
{
  "privileges": ["USAGE"],
  "objectType": "CATALOG",
  "objectQualifier": "lakekeeper_iceberg"
}
```

Сам по себе `USAGE ON CATALOG` не дает читать таблицы. Он дает право использовать catalog, но для чтения данных нужны grants на `TABLE`, `VIEW` или `MATERIALIZED_VIEW`.

DATABASE

Database privilege управляет самим database object и созданием объектов внутри него. Он не является table read/write privilege.

Privilege	За что отвечает
<code>ALTER</code>	Изменение свойств, rename, quotas.
<code>DROP</code>	Удаление database.
<code>CREATE TABLE</code>	Создание tables в database.
<code>CREATE VIEW</code>	Создание views.
<code>CREATE FUNCTION</code>	Создание database-level UDF.
<code>CREATE MATERIALIZED VIEW</code>	Создание materialized views.

Privilege	За что отвечает
ALL	Все database-level privileges.

Database admin role обычно требует не только `ALL ON DATABASE`, но и object grants на tables/views/materialized views/functions внутри database.

Пример:

```
{
  "privileges": ["CREATE TABLE", "CREATE VIEW"],
  "objectType": "DATABASE",
  "objectQualifier": "mart",
  "catalog": "lakekeeper_iceberg"
}
```

TABLE

Table privilege управляет чтением, записью и изменением tables.

Privilege	За что отвечает
SELECT	Чтение rows из table.
INSERT	Insert/load в table. Для <code>INSERT INTO ... SELECT ...</code> нужен еще <code>SELECT</code> на source tables.
UPDATE	Update rows.
DELETE	Delete rows.
EXPORT	Export data.
ALTER	Alter table; для external table также refresh metadata по StarRocks 4.0 docs.
DROP	Drop table.
ALL	Все table privileges.

Read-only на все tables database:

```
{
  "privileges": ["SELECT"],
  "objectType": "TABLE",
  "objectQualifier": "mart",
  "catalog": "lakekeeper_iceberg",
  "scope": "ALL_TABLES_IN_DATABASE"
}
```

Write на одну table:

```
{
  "privileges": ["SELECT", "INSERT"],
  "objectType": "TABLE",
  "objectQualifier": "mart.orders",
  "catalog": "lakekeeper_iceberg",
  "scope": "OBJECT"
}
```

`ALL_TABLES_IN_DATABASE` удобнее object-level grants, если в database будут появляться новые tables. Object-level grant на конкретную table привязан к этому object; при drop/recreate table privilege может потеряться.

VIEW

View privilege управляет logical views отдельно от base tables.

Privilege	За что отвечает
SELECT	Query existing view.
ALTER	Change view definition.
DROP	Drop view.
ALL	Все view privileges.

Grant на все views database:

```
{
  "privileges": ["SELECT"],
  "objectType": "VIEW",
  "objectQualifier": "mart",
  "catalog": "lakekeeper_iceberg",
  "scope": "ALL_VIEWS_IN_DATABASE"
}
```

Для создания view обычно нужен `CREATE VIEW` на database и `SELECT` на base tables, из которых строится view.

MATERIALIZED_VIEW

Materialized view privilege управляет отдельным materialized view object.

Privilege	За что отвечает
SELECT	Query materialized view напрямую и использовать его как acceleration target.
REFRESH	Refresh materialized view.
ALTER	Change materialized view.
DROP	Drop materialized view.
ALL	Все materialized view privileges.

Grant на refresh всех materialized views database:

```
{
  "privileges": ["REFRESH"],
  "objectType": "MATERIALIZED_VIEW",
  "objectQualifier": "mart",
  "catalog": "lakekeeper_iceberg",
  "scope": "ALL_MATERIALIZED_VIEWS_IN_DATABASE"
}
```

Для создания materialized view нужен CREATE MATERIALIZED VIEW на database и обычно SELECT на base tables.

FUNCTION и GLOBAL_FUNCTION

Database-level functions и global functions имеют похожую модель:

Object type	Privileges	За что отвечает
FUNCTION	USAGE, DROP, ALL	Database-level UDF.
GLOBAL_FUNCTION	USAGE, DROP, ALL	Global UDF.

СМ поддерживает object types FUNCTION и GLOBAL_FUNCTION, но function signatures в StarRocks syntax могут включать symbols вроде (,), ,. Если конкретный function grant нужен в production, сначала проверьте request на целевом СМ/Engine. Для большинства customer data-access ролей function grants не требуются.

Grant на все functions database:

```
{
  "privileges": ["USAGE"],
  "objectType": "FUNCTION",
  "objectQualifier": "mart",
  "catalog": "lakekeeper_iceberg",
  "scope": "ALL_FUNCTIONS_IN_DATABASE"
}
```

RESOURCE_GROUP

Resource group управляет resource governance/classifiers.

Privilege	За что отвечает
ALTER	Add/drop classifiers.
DROP	Drop resource group.
ALL	Все resource group privileges.

Обычным data users это не нужно.

RESOURCE

Resource используется для Spark Load и некоторых external integrations.

Privilege	За что отвечает
USAGE	Использовать resource.
ALTER	Изменить resource.
DROP	Удалить resource.
ALL	Все resource privileges.

Grant:

```
{
  "privileges": ["USAGE"],
  "objectType": "RESOURCE",
  "objectQualifier": "spark_resource"
}
```

STORAGE_VOLUME

Storage volume описывает remote storage credentials/config для shared-data, cloud-native storage и похожих сценариев.

Privilege	За что отвечает
USAGE	Describe storage volume и set as default storage volume.
ALTER	Change credentials/properties/comment/enabled status.
DROP	Drop storage volume.
ALL	Все storage volume privileges.

Обычным query users обычно не нужен ALTER или DROP на storage volume.

Grant:

```
{
  "privileges": ["USAGE"],
  "objectType": "STORAGE_VOLUME",
  "objectQualifier": "selena_s3_volume"
}
```

USER

USER object type сейчас используется для IMPERSONATE.

```
{
  "privileges": ["IMPERSONATE"],
  "objectType": "USER",
  "objectQualifier": "alice"
}
```

В Selena v1 external model не выдавайте IMPERSONATE business roles вручную. CM bootstrap и direct-SQL provisioning сами выдают нужные impersonation grants configured service users (cm, ide, ai) на canonical SQL users.

PIPE, WAREHOUSE, COLUMN

В upstream StarRocks privilege model есть дополнительные object types вроде PIPE, WAREHOUSE, COLUMN. Текущий public CM PrivilegeGrantRequest не имеет objectType

для них. Если customer deployment реально использует эти features, нужен отдельный product decision: расширить CM API или вести эти grants отдельным admin SQL runbook.

10.7 Типовые роли

Ниже приведены готовые fragments для `POST /selena/roles/{roleName}/grants`. Для выполнения используйте тот же `curl` формат, что в quick path выше, меняя только JSON body.

Read-only на internal database

Минимальный read-only для `default_catalog.mart`:

```
curl -fsS -X POST {cm_url}/selena/roles/mart_readonly/grants \
-H 'Authorization: Bearer {cm_admin_access_token}' \
-H 'Content-Type: application/json' \
-d '{
  "privileges": ["USAGE"],
  "objectType": "CATALOG",
  "objectQualifier": "default_catalog"
}' \
| jq
```

```
curl -fsS -X POST {cm_url}/selena/roles/mart_readonly/grants \
-H 'Authorization: Bearer {cm_admin_access_token}' \
-H 'Content-Type: application/json' \
-d '{
  "privileges": ["SELECT"],
  "objectType": "TABLE",
  "objectQualifier": "mart",
  "scope": "ALL_TABLES_IN_DATABASE"
}' \
| jq
```

Если users должны читать views или materialized views напрямую, добавьте отдельные grants на `VIEW` и `MATERIALIZED_VIEW`.

Read-only на external Iceberg/Lakekeeper catalog

Для external catalog важно дать `USAGE` на catalog и передавать `catalog` в нижестоящих grants:

```
curl -fsS -X POST {cm_url}/selena/roles/iceberg_readonly/grants \
-H 'Authorization: Bearer {cm_admin_access_token}' \
-H 'Content-Type: application/json' \
-d '{
  "privileges": ["USAGE"],
  "objectType": "CATALOG",
  "objectQualifier": "lakekeeper_iceberg"
}' \
| jq
```

```
curl -fsS -X POST {cm_url}/selena/roles/iceberg_readonly/grants \
-H 'Authorization: Bearer {cm_admin_access_token}' \
-H 'Content-Type: application/json' \
-d '{
  "privileges": ["SELECT"],
  "objectType": "TABLE",
  "objectQualifier": "mart",
  "catalog": "lakekeeper_iceberg",
  "scope": "ALL_TABLES_IN_DATABASE"
}' \
| jq
```

Writer на existing tables

Writer, который может читать и append/update/delete rows, но не создавать tables:

```
{
  "privileges": ["SELECT", "INSERT", "UPDATE", "DELETE"],
  "objectType": "TABLE",
  "objectQualifier": "mart",
  "catalog": "lakekeeper_iceberg",
  "scope": "ALL_TABLES_IN_DATABASE"
}
```

Для write-only ingestion service можно дать `INSERT` без `SELECT`, но тогда UI preview, validation queries и `INSERT INTO ... SELECT ...` из источника без дополнительных grants будут ломаться.

Database maintainer

Maintainer, который может создавать tables/views/materialized views и управлять объектами внутри database:

```
{
  "privileges": ["ALL"],
  "objectType": "DATABASE",
  "objectQualifier": "mart",
  "catalog": "lakekeeper_iceberg"
}
```

Плюс отдельно:

```
{
  "privileges": ["ALL"],
  "objectType": "TABLE",
  "objectQualifier": "mart",
  "catalog": "lakekeeper_iceberg",
  "scope": "ALL_TABLES_IN_DATABASE"
}
```

И, если нужно:

```
{
  "privileges": ["ALL"],
  "objectType": "VIEW",
  "objectQualifier": "mart",
  "catalog": "lakekeeper_iceberg",
  "scope": "ALL_VIEWS_IN_DATABASE"
}
```

```
{
  "privileges": ["ALL"],
  "objectType": "MATERIALIZED_VIEW",
  "objectQualifier": "mart",
  "catalog": "lakekeeper_iceberg",
  "scope": "ALL_MATERIALIZED_VIEWS_IN_DATABASE"
}
```

Причина: StarRocks FAQ explicitly calls out that database privileges do not make tables visible/readable by themselves. `SHOW TABLES` returns only tables where user has at least one table privilege.

10.8 Что будет в спорных комбинациях

Есть `USAGE ON CATALOG`, но нет `grants` на `database/table`

Пользователь получает право использовать `catalog namespace`. Но:

- `SELECT * FROM catalog.db.table` будет `denied`, потому что нет `SELECT` на `table/view/materialized view`;
- `SHOW TABLES` для `external catalog` требует `USAGE` на `catalog`, но без `table privileges` пользователь все равно не увидит `data objects`, на которые у него нет прав;
- BI/SQL tools могут показать `catalog`, но не показать `tables` или не дать выполнить `query`.

Практический вывод: `USAGE ON CATALOG` - это входная дверь в `catalog`, не доступ к данным.

Есть `grants` на `database`, но нет `grants` на `tables`

Например `role` имеет:

```
GRANT ALL ON DATABASE mart TO ROLE mart_admin
```

Но не имеет:

```
GRANT SELECT ON ALL TABLES IN DATABASE mart TO ROLE mart_admin
```

Тогда `role` может иметь `database-level capabilities` вроде `create/alter/drop database` или `create objects`, но `existing tables` не становятся `readable`. Это частая ошибка: `database grant` не наследуется автоматически на `tables`.

StarRocks FAQ отдельно говорит, что `SHOW TABLES` при `ALL ON DATABASE` может ничего не вернуть, если `user` не имеет `privileges` на сами `tables`.

Есть `grants` на `tables`, но нет `database-level CREATE TABLE`

Например `role` имеет:

```
GRANT SELECT, INSERT ON ALL TABLES IN DATABASE mart TO ROLE mart_writer
```

Но не имеет:

```
GRANT CREATE TABLE ON DATABASE mart TO ROLE mart_writer
```

Тогда role может читать/писать existing tables по table grants, но не может создавать новые tables в database. Это нормальная модель для ingestion users, которым нельзя менять schema.

Есть `CREATE TABLE ON DATABASE`, но нет table grants

Role может создавать tables в database, но это не заменяет grants на чтение, insert, update, delete или export данных. Для operational simplicity обычно сразу добавляют table-level grant pattern, например `ALL_TABLES_IN_DATABASE`, если role действительно является maintainer'ом этого database.

Есть table `SELECT`, но нет catalog `USAGE` на external catalog

Не делайте так в production. На external catalogs StarRocks docs отдельно требуют `USAGE` на соответствующем external catalog для metadata operations вроде `SHOW TABLES`. Даже если какой-то fully-qualified query пройдет на конкретной версии, SQL clients, UI metadata browsing и `SET CATALOG` будут вести себя непредсказуемо для пользователя.

Практический вывод: для external catalog всегда выдавайте:

```
USAGE ON CATALOG <catalog>  
SELECT/INSERT/... ON TABLE ... with catalog context
```

Есть `INSERT`, но нет `SELECT`

Такой user может быть write-only ingestion actor. Но:

- он не сможет прочесть rows для проверки;
- многие UI tools сначала делают `SELECT`/metadata queries и будут выглядеть "сломано";
- `INSERT INTO target SELECT ... FROM source` требует `SELECT` на source.

Для human writer обычно выдавайте `SELECT` вместе с `INSERT`.

Есть `SELECT` на table, но нет `SELECT` на view

View - отдельный object type. Если пользователь должен query'ить view, выдайте `SELECT` на `VIEW` или `ALL_VIEWS_IN_DATABASE`. Table grants не являются grants на view.

Есть `SELECT` на view, но нет direct `SELECT` на base table

Такой pattern может быть полезен как controlled access через curated views. Перед production use проверьте на целевом Selena Engine image, потому что view/base-table authorization details могут зависеть от версии и типа catalog.

Есть `REFRESH` на materialized view, но нет `SELECT`

`REFRESH` позволяет refresh operation, но не означает, что пользователь может читать materialized view. Для чтения нужен `SELECT` на `MATERIALIZED_VIEW`.

Role bound к external group, но user не получает access

Проверьте по цепочке:

1. Keycloak user реально состоит в expected group.
2. Group path совпадает с key в `cm.identity.external-groups.mappings`.
3. Mapping указывает на тот же `{external_group}`, к которому bound role.
4. `GET /selena/membership-snapshot/groups/{externalGroup}/members` содержит user.
5. `GET /me/drift` для user показывает expected role.
6. Если drift есть, дождитесь refresh или выполните user login/CLI reset path, который triggers reconcile.

Role имеет grants, но user видит no permission

Проверьте:

- role действительно materialized на SQL user: `SHOW GRANTS` или `GET /me/direct-sql`;
- CM при reconcile вызывает `SET DEFAULT ROLE ALL TO '<username>'@'%'`, но старая SQL session могла стартовать до изменения roles;
- переподключите MySQL/JDBC session;

- проверьте, что grant выдан на нужный catalog/database/table, а не на объект с похожим именем в другом catalog.

10.9 Revoke и удаление

Revoke privilege с role:

```
curl -fsS -X POST {cm_url}/selena/roles/analytics_readonly/grants:revoke \
-H 'Authorization: Bearer {cm_admin_access_token}' \
-H 'Content-Type: application/json' \
-d '{
  "privileges": ["SELECT"],
  "objectType": "TABLE",
  "objectQualifier": "mart",
  "catalog": "lakekeeper_iceberg",
  "scope": "ALL_TABLES_IN_DATABASE"
}' \
| jq
```

Revoke role from external group:

```
curl -fsS -X DELETE \
{cm_url}/selena/external-
groups/analytics_readers/roles/analytics_readonly \
-H 'Authorization: Bearer {cm_admin_access_token}' \
| jq
```

Delete role:

```
curl -i -sS -X DELETE {cm_url}/selena/roles/analytics_readonly \
-H 'Authorization: Bearer {cm_admin_access_token}'
```

Перед удалением role сначала снимите external group bindings и проверьте, что role не нужна другим группам. Если role используется в grants/bindings, Selena может вернуть SQL error.

10.10 Проверка результата

Проверить role grants через CM:

```
curl -fsS {cm_url}/selena/roles/{role_name}/grants \  
-H 'Authorization: Bearer {cm_admin_access_token}' \  
| jq
```

Проверить external group roles:

```
curl -fsS {cm_url}/selena/external-groups/{external_group}/roles \  
-H 'Authorization: Bearer {cm_admin_access_token}' \  
| jq
```

Проверить members external group:

```
curl -fsS {cm_url}/selena/membership-  
snapshot/groups/{external_group}/members \  
-H 'Authorization: Bearer {cm_admin_access_token}' \  
| jq
```

Проверить текущего пользователя:

```
curl -fsS {cm_url}/me/direct-sql \  
-H 'Authorization: Bearer {user_access_token}' \  
| jq  
  
curl -fsS {cm_url}/me/drift \  
-H 'Authorization: Bearer {user_access_token}' \  
| jq
```

Проверить direct SQL:

```
mysql --protocol=TCP \  
-h {selena_fe_mysql_host} \  
-P 9030 \  
-u {username} \  
-p{cli_secret} \  
-e "SELECT CURRENT_USER(); SELECT CURRENT_ROLE(); SHOW GRANTS;"
```

Если проверяете external catalog:

```
mysql --protocol=TCP \  
-h {selena_fe_mysql_host} \  
-P 9030 \  
-u {username} \  
-p{cli_secret} \  
-e "SET CATALOG lakekeeper_iceberg; SHOW TABLES FROM mart; SELECT * FROM  
mart.orders LIMIT 1;"
```

10.11 Практические правила

- Управляйте data access через roles и external groups, а не прямыми user grants.
- Для external catalog всегда выдавайте `USAGE ON CATALOG` плюс конкретные grants на tables/views/materialized views.
- Не считайте database grant заменой table grants.
- Для read-only human users обычно нужны `USAGE ON CATALOG`, `SELECT ON TABLE`, а если используются views/MVs - отдельные `SELECT ON VIEW` и `SELECT ON MATERIALIZED_VIEW`.
- Для writers обычно нужны `SELECT` и `INSERT`; `UPDATE/DELETE` выдавайте только если это реально требуется.
- `WITH GRANT OPTION` не нужен обычным business roles.
- Built-in roles `root`, `cluster_admin`, `db_admin`, `user_admin`, `security_admin` не используйте как business data roles.
- После изменения group membership учитывайте refresh interval CM.
- После изменения grants переподключайте long-lived SQL sessions при проверке.
- При сомнении смотрите live grants через CM или `SHOW GRANTS`, а не только values/config.

11 Состав поставки (Delivery Kit Contents)

Delivery-kit включает Helm charts и минимальные values-файлы. Charts делятся на две группы: **internal** (продуктовые компоненты Selena) и **external** (vendored open-source зависимости).

Для получения delivery-kit обращайтесь по адресу info@selena-lakehouse.ru.

11.1 Internal charts

Chart	Папка	Назначение
selena-operator	<code>charts/internal/selena-operator/</code>	Kubernetes Operator и CRD для управления Engine clusters. Не создаёт FE/BE/CN, только регистрирует типы ресурсов.
selena-cm	<code>charts/internal/selena-cm/</code>	Selena Cluster Manager — API для управления Engine, лицензиями, RBAC.
selena-ide	<code>charts/internal/selena-ide/</code>	Selena IDE — web-интерфейс, scheduler и worker.
selena-ai	<code>charts/internal/selena-ai/</code>	AI backend, AI frontend и Selena MCP.
selena-tls-proxy	<code>charts/internal/selena-tls-proxy/</code>	TLS-терминация и проксирование для компонентов Selena.

Каждый internal chart содержит:

- `chart/` — Helm chart (Chart.yaml, templates, values.yaml)
- `README.md` — краткое описание для внедренца
- `SOURCE.md` — источник и версия (для internal — ссылка на внутренний репозиторий)
- `SELENA_DEPLOY.md` — пошаговая инструкция установки

11.2 External charts

External charts ставятся **только если у заказчика нет готового сервиса**. Если сервис уже есть — в values передаются его endpoint и credentials.

Обязательные

Chart	Папка	Назначение
PostgreSQL	<code>charts/external/postgres/</code>	Основная БД для CM, Keycloak, IDE, AI. Upstream: <code>bitnami/postgresql</code> .

Chart	Папка	Назначение
Keycloak	charts/external/keycloak/	Identity provider (OIDC/OAuth2). Upstream: <code>codecentric/keycloakx</code> .

Опциональные

Chart	Папка	Назначение
OpenLDAP	charts/external/openldap/	Dev/demo LDAP-сервер для federation с Keycloak. Upstream: <code>helm-openldap/openldap-stack-ha</code> .
MinIO Operator	charts/external/minio/operator/	S3-compatible object storage operator. Upstream: <code>minio-operator/operator</code> .
MinIO Tenant	charts/external/minio/tenant/	S3 tenant (бакеты, политики). Upstream: <code>minio-operator/tenant</code> .
Lakekeeper	charts/external/lakekeeper/	Iceberg REST catalog. Upstream: <code>lakekeeper/lakekeeper</code> .
kube-prometheus-stack	charts/external/kube-prometheus-stack/	Мониторинг: Prometheus Operator, Prometheus, Alertmanager, Grafana, kube-state-metrics, node-exporter. Upstream: <code>prometheus-community/kube-prometheus-stack</code> .
Vault	charts/external/vault/	In-cluster Vault для примеров/демо. Upstream: <code>hashicorp/vault</code> .

Каждый external chart содержит:

- `chart/` — vendored upstream Helm chart
- `README.md` — описание для внедренца
- `SOURCE.md` — источник, версия, команда обновления
- `SELENA_DEPLOY.md` — пошаговая инструкция установки для Selena

11.3 Values-minimal

В поставку также входят минимальные values-файлы для быстрой установки всех компонентов:

```
docs-external/values-minimal/*.yaml
```

Они служат отправной точкой. Для production замените все secrets и проверьте resource limits, image tags, URL.

11.4 Порядок установки

1. External (обязательные): PostgreSQL, Keycloak.
2. External (опциональные): OpenLDAP, MinIO, Lakekeeper, kube-prometheus-stack, Vault.

3. Internal: selenator-operator, selenator-cm.
4. Bootstrap Engine через CM API.
5. Internal: selenator-ide, selenator-ai, selenator-tls-proxy.

Подробные пошаговые инструкции — в остальных документах этого руководства.

12 Values-prod — самодостаточные values для production

Файлы здесь = содержимое `../values-minimal/` **плюс** prod-тюнинг размещения (`resources` requests/limits, `tolerations`, и закомментированные `nodeSelector`/affinity). Каждый файл самодостаточен — подключается **одним** `-f`, второй файл не нужен.

12.1 Как применять

Один `-f` на чарт. Переключение dev ↔ prod — это просто смена директории:

```
# dev / минимальный профиль
helm install selenacm {chart_path} -f ../values-minimal/selenacm-values.yaml

# production
helm install selenacm {chart_path} -f ../values-prod/selenacm-values.yaml
```

Имена файлов в `values-minimal/` и `values-prod/` совпадают, поэтому в скрипте достаточно одной переменной с путём к директории.

Бизнес-логика, URL-ы и Secrets в обоих наборах идентичны — отличается только блок `--- prod tuning ---`. Если правите URL/Secret, синхронизируйте оба файла (или генерируйте `values-prod` из `values-minimal`).

12.2 Что активно, а что закомментировано

Блок	Состояние	Почему
<code>resources</code>	активен , конкретные стартовые числа	Безопасен при любом значении; числа — стартовая точка под правку.
<code>tolerations</code>	активен , с плейсхолдерами	Toleration на несуществующий taint безвреден, поды не ломаются.
<code>nodeSelector</code>	закомментирован	Незаполненный label оставит поды в <code>Pending</code> . Раскомментируйте после замены.
<code>affinity</code>	закомментирован	Требует label'ов конкретного чарта/кластера; неверное значение блокирует scheduling.

12.3 Плейсхолдеры

Замените перед установкой (формат `{var}`):

Placeholder	Значение
<code>{node_label_key}</code> / <code>{node_label_value}</code>	Label узлов целевого node pool, например <code>workload=selena</code> .
<code>{node_taint_key}</code> / <code>{node_taint_value}</code>	Taint узлов целевого node pool, например <code>dedicated=selena</code> .

Плюс все обычные плейсхолдеры основного guide (`{storage_class}`, `{registry}`, URL-ы Keycloak и т.д.) — они унаследованы из `minimal`-части.

12.4 Где какой ключ (схема у чартов разная)

Чарт	resources	scheduling
postgres	<code>primary.resources</code>	<code>primary.*</code>
keycloak	<code>resources</code> (top)	top-level
minio-operator	<code>operator.resources</code>	<code>operator.*</code>
minio-tenant	<code>tenant.pools[0].resources</code>	per-pool
lakekeeper	<code>catalog.resources</code>	<code>catalog.*</code>
openldap	<code>resources</code> (top)	top-level
vault	<code>server.resources</code>	<code>server.*</code>
kube-prometheus-stack	<code>prometheus.prometheusSpec</code> / <code>alertmanager.alertmanagerSpec</code> / <code>grafana</code> / <code>prometheusOperator</code>	те же scope'ы
selena-operator	<code>selenaOperator.resources</code>	<code>selenaOperator.*</code>
selena-cm	<code>resources</code> (top)	top-level
selena-ide	<code>web/worker/scheduler.resources</code>	top-level (общие)
selena-ai	<code>components.{backend,frontend,mcp}.resources</code>	top-level (общие)

12.5 Стартовые resources — не финальные

Числа — разумная отправная точка для среднего prod-стенда; правьте по факту (`kubectl top pod`, метрики Prometheus). Нюансы:

- **JVM** (Keycloak, CM): memory `limit` должен быть заметно больше heap (`-Xmx`), иначе OOMKill.

- **Stateful** (PostgreSQL, MinIO): CPU `limit` может throttle'ить под пиком — часто оставляют только requests + memory limit.
- **node-exporter** (kube-prometheus-stack): не пиннить — DaemonSet на все узлы.
- **minio-tenant**: minimal-геометрия 1 server / 1 volume = нет erasure coding; для prod ≥ 4 дисков.

12.6 Что ещё стоит усилить для prod (не входит в эти файлы)

PodDisruptionBudget; `topologySpreadConstraints`; `priorityClassName`; HPA/autoscaling для stateless; `securityContext`/`podSecurityContext`; managed/HA PostgreSQL + бэкапы вместо standalone-SPOF; Prometheus retention/Thanos; NetworkPolicy; `startupProbe` для медленных JVM.

12.7 Keycloak — values

Файл values для чарта `keycloak`.

keycloak-values.yaml

```
# Имя Keycloak Service/StatefulSet для команд из этой документации.
fullnameOverride: selena-keycloak

# Public hostname Keycloak; должен совпадать с issuer, redirect и logout
URLs.
args:
  # Production start mode Keycloak.
  - start
  # Вставьте внешний URL Keycloak, например https://keycloak.example.com.
  - --hostname={keycloak_public_url}
  # Разрешает работу за Ingress/LB, где Host приходит через proxy headers.
  - --hostname-strict=false

# Realm URL должен быть `{keycloak_public_url}/realms/selena`, без legacy
`/auth`.
http:
  # "/" убирает старый prefix `/auth` из публичных Keycloak URLs.
  relativePath: "/"

# Настройки proxy-aware режима Keycloak.
proxy:
  # Keycloak читает X-Forwarded-* headers от Ingress/LB.
  mode: xforwarded
  # TLS обычно завершается снаружи, внутри кластера Keycloak слушает HTTP.
  http:
    # Включает HTTP listener внутри Kubernetes.
    enabled: true

# Kubernetes Service Keycloak.
service:
  # Отключает отдельный HTTPS service port в minimal profile.
  httpsPort: null

# Bootstrap admin читается из Secret, чтобы пароль не лежал в values.yaml.
extraEnv: |
  - name: KC_BOOTSTRAP_ADMIN_USERNAME
    valueFrom:
      secretKeyRef:
        name: selena-keycloak-admin
        key: admin-user
  - name: KC_BOOTSTRAP_ADMIN_PASSWORD
    valueFrom:
      secretKeyRef:
```

```
    name: selena-keycloak-admin
    key: admin-password

# Подключение Keycloak к PostgreSQL.
database:
  # Selena deployment использует PostgreSQL как persistent storage
  Keycloak.
  vendor: postgres
  # DNS/service host PostgreSQL.
  hostname: "{postgres_service_host}"
  # PostgreSQL port.
  port: 5432
  # Database для Keycloak.
  database: keycloak
  # DB user, созданный bundled PostgreSQL chart-ом.
  username: selena
  # Secret с DB password.
  existingSecret: selena-keycloak-db
  # Key внутри Secret с DB password.
  existingSecretKey: password

# --- prod tuning (resources/scheduling) ---
# Keycloak — JVM. memory limit должен быть заметно больше heap (-Xmx),
# иначе OOMKill.
# Согласуйте с JAVA_OPTS_APPEND.
resources:
  requests:
    cpu: "1"
    memory: "1Gi"
  limits:
    cpu: "2"
    memory: "2Gi"
# Toleration безвреден, даже если taint не задан.
#tolerations:
# - key: "{node_taint_key}"
#   operator: "Equal"
#   value: "{node_taint_value}"
#   effect: "NoSchedule"
# Раскомментируйте после замены плейсхолдеров (иначе под останется
# Pending).
# nodeSelector:
#   "{node_label_key}": "{node_label_value}"
# Для HA задайте soft podAntiAffinity по hostname и:
# replicas: 2
# podDisruptionBudget:
```

```
# minAvailable: 1  
# affinity: {}
```

12.8 Kube-prometheus-stack — values

Файл values для чарта `kube-prometheus-stack`.

kube-prometheus-stack-values.yaml

```
# Стабильное имя monitoring resources.
fullnameOverride: selena-monitoring

# CRD Prometheus Operator.
crds:
  # true ставит ServiceMonitor/PodMonitor/PrometheusRule CRD вместе с
  chart.
  enabled: true

# Prometheus server.
prometheus:
  # Спецификация Prometheus instance.
  prometheusSpec:
    # Сколько хранить metrics history.
    retention: 15d
    # --- prod tuning: resources/scheduling Prometheus (масштабируется с
    числом series) ---
    resources:
      requests:
        cpu: "500m"
        memory: "2Gi"
      limits:
        cpu: "2"
        memory: "4Gi"
    # tolerations:
    #   - key: "{node_taint_key}"
    #     operator: "Equal"
    #     value: "{node_taint_value}"
    #     effect: "NoSchedule"
    # nodeSelector:
    #   "{node_label_key}": "{node_label_value}"
    # affinity: {}
    # false позволяет Prometheus видеть ServiceMonitor из Selena
    namespaces.
    serviceMonitorSelectorNilUsesHelmValues: false
    # false позволяет Prometheus видеть PodMonitor из Selena namespaces.
    podMonitorSelectorNilUsesHelmValues: false
    # false позволяет Prometheus видеть PrometheusRule из Selena
    namespaces.
    ruleSelectorNilUsesHelmValues: false
    # false позволяет Prometheus видеть Probe из Selena namespaces.
    probeSelectorNilUsesHelmValues: false
    # false позволяет Prometheus видеть ScrapeConfig из Selena namespaces.
```

```
scrapeConfigSelectorNilUsesHelmValues: false
# {} разрешает искать ServiceMonitor во всех namespaces.
serviceMonitorNamespaceSelector: {}
# {} разрешает искать PodMonitor во всех namespaces.
podMonitorNamespaceSelector: {}
# {} разрешает искать PrometheusRule во всех namespaces.
ruleNamespaceSelector: {}
# {} разрешает искать Probe во всех namespaces.
probeNamespaceSelector: {}
# {} разрешает искать ScrapeConfig во всех namespaces.
scrapeConfigNamespaceSelector: {}
# Persistent storage Prometheus.
storageSpec:
  # PVC template для Prometheus TSDB.
  volumeClaimTemplate:
    # PVC spec.
    spec:
      # StorageClass кластера.
      storageClassName: "{storage_class}"
      # Запрашиваемые ресурсы PVC.
      resources:
        # Запрошенный размер диска.
        requests:
          # Размер Prometheus TSDB для bundled minimal profile.
          storage: 64Gi

# Alertmanager.
alertmanager:
  # Спецификация Alertmanager instance.
  alertmanagerSpec:
    # --- prod tuning: resources/scheduling Alertmanager ---
    resources:
      requests:
        cpu: "50m"
        memory: "128Mi"
      limits:
        cpu: "200m"
        memory: "256Mi"
    #
    tolerations:
    #   - key: "{node_taint_key}"
    #     operator: "Equal"
    #     value: "{node_taint_value}"
    #     effect: "NoSchedule"
    # nodeSelector:
    #   "{node_label_key}": "{node_label_value}"
    # affinity: {}
```

```
# Persistent storage Alertmanager.
storage:
  # PVC template для Alertmanager data.
  volumeClaimTemplate:
    # PVC spec.
    spec:
      # StorageClass кластера.
      storageClassName: "{storage_class}"
      # Запрашиваемые ресурсы PVC.
      resources:
        # Запрошенный размер диска.
        requests:
          # Размер Alertmanager PVC для bundled minimal profile.
          storage: 8Gi

# Grafana UI.
grafana:
  # Стабильное имя Grafana resources.
  fullnameOverride: selenagrafana
  # --- prod tuning: resources/scheduling Grafana ---
  resources:
    requests:
      cpu: "100m"
      memory: "256Mi"
    limits:
      cpu: "500m"
      memory: "512Mi"
  # tolerations:
  #   - key: "{node_taint_key}"
  #     operator: "Equal"
  #     value: "{node_taint_value}"
  #     effect: "NoSchedule"
  # nodeSelector:
  #   "{node_label_key}": "{node_label_value}"
  # affinity: {}
  # Recreate избегает одновременного монтирования одного RWO PVC двумя pod-ами.
  deploymentStrategy:
    # Strategy deployment-a Grafana.
    type: Recreate
  # Local admin user Grafana.
  admin:
    # Secret с admin username/password.
    existingSecret: selenagrafana-admin
    # Key внутри Secret с admin username.
    userKey: admin-user
```

```
# Key внутри Secret с admin password.
passwordKey: admin-password
# Secret env vars для Grafana.
envValueFrom:
  # OAuth client secret нельзя хранить в ConfigMap.
  GF_AUTH_GENERIC_OAUTH_CLIENT_SECRET:
    # Key берется из Kubernetes Secret.
    secretKeyRef:
      # Secret с Keycloak client secret для client `grafana`.
      name: selena-grafana-keycloak-client
      # Key внутри Secret.
      key: client-secret
# Persistent storage Grafana.
persistence:
  # true сохраняет dashboards/settings между restart-ами.
  enabled: true
  # StorageClass кластера.
  storageClassName: "{storage_class}"
  # Размер PVC Grafana.
  size: 16Gi
# Grafana INI config.
grafana.ini:
  # Public server URL.
  server:
    # Public root URL Grafana; должен совпадать с Keycloak
    redirect/logout URLs.
    root_url: "{grafana_public_url}/"
  # User defaults.
  users:
    # true автоматически добавляет OAuth users в default org.
    auto_assign_org: true
    # Базовая роль; strict role mapping ниже запретит вход без Grafana
    role.
    auto_assign_org_role: Viewer
  # Общие настройки auth.
  auth:
    # true убирает local login form, чтобы вход был только через
    Keycloak.
    disable_login_form: true
    # true сразу отправляет пользователя в Keycloak OAuth flow.
    oauth_auto_login: true
  # Local basic auth.
  auth.basic:
    # false запрещает basic auth в UI.
    enabled: false
  # Anonymous auth.
```

```
auth.anonymous:
  # false запрещает анонимный доступ.
  enabled: false
# Generic OAuth provider Keycloak.
auth.generic_oauth:
  # true включает вход через Keycloak.
  enabled: true
  # Имя provider-а на стороне Grafana.
  name: Keycloak
  # true разрешает создавать Grafana user при первом OAuth login.
  allow_sign_up: true
  # Keycloak client id для Grafana.
  client_id: grafana
  # Grafana подставляет secret из env; это не shell placeholder.
  client_secret: $__env{GF_AUTH_GENERIC_OAUTH_CLIENT_SECRET}
  # OIDC scopes, которые Grafana запрашивает у Keycloak.
  scopes: openid profile email
  # Browser authorization endpoint Keycloak.
  auth_url: "{keycloak_public_url}/realms/selena/protocol/openid-
connect/auth"
  # Token endpoint Keycloak.
  token_url: "{keycloak_public_url}/realms/selena/protocol/openid-
connect/token"
  # UserInfo endpoint Keycloak.
  api_url: "{keycloak_public_url}/realms/selena/protocol/openid-
connect/userinfo"
  # Logout redirect с client_id и post_logout_redirect_uri.
  signout_redirect_url: "
{keycloak_public_url}/realms/selena/protocol/openid-connect/logout?
client_id=grafana&post_logout_redirect_uri={grafana_login_url_encoded}"
  # true включает PKCE для OAuth code flow.
  use_pkce: true
  # Mapping Keycloak realm roles в Grafana org role.
  role_attribute_path: contains(realm_access.roles[*], 'grafana_admin')
&& 'Admin' || contains(realm_access.roles[*], 'grafana_viewer') && 'Viewer'
|| ''
  # true запрещает вход пользователю без grafana_admin/grafana_viewer.
  role_attribute_strict: true
  # Claim email из Keycloak.
  email_attribute_path: email
  # Claim username из Keycloak.
  login_attribute_path: preferred_username
  # Claim display name из Keycloak.
  name_attribute_path: name

# Selena dashboards, поставляемые вместе с chart.
```

```
selenaDashboards:
  # true устанавливает Selena dashboards ConfigMaps.
  enabled: true
  # Default folder в Grafana.
  folder: Selena
  # Список dashboard ConfigMaps.
  items:
    # Platform overview dashboard.
    - name: selena-platform-overview-dashboard
      # JSON dashboard file внутри chart.
      file: dashboards/selena-platform-overview.json
      # Folder dashboard-a в Grafana.
      folder: Selena
    # Engine dashboard.
    - name: selena-engine-dashboard
      # JSON dashboard file внутри chart.
      file: dashboards/selena-engine-dashboard.json
      # Folder dashboard-a в Grafana.
      folder: Selena Engine

# --- prod tuning: Prometheus Operator (новый top-level блок) ---
prometheusOperator:
  resources:
    requests:
      cpu: "100m"
      memory: "128Mi"
    limits:
      cpu: "200m"
      memory: "256Mi"
  # tolerations:
  #   - key: "{node_taint_key}"
  #     operator: "Equal"
  #     value: "{node_taint_value}"
  #     effect: "NoSchedule"
  # nodeSelector:
  #   "{node_label_key}": "{node_label_value}"
  # affinity: {}

# node-exporter — DaemonSet, должен идти на ВСЕ узлы (включая tainted),
# поэтому
# не задавайте ему nodeSelector. При необходимости расширьте его
# tolerations:
# prometheus-node-exporter:
#   resources:
#     requests: { cpu: "50m", memory: "64Mi" }
#     limits: { cpu: "200m", memory: "128Mi" }
```

```
# tolerations:  
#   - operator: "Exists"
```

12.9 Lakekeeper — values

Файл values для чарта `lakekeeper`.

lakekeeper-values.yaml

```
# Стабильное имя Lakekeeper Service/Deployment для команд из этой
# документации.
fullnameOverride: selena-lakekeeper

# Встроенный PostgreSQL из Lakekeeper chart выключен: используется общий
# PostgreSQL.
postgresql:
  # false означает использовать `externalDatabase.*`.
  enabled: false

# Подключение Lakekeeper к PostgreSQL.
externalDatabase:
  # Тип внешней DB.
  type: postgres
  # Read host PostgreSQL.
  host_read: "{postgres_service_host}"
  # Write host PostgreSQL.
  host_write: "{postgres_service_host}"
  # PostgreSQL port.
  port: 5432
  # Database для Lakekeeper.
  database: lakekeeper
  # Secret с DB username.
  userSecret: selena-lakekeeper-db
  # Key внутри Secret с DB username.
  userSecretKey: postgresql-user
  # Secret с DB password.
  passwordSecret: selena-lakekeeper-db
  # Key внутри Secret с DB password.
  passwordSecretKey: postgresql-password

# Backend для секретов Lakekeeper.
secretBackend:
  # Postgres backend хранит Lakekeeper secrets в DB.
  type: Postgres
  # Настройки Postgres secret backend.
  postgres:
    # Secret со stable encryption key.
    encryptionKeySecret: selena-lakekeeper-postgres-encryption
    # Key внутри Secret с encryption key.
    encryptionKeySecretKey: encryptionKey

# --- prod tuning (resources/scheduling) ---
```

```
# Lakekeeper catalog – stateless; реплики безопасно масштабировать
# (catalog.replicas).
catalog:
  resources:
    requests:
      cpu: "250m"
      memory: "512Mi"
    limits:
      cpu: "1"
      memory: "1Gi"
  # Toleration безвреден, даже если taint не задан.
  # tolerations:
  #   - key: "{node_taint_key}"
  #     operator: "Equal"
  #     value: "{node_taint_value}"
  #     effect: "NoSchedule"
  # Раскомментируйте после замены плейсхолдеров (иначе под останется
  # Pending).
  # nodeSelector:
  #   "{node_label_key}": "{node_label_value}"
  # affinity: {}
```

12.10 Minio-operator — values

Файл values для чарта `minio-operator`.

minio-operator-values.yaml

```
# MinIO Operator работает на upstream chart defaults.
# Этот файл намеренно не содержит overrides: заказчик ничего не обязан
менять.
# Если нужен private registry/mirror, добавьте сюда `operator.image.*`.

# --- prod tuning (resources/scheduling) ---
# MinIO Operator – лёгкий control-plane контроллер.
operator:
  resources:
    requests:
      cpu: "100m"
      memory: "128Mi"
    limits:
      cpu: "200m"
      memory: "256Mi"
# Toleration безвреден, даже если taint не задан.
tolerations:
  - key: "{node_taint_key}"
    operator: "Equal"
    value: "{node_taint_value}"
    effect: "NoSchedule"
# Раскомментируйте после замены плейсхолдеров (иначе под останется
Pending).
# nodeSelector:
#   "{node_label_key}": "{node_label_value}"
# affinity: {}
```

12.11 Minio-tenant — values

Файл values для чарта `minio-tenant`.

minio-tenant-values.yaml

```
# MinIO Tenant дает S3-compatible storage для Selena lakehouse сценариев.
tenant:
  # Стабильное имя tenant-a.
  name: selena-minio

  # Secret с MinIO root credentials.
  configSecret:
    # Имя Secret, который создается администратором до Helm install.
    name: selena-minio-env-configuration
    # true говорит chart-у использовать уже существующий Secret.
    existingSecret: true

  # Storage pools tenant-a.
  pools:
    # Minimal pool: один server и один volume.
    - servers: 1
      # Имя pool-a.
      name: pool-0
      # Количество volumes на server.
      volumesPerServer: 1
      # Размер каждого PVC.
      size: 64Gi
      # StorageClass кластера.
      storageClassName: "{storage_class}"
      # --- prod tuning (resources/scheduling) ---
      # ВАЖНО: ресурсы/размещение MinIO Tenant задаются ВНУТРИ pool.
      # Minimal-геометрия = 1 server / 1 volume (нет erasure coding). Для
      prod >=4 дисков.
      resources:
        requests:
          cpu: "500m"
          memory: "2Gi"
        limits:
          # CPU limit на storage может throttle'ить под пиком I/O – при
          необходимости уберите.
          cpu: "2"
          memory: "4Gi"
      # Toleration безвреден, даже если taint не задан.
      tolerations:
        - key: "{node_taint_key}"
          operator: "Equal"
          value: "{node_taint_value}"
          effect: "NoSchedule"
```

```
# Раскомментируйте после замены плейсхолдеров (иначе под останется Pending).
# nodeSelector:
#   "{node_label_key}": "{node_label_value}"
# affinity: {}

# Buckets, которые нужны Selenia stack.
buckets:
  # Общий bucket для Selenia artifacts/examples.
  - name: selenia
  # Warehouse bucket для lake/iceberg data.
  - name: warehouse

# MinIO users, которые создаст tenant.
users:
  # User Secret должен существовать и содержать access/secret keys.
  - name: selenia-minio-user
```

12.12 Openldap — values

Файл values для чарта `openldap`.

openldap-values.yaml

```
# OpenLDAP приведен как пример для test/demo; в production обычно
используйте LDAP/AD заказчика.

# Стабильное имя OpenLDAP Service/StatefulSet для команд из этой
документации.
fullnameOverride: selenal-openldap

# Общие LDAP settings chart-a.
global:
  # LDAP domain, например example.org; chart преобразует его в base DN.
  ldapDomain: "{ldap_domain}"
  # Secret с LDAP admin/config passwords.
  existingSecret: selenal-openldap-auth

# Один pod достаточно для demo LDAP.
replicaCount: 1

# LDAP replication отключена в demo profile.
replication:
  # false упрощает test/demo установку.
  enabled: false

# Kubernetes Service OpenLDAP.
service:
  # LDAPS port отключен, потому что demo LDAP используется только внутри
кластера.
  enableSslLdapPort: false

# LDAP runtime env vars для demo profile.
env:
  # no отключает TLS listener в demo LDAP.
  LDAP_ENABLE_TLS: "no"

# Persistent storage LDAP data.
persistence:
  # StorageClass кластера.
  storageClass: "{storage_class}"
  # Размер PVC LDAP demo directory.
  size: 16Gi

# LTB password self-service UI.
ltb-passwd:
  # false, потому что пользователи входят через Keycloak, а не через
```

```
отдельный LDAP UI.  
  enabled: false  
  
# phpLDAPadmin UI.  
phpldapadmin:  
  # false, чтобы minimal profile не публиковал лишний LDAP admin UI.  
  enabled: false  
  
# --- prod tuning (resources/scheduling) ---  
# Напоминание: OpenLDAP здесь demo. В production обычно подключают LDAP/AD  
# заказчика.  
resources:  
  requests:  
    cpu: "100m"  
    memory: "128Mi"  
  limits:  
    cpu: "500m"  
    memory: "512Mi"  
# Toleration безвреден, даже если taint не задан.  
#tolerations:  
# - key: "{node_taint_key}"  
#   operator: "Equal"  
#   value: "{node_taint_value}"  
#   effect: "NoSchedule"  
# Раскомментируйте после замены плейсхолдеров (иначе под останется  
# Pending).  
# nodeSelector:  
#   "{node_label_key}": "{node_label_value}"  
# affinity: {}
```

12.13 Postgres — values

Файл values для чарта `postgres`.

postgres-values.yaml

```
# Стабильное имя PostgreSQL Service/StatefulSet для команд из этой
# документации.
fullnameOverride: selenapostgres

# PostgreSQL users/passwords.
auth:
  # Application DB user, которым будут пользоваться Selena components.
  username: selenapostgres
  # Default DB, которую chart создает при первом старте.
  database: selenapostgres
  # Secret с PostgreSQL passwords.
  existingSecret: selenapostgres-auth

# Primary PostgreSQL node.
primary:
  # Persistent storage primary node.
  persistence:
    # StorageClass кластера.
    storageClass: "{storage_class}"
    # Размер PVC PostgreSQL для bundled minimal profile.
    size: 64Gi
  # Init scripts выполняются только при первом создании DB volume.
  initdb:
    # Набор SQL scripts.
    scripts:
      # Скрипт создает databases, которые нужны bundled Selena stack.
      01-create-selena-databases.sql: |
        SELECT 'CREATE DATABASE keycloak'
        WHERE NOT EXISTS (SELECT FROM pg_database WHERE datname =
        'keycloak')\gexec

        SELECT 'CREATE DATABASE lakekeeper'
        WHERE NOT EXISTS (SELECT FROM pg_database WHERE datname =
        'lakekeeper')\gexec

        SELECT 'CREATE DATABASE selena_cm'
        WHERE NOT EXISTS (SELECT FROM pg_database WHERE datname =
        'selena_cm')\gexec

        SELECT 'CREATE DATABASE selena_ide'
        WHERE NOT EXISTS (SELECT FROM pg_database WHERE datname =
        'selena_ide')\gexec
```

```
SELECT 'CREATE DATABASE selena_ai'
WHERE NOT EXISTS (SELECT FROM pg_database WHERE datname =
'selena_ai')\gexec

GRANT ALL PRIVILEGES ON DATABASE keycloak TO selena;
GRANT ALL PRIVILEGES ON DATABASE lakekeeper TO selena;
GRANT ALL PRIVILEGES ON DATABASE selena_cm TO selena;
GRANT ALL PRIVILEGES ON DATABASE selena_ide TO selena;
GRANT ALL PRIVILEGES ON DATABASE selena_ai TO selena;

# --- prod tuning (resources/scheduling) ---
# Bitnami по умолчанию ставит resourcesPreset: "nano"; явный resources
его перекрывает.
resources:
  requests:
    cpu: "500m"
    memory: "1Gi"
  limits:
    # CPU limit на БД может вызывать throttling под пиком — при
    необходимости уберите.
    cpu: "2"
    memory: "4Gi"
# Toleration безвреден, даже если taint не задан.
#tolerations:
# - key: "{node_taint_key}"
#   operator: "Equal"
#   value: "{node_taint_value}"
#   effect: "NoSchedule"
# Раскомментируйте после замены плейсхолдеров (иначе под останется
Pending).
# nodeSelector:
#   "{node_label_key}": "{node_label_value}"
# Bitnami уже включает podAntiAffinityPreset: soft. Для жёсткого
разнесения:
# affinity: {}
```

12.14 Selena-ai — values

Файл values для чарта `selena-ai`.

selena-ai-values.yaml

```
# Secret для доступа к private registry; удалите блок, если registry
# публичный.
imagePullSecrets:
  # Имя docker-registry Secret в namespace Selena applications.
  - name: selena-registry

# Selena AI chart ставит backend, frontend и MCP.
components:
  # Backend хранит AI sessions, проверяет auth и вызывает LLM/MCP.
  backend:
    # Образ Selena AI backend.
    image:
      # Repository AI backend image; меняйте, если используете внутренний
      # registry/mirror.
      repository: "hub.selena.sh/selena/selena-ai-backend"
      # Точный release tag AI backend image, который вы устанавливаете.
      tag: "5.0.15"
    # Только параметры, которые зависят от LLM provider и URL-ов вашего
    # стенда.
    env:
      # Base URL LLM provider endpoint.
      AI_BACKEND_LLM_BASE_URL: "{llm_base_url}"
      # Имя LLM model, которую будет использовать AI Agent.
      AI_BACKEND_LLM_MODEL: "{llm_model}"
      # Public issuer Keycloak realm, ожидаемый backend в user tokens.
      AI_BACKEND_AUTH_ISSUER_URL: "{keycloak_public_url}/realms/selena"
      # Browser authorization endpoint Keycloak для AI login/callback.
      AI_BACKEND_AUTH_AUTHORIZATION_URL: "
{keycloak_public_url}/realms/selena/protocol/openid-connect/auth"
      # Internal token endpoint Keycloak для OAuth code exchange.
      AI_BACKEND_AUTH_TOKEN_URL: "
{keycloak_internal_url}/realms/selena/protocol/openid-connect/token"
      # Internal JWKS endpoint для проверки JWT signatures.
      AI_BACKEND_AUTH_JWKS_URL: "
{keycloak_internal_url}/realms/selena/protocol/openid-connect/certs"
      # Public logout endpoint Keycloak.
      AI_BACKEND_AUTH_LOGOUT_URL: "
{keycloak_public_url}/realms/selena/protocol/openid-connect/logout"
      # Redirect URI, зарегистрированный в Keycloak client `selena-ai`.
      AI_BACKEND_AUTH_REDIRECT_URI: "{ai_public_url}/api/v1/auth/callback"
      # Куда отправлять browser после успешного AI login.
      AI_BACKEND_AUTH_POST_LOGIN_REDIRECT_URI: "{ai_public_url}/"
      # Куда отправлять browser после logout.
```

```
AI_BACKEND_AUTH_POST_LOGOUT_REDIRECT_URI: "{ai_public_url}/"
# Browser-reachable AI iframe URL для cookies, redirects и CORS.
AI_BACKEND_AUTH_FRONTEND_URL: "{ai_public_url}"
# Разрешенный origin IDE, откуда browser может обращаться к AI
backend.
AI_BACKEND_CORS_ORIGINS: "{ide_public_url}"
# Secret env vars для backend.
secrets:
# Имя Kubernetes Secret с DB URL, LLM key и OIDC secrets.
existingSecret: selenai-secrets
# Keys из Secret, которые chart пробросит как env vars.
envKeys:
# PostgreSQL connection string AI backend.
- AI_BACKEND_DATABASE_URL
# API key LLM provider; не хранится в ConfigMap.
- AI_BACKEND_LLM_API_KEY
# Secret Keycloak browser client `selenai`.
- AI_BACKEND_AUTH_CLIENT_SECRET
# Stable encryption key для AI tokens/sessions.
- AI_BACKEND_AUTH_TOKEN_ENCRYPTION_KEY
# --- prod tuning: resources backend ---
resources:
  requests:
    cpu: "250m"
    memory: "512Mi"
  limits:
    cpu: "1"
    memory: "1Gi"

# Frontend отдает browser UI AI Agent.
frontend:
# Образ Selena AI frontend.
image:
# Repository AI frontend image; меняйте, если используете внутренний
registry/mirror.
repository: "hub.selena.sh/selena/selena-ai-frontend"
# Точный release tag AI frontend image, который вы устанавливаете.
tag: "5.0.19"
# --- prod tuning: resources frontend ---
resources:
  requests:
    cpu: "100m"
    memory: "128Mi"
  limits:
    cpu: "300m"
    memory: "256Mi"
```

```
# MCP server исполняет AI tool calls в Selena.
mcp:
  # Образ Selena MCP.
  image:
    # Repository MCP image; меняйте, если используете внутренний
registry/mirror.
    repository: "hub.selena.sh/selena/selena-mcp"
    # Точный release tag MCP image, который вы устанавливаете.
    tag: "latest"
  # Только параметры, которые зависят от Engine cluster и Keycloak URL-
ОВ.
  env:
    # Internal FE service DNS Selena Engine.
    MCP_SELENA_HOST: "selena-prod-fe-service.
{engine_namespace}.svc.cluster.local"
    # Internal JWKS endpoint для проверки JWT, которые приходят в MCP.
    MCP_KEYCLOAK_JWKS_URL: "
{keycloak_internal_url}/realms/selena/protocol/openid-connect/certs"
    # Public issuer Keycloak realm, ожидаемый MCP.
    MCP_KEYCLOAK_ISSUER: "{keycloak_public_url}/realms/selena"
  # Secret env vars для MCP.
  secrets:
    # Имя Kubernetes Secret с impersonator password.
    existingSecret: selena-ai-secrets
    # Keys из Secret, которые chart пробросит как env vars.
    envKeys:
      # Password `ai_impersonator_all`, который СМ заполнит после Engine
bootstrap.
      - MCP_SELENA_IMPERSONATOR_PASSWORD
  # --- prod tuning: resources mcp ---
  resources:
    requests:
      cpu: "250m"
      memory: "512Mi"
    limits:
      cpu: "1"
      memory: "1Gi"

# --- prod tuning: общее размещение всех подов AI ---
# Toleration безвреден, даже если taint не задан.
#tolerations:
# - key: "{node_taint_key}"
#   operator: "Equal"
#   value: "{node_taint_value}"
#   effect: "NoSchedule"
```

```
# Раскомментируйте после замены плейсхолдеров (иначе поды останутся Pending).  
# nodeSelector:  
#   "{node_label_key}": "{node_label_value}"  
# affinity: {}  
# topologySpreadConstraints: []
```

12.15 Selenacm — values

Файл values для чарта `selenacm`.

selena-cm-values.yaml

```
# Образ Selena Cluster Manager.
image:
  # Repository CM image; меняйте, если используете внутренний
  registry/mirror.
  repository: "hub.selena.sh/selena/selena-cm"
  # Точный release tag CM image, который вы устанавливаете.
  tag: "5.0.85"

# Secret для доступа к private registry; удалите блок, если registry
# публичный.
imagePullSecrets:
  # Имя docker-registry Secret в namespace Selena applications.
  - name: selena-registry

# Только параметры, которые зависят от URL-ов вашего стенда.
config:
  # Env vars, которые попадут в CM ConfigMap.
  env:
    # Internal Keycloak token endpoint для user password login.
    CM_KEYCLOAK_TOKEN_URI: "
{keycloak_internal_url}/realms/selena/protocol/openid-connect/token"
    # Public Keycloak logout endpoint для browser logout.
    CM_KEYCLOAK_LOGOUT_URI: "
{keycloak_public_url}/realms/selena/protocol/openid-connect/logout"
    # Internal JWKS endpoint для проверки JWT signatures.
    CM_KEYCLOAK_JWKS_URI: "
{keycloak_internal_url}/realms/selena/protocol/openid-connect/certs"
    # Public issuer; должен точно совпадать с claim `iss` в Keycloak token.
    CM_KEYCLOAK_ISSUER_URI: "{keycloak_public_url}/realms/selena"
    # Internal token endpoint для service account `selena-cluster-manager-
    api`.
    CM_KEYCLOAK_ADMIN_TOKEN_URI: "
{keycloak_internal_url}/realms/selena/protocol/openid-connect/token"
    # Internal Keycloak Admin API base URL, откуда CM читает users/groups.
    CM_KEYCLOAK_ADMIN_BASE_URI: "
{keycloak_internal_url}/admin/realms/selena"
    # Public IDE endpoint
    CM_UI_IDE_URL: "{ide_public_url}"
    # Public Grafana endpoint
    CM_UI_GRAFANA_URL: "{grafana_public_url}"

# Business mappings: здесь администратор описывает свои Keycloak groups.
extraEnv:
```

SPRING_APPLICATION_JSON нужен для тар-ключей с `/` в Keycloak group paths.

```
- name: SPRING_APPLICATION_JSON
  value: |
    {
      "cm": {
        "identity": {
          "external-groups": {
            "mappings": {
              "[/local/selena-readers]": "local_selena_readers",
              "[/local/selena-writers]": "local_selena_writers"
            }
          }
        },
        "security": {
          "group-role-mappings": {
            "[/cm/admins]": "cm_admin",
            "[/cm/role-admins]": "cm_role_admin",
            "[/cm/viewers]": "cm_viewer"
          }
        }
      }
    }
```

Secret env vars для CM.

secrets:

Имя Kubernetes Secret с DB password, Keycloak client secrets и encryption key.

existingSecret: selena-cm-secrets

Keys из Secret, которые chart пробросит как env vars.

envKeys:

JDBC URL CM PostgreSQL.

- CM_DB_URL

Username CM PostgreSQL.

- CM_DB_USER

Password CM PostgreSQL.

- CM_DB_PASSWORD

Secret Keycloak client `selena-engine`.

- CM_KEYCLOAK_LOGIN_CLIENT_SECRET

Secret Keycloak client `selena-cluster-manager-api`.

- CM_KEYCLOAK_ADMIN_CLIENT_SECRET

Base64 encryption key для CM secrets в DB.

- CM_ENCRYPTION_KEY_BASE64

Идентификатор активного encryption key.

- CM_ENCRYPTION_KEY_ID

```
# --- prod tuning (resources/scheduling) ---
# CM – JVM/Spring. memory limit должен быть заметно больше heap (-Xmx),
# иначе OOMKill.
# В chart по умолчанию replicaCount: 2 (НА через PostgreSQL snapshot +
# ShedLock).
resources:
  requests:
    cpu: "2"
    memory: "2Gi"
  limits:
    cpu: "4"
    memory: "4Gi"
# Toleration безвреден, даже если taint не задан.
#tolerations:
# - key: "{node_taint_key}"
#   operator: "Equal"
#   value: "{node_taint_value}"
#   effect: "NoSchedule"
# Раскомментируйте после замены плейсхолдеров (иначе под останется
# Pending).
# nodeSelector:
#   "{node_label_key}": "{node_label_value}"
# Для разнесения 2 реплик по узлам (лучше – topologySpreadConstraints):
# affinity: {}
# topologySpreadConstraints: []
```

12.16 Selena-ide — values

Файл values для чарта `selena-ide`.

selena-ide-values.yaml

```
# Образ Selena IDE.
image:
  # Repository IDE image; меняйте, если используете внутренний
  registry/mirror.
  repository: "hub.selena.sh/selena/selena-ide"
  # Точный release tag IDE image, который вы устанавливаете.
  tag: "5.0.23"

# Secret для доступа к private registry; удалите блок, если registry
# публичный.
imagePullSecrets:
  # Имя docker-registry Secret в namespace Selena applications.
  - name: selena-registry

# Только параметры, которые зависят от URL-ов вашего стенда и имени Engine
# cluster.
config:
  # Env vars, которые попадут в IDE ConfigMap.
  env:
    # Public URL IDE; должен совпадать с Keycloak redirect URI.
    PUBLIC_URL: "{ide_public_url}"
    # Public issuer Keycloak realm, ожидаемый IDE.
    IDE_KEYCLOAK_ISSUER_URL: "{keycloak_public_url}/realms/selena"
    # Browser authorization endpoint Keycloak.
    IDE_KEYCLOAK_AUTHORIZATION_URL: "
{keycloak_public_url}/realms/selena/protocol/openid-connect/auth"
    # Internal token endpoint для IDE OAuth exchange.
    IDE_KEYCLOAK_TOKEN_URL: "
{keycloak_internal_url}/realms/selena/protocol/openid-connect/token"
    # Internal JWKS endpoint для проверки JWT.
    IDE_KEYCLOAK_JWKS_URL: "
{keycloak_internal_url}/realms/selena/protocol/openid-connect/certs"
    # Public Keycloak logout endpoint.
    IDE_KEYCLOAK_LOGOUT_URL: "
{keycloak_public_url}/realms/selena/protocol/openid-connect/logout"
    # Token endpoint для service client `selena-ide-service`.
    IDE_SERVICE_TOKEN_URL: "
{keycloak_internal_url}/realms/selena/protocol/openid-connect/token"
    # Internal FE service DNS для SQL execution.
    IDE_SELENA_HOST: "selena-dev-fe-service.
{engine_namespace}.svc.cluster.local"
    # Browser-reachable AI iframe URL, который IDE открывает пользователю.
    AI_AGENT_URL: "{ai_public_url}"
```

```
extraEnv:
  - name: OAUTHLIB_INSECURE_TRANSPORT
    value: "1"
  - name: WS_CORS_ALLOWED_ORIGINS
    value: "{ide_public_url}"

# Secret env vars для IDE.
secrets:
  # Имя Kubernetes Secret с IDE secrets и impersonator password.
  existingSecret: selena-ide-secrets
  # Keys из Secret, которые chart пробросит как env vars.
  envKeys:
    # Flask/session signing key.
    - FLASK_SECRET_KEY
    # PostgreSQL connection string IDE.
    - DATABASE_CONN
    # Secret Keycloak browser client `selena-ide`.
    - IDE_KEYCLOAK_CLIENT_SECRET
    # Stable encryption key для IDE tokens/sessions.
    - IDE_TOKEN_ENCRYPTION_KEY
    # Password `ide_impersonator_all`, который СМ заполнит после Engine
bootstrap.
    - IDE_SELENA_IMPERSONATOR_PASSWORD
    # Secret Keycloak service client `selena-ide-service`.
    - IDE_SERVICE_CLIENT_SECRET
    # Optional key для встроенного IDE assistant.
    - OPENAI_API_KEY

# --- prod tuning (resources/scheduling) ---
# В этом чарте resources задаются ПО ворклоадам (web/worker/scheduler),
# а nodeSelector/tolerations/affinity – общие на верхнем уровне.
web:
  resources:
    requests:
      cpu: "1"
      memory: "1Gi"
    limits:
      cpu: "2"
      memory: "2Gi"
worker:
  resources:
    requests:
      cpu: "1"
      memory: "1Gi"
    limits:
```

```
    cpu: "2"
    memory: "2Gi"
scheduler:
  resources:
    requests:
      cpu: "500m"
      memory: "512Mi"
    limits:
      cpu: "1"
      memory: "1Gi"
# Общее размещение для всех подов IDE. Toleration безвреден, даже если
# taint не задан.
#tolerations:
# - key: "{node_taint_key}"
#   operator: "Equal"
#   value: "{node_taint_value}"
#   effect: "NoSchedule"
# Раскомментируйте после замены плейсхолдеров (иначе поды останутся
# Pending).
# nodeSelector:
#   "{node_label_key}": "{node_label_value}"
# affinity: {}
# topologySpreadConstraints: []
```

12.17 Selena-operator — values

Файл values для чарта `selena-operator`.

selena-operator-values.yaml

```
# Настройки Selena Engine operator.
selenaOperator:
  # Образ operator controller.
  image:
    # Repository operator image; меняйте, если используете внутренний
    registry/mirror.
    repository: "hub.selena.sh/selena/operator"
    # Точный tag operator image, согласованный с поддерживаемой версией
    Engine CRD.
    tag: "latest"
  # Secret для доступа к private registry; удалите блок, если registry
  публичный.
  imagePullSecrets:
    # Имя docker-registry Secret, созданного в namespace operator-a.
    - name: selena-registry
  # --- prod tuning (resources/scheduling) ---
  resources:
    requests:
      cpu: "100m"
      memory: "128Mi"
    limits:
      cpu: "500m"
      memory: "512Mi"
  # Toleration безвреден, даже если taint не задан.
  # tolerations:
  #   - key: "{node_taint_key}"
  #     operator: "Equal"
  #     value: "{node_taint_value}"
  #     effect: "NoSchedule"
  # Раскомментируйте после замены плейсхолдеров (иначе под останется
  Pending).
  # nodeSelector:
  #   "{node_label_key}": "{node_label_value}"
  # affinity: {}
```

12.18 Vault — values

Файл values для чарта `vault`.

vault-values.yaml

```
# Bundled Vault optional. В production чаще используется customer-managed
Vault.

# Стабильное имя Vault Service/StatefulSet для команд из этой документации.
fullnameOverride: selena-vault

# Vault Agent Injector не нужен minimal Selena stack.
injector:
  # false отключает mutating webhook injector.
  enabled: false

# Vault server.
server:
  # Persistent storage Vault.
  dataStorage:
    # StorageClass кластера.
    storageClass: "{storage_class}"
    # Размер PVC Vault.
    size: 20Gi
  # --- prod tuning (resources/scheduling) ---
  resources:
    requests:
      cpu: "250m"
      memory: "256Mi"
    limits:
      cpu: "500m"
      memory: "512Mi"
  # Toleration безвреден, даже если taint не задан.
  # tolerations:
  #   - key: "{node_taint_key}"
  #     operator: "Equal"
  #     value: "{node_taint_value}"
  #     effect: "NoSchedule"
  # Раскомментируйте после замены плейсхолдеров (иначе под останется
  Pending).
  # nodeSelector:
  #   "{node_label_key}": "{node_label_value}"
  # Для prod-НА Vault обычно включают raft + podAntiAffinity; см.
  документацию чарта.
  # affinity: {}

# Vault UI service.
ui:
```

```
# true создает Service для Vault UI/API.  
enabled: true
```