

# Селена Моделирование

## РУКОВОДСТВО ПО УСТАНОВКЕ

### Версия программы 1

Москва

2026

© ООО «Платформа Данных Селена», 2026

## АННОТАЦИЯ

Селена Моделирование представляет собой решение по визуализации и проектированию структурированных данных на основе реляционной модели данных.

Установка и конфигурация программного обеспечения осуществляется только сертифицированными инженерами.

По вопросам установки и обновления программного обеспечения обращаться по адресу: [info@selena-lakehouse.ru](mailto:info@selena-lakehouse.ru).

## ТРЕБОВАНИЯ К ОБОРУДОВАНИЮ И ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Поддерживаемые операционные системы:

- Ubuntu 22.04 LTS

Минимальные требования к вычислительным ресурсам

- Количество ядер процессора: 4 vCPU
- Размер оперативной памяти: 16 GB RAM
- Размер свободного дискового пространства: 100 GB SSD

Поддерживаемое программное обеспечение:

- Docker 29.x + Docker Compose Plugin

Требования к сетевому обеспечению

- Открытый сетевой порт 80 - для HTTP (редирект на HTTPS или standalone)
- Открытый сетевой порт 443 - для HTTPS

## УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

### Описание

Данный документ описывает процедуру установки программного обеспечения Селена Моделирование. Установка возможна, в средах:

- Среде контейнеризации Docker.

### Подготовительные действия

Для установки компонентов системы необходимо выполнить ряд подготовительных действий:

- подготовить серверы для установки платформы хранения данных Селена;
- ознакомиться с аппаратными требованиями и технической архитектурой решения;
- организовать доступ к репозиторию nexus - [hub.selena.sh](https://hub.selena.sh)
- при невозможности организации доступа к [hub.selena.sh](https://hub.selena.sh), скачать и выложить дистрибутив на серверы для установки. За предоставлением дистрибутива, просьба обращаться по адресу: [info@selena-lakehouse.ru](mailto:info@selena-lakehouse.ru);
- скачать лицензию. За предоставлением лицензии, просьба обращаться по адресу: [info@selena-lakehouse.ru](mailto:info@selena-lakehouse.ru).

Основными устанавливаемыми компонентами системы являются:

- Фронтенд компонент Селена Моделирование;
- Бэкенд компонент Селена Моделирование.

# ИНСТРУКЦИЯ ПО УСТАНОВКЕ СЕЛЕНА МОДЕЛИРОВАНИЕ В КОНТЕЙНЕРЕ

## 1. Предварительные настройки

### 1.1 Установка Docker

Обновление системы и установка зависимостей

```
sudo apt-get update
```

```
sudo apt-get install -y ca-certificates curl gnupg
```

Добавление Docker GPG ключа

```
sudo install -m 0755 -d /etc/apt/keyrings
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

```
sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

Добавление репозитория Docker

```
echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu jammy stable" | \
```

```
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Установка Docker

```
sudo apt-get update
```

```
sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

Добавление текущего пользователя в группу docker

```
sudo usermod -aG docker $USER
```

```
newgrp docker
```

## 2. Режимы авторизации

Режим	ТИП (CONFIG_DEPLOY- TYPE)	Описание
Test (No-Auth)	test	Авторизация автоматическая, Casdoor не нужен
Casdoor		OAuth2 через Casdoor (требуется отдельный контейнер)
DG Universe	dg	Встраивание в iframe DG Universe с SSO

### 3. Установка в режиме без авторизации (No-Auth / Test Mode)

В этом режиме авторизация происходит автоматически — при открытии приложения пользователь сразу получает JWT токен без ввода логина/пароля.

#### 3.1 Создание директории

```
mkdir -p ~/datamodeller/volumes/postgresql-data
```

```
cd ~/datamodeller
```

#### 3.2 Авторизация в Container Registry

```
sudo docker login hub.selena.sh -u YOUR_USERNAME
```

#### 3.3 Создание docker-compose.yaml

```
cat > docker-compose.yaml << 'EOF'
```

```
services:
```

```
  postgres:
```

```
    image: hub.selena.sh/tools/postgres:17
```

```
    restart: unless-stopped
```

```
    environment:
```

```
      - POSTGRES_PASSWORD=postgres
```

```
    volumes:
```

```
      - ./volumes/postgresql-data:/var/lib/postgresql/data
```

```
    healthcheck:
```

```
      test: ["CMD-SHELL", "pg_isready -U postgres"]
```

```
      interval: 5s
```

```
      timeout: 5s
```

```
      retries: 5
```

**dm-backend:****image: hub.selena.sh/selena/datamodeller-backend:1.0.0****restart: unless-stopped****container\_name: dm-backend****environment:****- SERVER\_PORT=8080****- JWT\_SECRET=your-super-secret-jwt-key-change-me****-****SPRING\_DATASOURCE\_URL=jdbc:postgresql://postgres:5432/postgres?currentSchema=dm****- SPRING\_DATASOURCE\_USERNAME=postgres****- SPRING\_DATASOURCE\_PASSWORD=postgres****- CONFIG\_DEPLOY-TYPE=test****- TEST-USER\_LOGIN=testuser****- TEST-USER\_FULL-NAME=Test User****ports:****- "127.0.0.1:8082:8080"****depends\_on:****postgres:****condition: service\_healthy****dm-frontend:****image: hub.selena.sh/selena/datamodeller-frontend:1.0.0****restart: unless-stopped****container\_name: dm-frontend****ports:**

- "127.0.0.1:4202:8080"

*environment:*

- *API\_URL=http://dm-backend:8080*

*depends\_on:*

- *dm-backend*

*EOF*

### 3.4 Запуск контейнеров

*sudo docker compose pull*

*sudo docker compose up -d*

### 3.5 Создание тестового пользователя

После первого запуска необходимо создать тестового пользователя в БД:

*sudo docker exec datamodeller-postgres-1 psql -U postgres -d postgres -c \*

*"INSERT INTO dm.users (casdoor\_id, username, name, creation\_time) VALUES ('test-user-id', 'testuser', 'Test User', NOW()) ON CONFLICT (username) DO NOTHING;"*

### 3.6 Загрузка лицензии

Перед началом работы с системой необходимо загрузить лицензионный файл:

1. Откройте приложение в браузере
2. Перейдите в раздел \*License (иконка в боковом меню)
3. Загрузите файл лицензии (.bin)
4. После успешной загрузки система будет готова к работе

Примечание: Без валидной лицензии функциональность системы ограничена.

### 3.7 Проверка работоспособности

Проверка статуса контейнеров

```
sudo docker compose ps
```

Проверка логов backend

```
sudo docker logs dm-backend --tail 50
```

Проверка API (должен вернуть JSON с deployType: TEST)

```
curl -s http://127.0.0.1:4202/api/config | jq .deployType
```

## 4. Настройка HTTPS (Self-Signed Certificate)

Для работы через HTTPS на IP-адресе используется self-signed сертификат. Браузер покажет предупреждение — это нормально.

### 4.1 Установка Nginx

```
sudo apt-get update
```

```
sudo apt-get install -y nginx openssl
```

### 4.2 Генерация сертификата для IP

Замените `YOUR\_IP` на реальный IP-адрес сервера:

```
export SERVER_IP="YOUR_IP"
```

```
sudo mkdir -p /etc/ssl/selfsigned
```

```
sudo tee /etc/ssl/selfsigned/ip.conf >/dev/null <<EOF
```

```
[req]
```

```
default_bits = 2048
```

```
prompt = no
```

```
default_md = sha256
```

```
req_extensions = req_ext
```

```
distinguished_name = dn
```

```
[dn]
```

```
CN = ${SERVER_IP}
```

```
[req_ext]
```

```
subjectAltName = @alt_names
```

```
[alt_names]
```

```
IP.1 = ${SERVER_IP}
```

```
EOF
```

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
```

```
-keyout /etc/ssl/selfsigned/ip.key \
```

```
-out /etc/ssl/selfsigned/ip.crt \
```

```
-config /etc/ssl/selfsigned/ip.conf
```

```
sudo chmod 600 /etc/ssl/selfsigned/ip.key
```

### 4.3 Конфигурация Nginx

```
# WebSocket support
```

```
sudo tee /etc/nginx/conf.d/upgrade_map.conf >/dev/null <<'EOF'
```

```
map $http_upgrade $connection_upgrade {
```

```
    default upgrade;
```

```
    ""    close;
```

```
}
```

```
EOF
```

```
# Основной конфиг
```

```
sudo tee /etc/nginx/sites-available/dm.conf >/dev/null <<'EOF'
```

```
# Redirect HTTP -> HTTPS
```

```
server {
```

```
listen 80 default_server;

listen [::]:80 default_server;

server_name _;

return 301 https://$host$request_uri;
}

# Frontend @ 443

server {

    listen 443 ssl http2 default_server;
    listen [::]:443 ssl http2 default_server;

    server_name _;

    ssl_certificate /etc/ssl/selfsigned/ip.crt;
    ssl_certificate_key /etc/ssl/selfsigned/ip.key;

    client_max_body_size 64m;

    location / {

        proxy_pass http://127.0.0.1:4202;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto https;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection $connection_upgrade;
```

```
}  
}  
EOF
```

Активация конфига

```
sudo ln -sf /etc/nginx/sites-available/dm.conf /etc/nginx/sites-enabled/dm.conf  
sudo unlink /etc/nginx/sites-enabled/default 2>/dev/null || true
```

Проверка и перезагрузка

```
sudo nginx -t && sudo systemctl reload nginx
```

#### 4.4 Проверка HTTPS

```
curl -kI https://127.0.0.1/  
-ks https://127.0.0.1/api/config | jq .deployType
```

## 5. Переход на реальный домен

При появлении реального домена (например, `dm.example.com`) нужно:

1. Добавить DNS A-запись `dm.example.com` → YOUR\_IP`
2. Обновить Nginx конфиг— заменить `server\_name \_` на `server\_name dm.example.com`
3. Получить сертификат Let's Encrypt

```
sudo apt-get install -y certbot python3-certbot-nginx
```

```
sudo certbot --nginx -d dm.example.com
```

4. Включить HSTS (опционально) — добавить в server блок:

```
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;
```

5. Проверить автопродление

```
sudo certbot renew --dry-run
```

## 6. Полезные команды

Просмотр логов

```
sudo docker compose logs -f
```

```
sudo docker logs dm-backend -f
```

```
sudo docker logs dm-frontend -f
```

Перезапуск сервисов

```
sudo docker compose restart
```

Остановка

```
sudo docker compose down
```

Полная переустановка (с удалением данных, не забудьте создать тестового пользователя заново!)

```
sudo docker compose down -v
```

```
sudo rm -rf volumes/postgresql-data
```

```
sudo docker compose up -d
```

Обновление образов

```
sudo docker compose pull
```

```
sudo docker compose up -d
```

## 7. Troubleshooting

Backend не запускается

1. Проверить логи

```
sudo docker logs dm-backend
```

2. Частая проблема: Flyway не может применить миграции

3. Решение: удалить данные PostgreSQL и перезапустить

```
sudo docker compose down -v
```

```
sudo rm -rf volumes/postgresql-data
```

```
sudo docker compose up -d
```

Ошибка "user not found" (Тестовый пользователь не создан.)

Выполните:

```
sudo docker exec datamodeller-postgres-1 psql -U postgres -d postgres -c \
```

```
"INSERT INTO dm.users (casdoor_id, username, name, creation_time) VALUES ('test-user-id', 'testuser', 'Test User', NOW()) ON CONFLICT (username) DO NOTHING;"
```

Ошибка Nginx возвращает 502 Bad Gateway

Frontend контейнер не запущен или недоступен:

```
sudo docker compose ps
```

```
sudo docker logs dm-frontend
```