1) Сделать docker login

`docker login ghcr.io -u mrsergeifrolov -p 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`

2) Создать docker-compose.yml file. Замените `{подставьте_ваш_ip}` на ваш ip адрес (в переменных CONFIG_CASDOOR_FRONTEND_URL и CONFIG_CASDOOR_REDIRECT_URL).

```yml
services:
  dm-backend:
    image: ghcr.io/selena-lakehouse/dm-backend:dev-amd64
    restart: unless-stopped
    container_name: dm-backend
    ports:
      - "8080:8080"
    environment:
      - SERVER_PORT=8080
      - JWT_SECRET=secretdevasdasdasd
      - SPRING_DATASOURCE_URL=jdbc:postgresql://postgres:5432/postgres?currentSchema=dm
      - SPRING_DATASOURCE_USERNAME=postgres
      - SPRING_DATASOURCE_PASSWORD=qeqwbfnmbnewmrb
      - CONFIG_CASDOOR_FRONTEND_URL=http://{подставьте_ваш_ip}:8000
      - CONFIG_CASDOOR_REDIRECT_URL=http://{подставьте_ваш_ip}:4202/oauth2callback
      - CASDOOR_CLIENT_ID=datamodeller-app
      - CASDOOR_CLIENT_SECRET=d7b4c3iio62900c0d56bcae4005b1efa1861d7d5
      - CASDOOR_ENDPOINT=http://casdoor:8000

  dm-frontend:
    image: ghcr.io/selena-lakehouse/dm-frontend:dev-amd64
    restart: unless-stopped
    container_name: dm-frontend
    ports:
      - "4202:8080"
```

```
    environment:
      - API_URL=http://dm-backend:8080

  postgres:
    image: "postgres:15"
    volumes:
      - ./volumes/postgresql-data:/var/lib/postgresql/data
    environment:
      - POSTGRES_PASSWORD=qeqwbfnmbnewmrb
    ports:
      - "5432:5432"
    deploy:
      resources:
        limits:
          memory: 150M

  casdoor:
    restart: unless-stopped
    image: ghcr.io/selena-lakehouse/auth:1.3.20-amd64
    environment:
      - driverName=postgres
      - dataSourceName=user=postgres password=qeqwbfnmbnewmrb host=postgres port=5432
sslmode=disable dbname=casdoor
      - dbName=casdoor
      - initDataFile=/home/casdoor/init_data.json
    entrypoint: /bin/sh start.sh
    volumes:
      - ./volumes/casdoor:/home/casdoor
      - ./start_casdoor.sh:/start.sh
    ports:
      - "8000:8000"
    depends_on:
      - postgres
```

```

3) Рядом с docker-compose.yml положить файл start_casdoor.sh. Замените `{подставьте_ваш_ip}` на ваш ip адрес.

```sh
#!/bin/sh

default_user=${DEFAULT_USER:-"web_console_superadmin"}
default_user_password=${DEFAULT_USER_PASSWORD:-"1234"}

datamodeller_app_client_secret=${DATAMODELLER_APP_CLIENT_SECRET:-"d7b4c3iio62900c0d56bcae4005b1efa1861d7d5"}
datamodeller_app_url=${DATAMODELLER_APP_URL:-"http://{подставьте_ваш_ip}:4202"}
datamodeller_app_certificate=${DATAMODELLER_APP_CERTIFICATE:-"-----BEGIN CERTIFICATE-----\nMIIE2TCCAsGgAwIBAgIDAeJAMA0GCSqGSIb3DQEBCwUAMCYxDjAMBgNVBAoTBWFk\nbWluMRQwEgYDVQQDDAtjZXJ0XzZsN3FzbjAeFw0yNDEwMDIxMTUxMzBaFw00NDEw\nMDIxMTUxMzBaMCYxDjAMBgNVBAoTBWFkbWluMRQwEgYDVQQDDAtjZXJ0XzZsN3Fz\nbjCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBALut7HeQN3GfQVmqfaUt\nY7+au4FjoDd3Cs7D49WUx+IWavpgrAN2YFoSeSj4zr8WmIyvW0qmCDJVYqt6G2t6\nXcx7yM59wTZD8sFI1WNrZMHFZTaw9N2FcyFrVJxiqZgWwLACCwZOgdA/rHUY9AYl\nmMsY9sUwQrj5vj4VkNMxYLMZ/bffenZ/he9NVG7oD3oLeSPGrZIq5BXrkPdn8Bon\nHuYf1Ws97NFvQAR9WDpmYbRK/orjgsHKK1QYpesgKv6hTFgnxyPvUPrPZnzUMQFf\nxlY+2qlBJe6nLXbxhNhCl9eHozKhsvcx9N2Msd8Yp3hXyinOr8ogQ7/wTNT4J1v0\nYvip6sTfG5uRwVxk4YGoiVaAuBt9/7myrwTDxXk/4x5dgVa2XdJbB5P6XoWVug2Q\ndNPZi2AuikR2bR8GD8fs1XYmtyAEHqdhMDZSDyD6wthRe/ckuKNgSrDVH+ivoR5N\nGPEN2/L6LVOWW9E/YNs/AdUCf4tzGBnSMH9cB5h+FAmtl07yJMwMEvONkNrdl8gN\n0OUuM/boC7/QFLfCBigzXWabUpcBvJCsxkUrehUMI6V6waGJw+gJ8GpR1uOw1k+S\nKX9mUHkpq3jA6ZSdtnLe4HF8PsNooTFaMtmEGYQ8+jBWv5OpG8Ms7p8GRQE1q9Og\nZySGxnAhVSAvhoNZInxWQml9AgMBAAGjEDAOMAwGA1UdEwEB/wQCMAAwDQYJKoZI\nhvcNAQELBQADggIBAEm4Ofv9RaqsSlbv011hLJQc7Zevdj3j5MXJLKEHfxEfYvn0\nROicU4C1LXPLQRePDPE4qoqTS3E3SA1tvRI/JR4x1A+rpnIifZgintgVnudsWpFX\n2Il3M66g40525w02eWxRfnBDWQU6Oa86mWd8sF15TZiXH+Ik1jKmyOsYoLfc9PF9\no5wFPhAxoVNpajenOIOgqGZUW6Vxf7A5LqeNVsm4FfvUwNz/bQoime5Ks4G8SPWW\n5VJtxy36f3mMx8V9Vyr3OlkafXhshqchM8slUg6mkI7LyaAulZuzkbelDUZtSZFv\nvganWfPlnIfoCLAZxr+dB3efuD5JfBr/noEGWbYyjOX+nCojrXVZtLUSE6eCq9Df\nEDLjLGGNtRtpPa2Ntfvl75uRw471Y7qUaKYyb7qoAQkz+GfH/dyrYKb7zpPgXiBd\nnNYYxrdWoaM7orK1gZyVK/Zabmw1jvfEbcV3is0+AA6GRKe84vssDpaNX2KdBBcUU\n2U40h+5ooU/E3+MINiFfxD2Qo1TwZR5N2Iwjyl2dZxi6vg5Y8rgrUAYdhWFWB4jH\nEy9f8dMuhEAv9gtmO4aBki6Q4hWnsl7N8AOeH+MMRhrGlNN0UgXq3KUhH8ehDe2N\nt7f7A5gq3MVf/mq6hpbAsdKdl7gWd/tjdBriHp10AmBiAQVbAtp6MlndE/tO\n-----END CERTIFICATE-----\n"}
datamodeller_app_private_key=${DATAMODELLER_APP_PRIVATE_KEY:-"-----BEGIN RSA PRIVATE KEY-----\nMIIJKAIBAAKCAgEAu63sd5A3cZ9BWap9pS1jv5q7gWOgN3cKzsPj1ZTH4hZq+mCs\nA3ZgWhJ5KPjOvxaYjK9bSqYIMlViq3oba3pdzHvIzn3BNkPywUjVY2tkwcVlNrD0\n3YVzIWtUnGKpmBbAsAILBk6B0D+sdRj0BiWYyxj2xTBCuPm+PhWQ0zFgsxn9t996\ndn+F701UbugPegt5I8atkirkFeuQ92fwGice5h/Vaz3s0W9ABH1YOmZhtEr+iuOC\nwcorVBil6yAq/qFMWCfHI+9Q+s9mfNQxAV/GVj7aqUEl7qctdvGE2EKX14ejMqGy\n9zH03Yyx3xineFfKKc6vyiBDv/BM1PgnW/Ri+KnqxN8bm5HBXGThgaiJVoC4G33/\nubKvBMPFeT/jHl2BVrZd0IsHk/pehZW6DZB009mLYC6KRHZtHwYPx+zVdia3IAQe\np2EwNlIPIPrC2FF79yS4o2BKsNUf6K+hHk0Y8Q3b8votU5
```
```

Zb0T9g2z8B1QJ/i3MY\nGdIwf1wHmH4UCa2XTvIkzAwS842Q2t2XyA3Q5S4z9ugLv9AUt8IGKDNdZptSlwG8
\nkKzGRSt6FQwjpXrBoYnD6AnwalHW47DWT5Ipf2ZQeSmreMDplJ22ct7gcXw+w2ih\nMVoy2YQZhDz6MF
a/k6kbwyzunwZFATWr06BnJIbGcCFVIC+Gg1kifFZCaX0CAwEA\nAQKCAgEAhjd3aHoGbIdLoOhAOqQfNaNe
fMXWilPjssJ8aP2hhJornysLWN6LQV78\nrsRgdl+ecN3nF5dtPTG7Ao6slUxJUDcmACYGytwE+x0vHY5r2/diz
p32P4V0NG4h\ncEVOq6l522VapQP36gnfN07FFLBGeNg9gf9leAuCdbvOenpRgUQRIjH7zkqcaeqP\ne1ER4
GHcHC21a1apDtQMNCJsnwdjHYeehLcgaRZ6XbuWIQ2eyeTn7EOP+y6tldKa\ntW+nz7lic5je3WAENZmk3Zi
SAz2PBZaloW0px2ZfKTTHoDDfjg8+MrNY3VcomlhH\nzBhBi9qchRF+vbodOWKMT7G31BcNvzRqHQkh4gC
4xLx0TVlInEk8VoMOmp42jfdw\nxH7SqwxEh+S2fJtK+BmBboyANwro4T3iWLzCiWKGj/q2nGbN28Pi4G0oO
mP1QzF6\n3rJ9JYgNUcYr0h0a4vHFkSdf6nNfYNzsodLXa6g5eAK2U2NaKCngZE425wMwAteS\nZEeOBhxUb
qDVEV9oSyfNhqirhF/WRy1pICm98qupxtYhI/tBQs21UAN0nI4RySy7\nO4kD1SR+luJj8cRxDQJ+nvADR0EX
Wr7sH0fvpJhxapB1VSXqe5p0loRRfjVdWn74\n8dWxt6NHv6OB4Pd6H2LH3V8D9p+aDMagVfiQyuVmRkfE3
y711LECggEBAO9g5MDQ\n9EBUrlTXMaPSoTXzVmudQzRwxQHAGGBodlfEcHqRi/GZsQoPQM67r2nCnGQ
Km/wu\nQi7/ELPEd211HIhJ/xmnQeSOh4KJIeiOV8k1aPK8R3j3osurloCw7uCoPnvyQ/MO\nASAeOnEqilKP
u8vpZUKCcGUrNju05FZtF1zlILMrH7VjCDcNlVme3L7OgfvFEDCX\n6g9x2ldjCiAgCrrZU3Y/zs/VJOvBNZb+Ei6
hN6JdXgE0kLh5w9FSWitcr8oOgTGv\nJobj2pEBO16Xphi4pRgj7/CDv8Er6FAzfg+ZlYXmB7ikOtXDWGZOmy
DwYxA9Te49\nQ5O5xe9g7rm/LC8CggEBAMi2C598/1qKx4J0qWRxjH2bXjvY34PO4sJmO8xSjc9W\nUb5lVs
926cvsYbG934sXQ/woe9bLwLShJWiQzD2DKabS1QOAVcE88ZChw9UHpZS8\nTzVirKalqgKts4kZ/lVsdSbj5p
HgQOAto2lCRxUkJTU6l3Xkh08MuZVQyChWtZth\n6iMTh6b4v55NtQplFfuC6AQtllHpGBkOFYvNsFlutCEDi
MHMrewoV2y6YQS3pK/j\nhmk1now/N7muCzvQTTWEZ9bGP/WhwYBGxoPTtWsuPXodQv5F4Juh3HPW3
DuWJDwY\nq8W7uiLM4Jy/nwcofbd1SjAW+K6s9aKWmc+CdcwdfhMCggEAG1fYCWZU4BxMJYjH\nNM1U
eNMo5fX6vTy8ZCzLHAgSrIfVyUpiz+AcZufDIVoy+UOvlYXYt3+ZM8ES6VXh\nxnEAJ1k8eHOwhSj6taBwbR4F
TQh6ym9TQrAWumkJMf4D60mMiZ/svxC0dDS/vDl9\nv1yHuQ0Z8fTwXb4H3d4VzwF7D7NkftLINnHveEpm
6uvIj0sLcfTampJfvA/nPuAj\nQfQnBZzZecJkHz1bkEprkqnG7eigcD12BXE5rrjJ81kSz60wr0Xk4AX/wpy4VKPt
\nYiq0YFKHGTDRsYl7F9eqG68mESYb/EWAUEkCJ3S+3kDQW3zZ8/rsse/xjihEQcmy\nuyjpJQKCAQAhfsf3y
Mtwl8Jt0fxG5/AzMg+K20mJ3KfZG2iVU52Y4WXo6U6Qdt7T\nilJPcElzE+Hjnfn0wbIaMt5kUe3TwiTcU0j7OZ
qWEg6aPXTKA2mtpbNuTFZVxUXX\nBpf0pGLebKgsYlIjPBJ3lwYzVb05GPwgovON2CKAnbYG5iFRIg9SrAd/7
Q1xXwfT\nalJxNqdxfsorkFQKvWSpIcGUwDHS6veins1e0kKgK6dTXNM+D0urksPmoO0hqJcr\nJJF5XOzVpp
ARtqoL2v/l4YbVjyoyhdf6nhGFD7JJJrz8k/nS895+LPvVltucXSaK\n4iuu7SH10NwdpGfB9tlTVeC7UaFI/A8DA
oIBAB3U+NuksI6S5s7pfIwgziGIKeGD\nM/28o6dHJB42q317D7cxBU/eKX+L1Q83iiGSt1fFKNl/D1MpVWW4
pyesTDhVZCB2\naAlJ9h/qpRP06jsxItZDx3MXUYYG25l4xeIwSuhohKei7jB7JE4Y3unsNnxSGwrX\nsbiqGw3j
gzdOvZEa1upV4vUUOK45uMmfpBzHNLk1AfoTKPwpCIGde5KsQ/P8pATk\nBtE43nzgUKSdJi1tucjRUULC3z
q2QfhQ3mnOHvI5u8dbxxEQp/Wvb04DqL0qN42k\nojP+QtNEPuvHJTyL9NkaHdxmN5NIU2up4BYM8j9fsR
5TwdE9kcsE4Z3U5ak=\n-----END RSA PRIVATE KEY-----\n"}

```
json=$(cat <<EOF
{
  "organizations": [],
  "applications": [
    {
      "owner": "admin",
      "name": "datamodeller-app",
      "displayName": "Datamodeller app",
      "homepageUrl": "$datamodeller_app_url",
      "organization": "selena",
```

```
  "cert": "Selena datamodeller certificate",
  "enablePassword": true,
  "enableSignUp": false,
  "enableSigninSession": true,
  "enableAutoSignin": true,
  "clientId": "datamodeller-app",
  "clientSecret": "$datamodeller_app_client_secret",
  "providers": [],
  "signinMethods": [
   {
    "name": "Password",
    "displayName": "Password",
    "rule": "All"
   }
  ],
  "signinItems": [],
  "orgChoiceMode": "None",
  "signupItems": [],
  "grantTypes": [
   "authorization_code",
   "token",
   "id_token"
  ],
  "redirectUris": [
   "$datamodeller_app_url/oauth2callback"
  ],
  "tokenFormat": "JWT",
  "tokenFields": [],
  "expireInHours": 168,
  "failedSigninLimit": 5,
  "failedSigninFrozenTime": 15
}
```

```
],
"users": [
 {
  "owner": "selena",
  "name": "$default_user",
  "type": "normal-user",
  "password": "$default_user_password",
  "displayName": "Cluster manager admin",
  "ranking": 1,
  "isAdmin": true,
  "signupApplication": "datamodeller-app"
 }
],
"providers": [],
"certs": [
 {
  "owner": "selena",
  "name": "Selena datamodeller certificate",
  "displayName": "Selena datamodeller certificate",
  "scope": "JWT",
  "type": "x509",
  "cryptoAlgorithm": "RS256",
  "bitSize": 4096,
  "expireInYears": 20,
  "certificate": "$datamodeller_app_certificate",
  "privateKey": "$datamodeller_app_private_key"
 }
],
"ldaps": [],
"models": [],
"permissions": [],
"payments": [],
```

```
  "products": [],
  "resources": [],
  "roles": [
    {
      "displayName": "Datamodeller admin",
      "isEnabled": true,
      "name": "DATAMODELLER_ADMIN",
      "owner": "selena",
      "roles": [],
      "users": []
    },
    {
      "displayName": "Datamodeller user",
      "isEnabled": true,
      "name": "DATAMODELLER_USER",
      "owner": "selena",
      "roles": [],
      "users": []
    }
  ],
  "syncers": [],
  "tokens": [],
  "webhooks": [],
  "groups": [],
  "adapters": [],
  "enforcers": [],
  "plans": [],
  "pricings": []
}
EOF
)
```

printf "%s" "$json" > /home/casdoor/init_data.json

./server --createDatabase=true

```

3) Запустить: `docker compose up -d postgres`

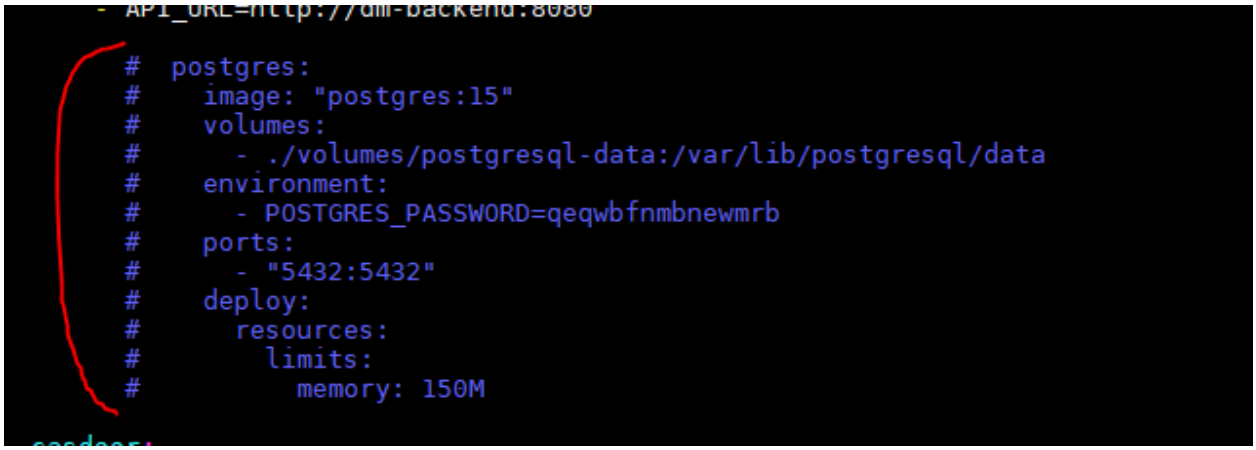
4) Подключиться к postgres и запустить: `CREATE DATABASE casdoor;`

5) Запустить: `docker compose up -d`

6) Перейти на `http://localhost:4202` и войти с помощью `web_console_superadmin` / `1234`

Если у вас отдельный контейнер postgre или установлен локально:

В docker-compose.yml нужно закомментировать контейнер postgres:




Закоментировать зависимость:


Указать ваши данные по подключению к postgre:
```

```yaml
services:
  dm-backend:
    image: ghcr.io/selena-lakehouse/dm-backend:dev-amd64
    restart: unless-stopped
    container_name: dm-backend
    ports:
      - "8070:8080"
    environment:
      - SERVER_PORT=8080
      - JWT_SECRET=secretdevasdasdasd
      - SPRING_DATASOURCE_URL=jdbc:postgresql://unidg04-cons.data-integration.ru:5431/postgres?currentSchema=dm
      - SPRING_DATASOURCE_USERNAME=postgres
      - SPRING_DATASOURCE_PASSWORD=qeqwbfnmbnewmrb
      - CONFIG_CASDOOR_FRONTEND_URL=http://unidg04-cons.data-integration.ru:9000
      - CONFIG_CASDOOR_REDIRECT_URL=http://unidg04-cons.data-integration.ru:4402/oauth2callback
      - CASDOOR_CLIENT_ID=datamodeller-app
      - CASDOOR_CLIENT_SECRET=d7b4c3iio62900c0d56bcae4005b1efa1861d7d5
      - CASDOOR_ENDPOINT=http://casdoor:8000
```

```yaml
  casdoor:
    restart: unless-stopped
    image: ghcr.io/selena-lakehouse/auth:1.3.20-amd64
    environment:
      - driverName=postgres
      - dataSourceName=user=postgres password=qeqwbfnmbnewmrb host=unidg04-cons.data-integration.ru port=5431 sslmode=disable dbname=casdoor
      - dbName=casdoor
      - initDataFile=/home/casdoor/init_data.json
    entrypoint: /bin/sh start.sh
    volumes:
      - ./volumes/casdoor:/home/casdoor
      - ./start_casdoor.sh:/start.sh
    ports:
      - "9000:8000"
      #    depends_on:
      #      - postgres
```

Создать БД casdoor в вашем postgre:

CREATE DATABASE casdoor;

Смотрим какие БД есть :

```
postgres@b8953f09fac1:~$ psql
psql (16.3 (Debian 16.3-1.pgdg120+1))
Type "help" for help.

postgres=# \l
                                                  List of databases
   Name    |  Owner   | Encoding | Locale Provider |  Collate   |   Ctype    | ICU Locale | ICU Rules |   Access privileges
-----------+----------+----------+-----------------+------------+------------+------------+-----------+-----------------------
 discovery | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |            |           | =Tc/postgres          +
           |          |          |                 |            |            |            |           | postgres=CTc/postgres
 postgres  | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |            |           | =Tc/postgres          +
           |          |          |                 |            |            |            |           | postgres=CTc/postgres
 scanner01 | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |            |           |
 template0 | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |            |           | =c/postgres           +
           |          |          |                 |            |            |            |           | postgres=CTc/postgres
 template1 | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |            |           | =c/postgres           +
           |          |          |                 |            |            |            |           | postgres=CTc/postgres
(5 rows)

postgres=#
```

Создаем:

```
postgres=# CREATE DATABASE casdoor;
CREATE DATABASE
postgres=# \l
                                                 List of databases
   Name    |  Owner   | Encoding | Locale Provider |  Collate   |   Ctype    | ICU Locale | ICU Rules |   Access privileges
-----------+----------+----------+-----------------+------------+------------+------------+-----------+-----------------------
 casdoor   | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |            |           |
 discovery | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |            |           | =Tc/postgres         +
           |          |          |                 |            |            |            |           | postgres=CTc/postgres
 postgres  | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |            |           | =Tc/postgres         +
           |          |          |                 |            |            |            |           | postgres=CTc/postgres
 scanner01 | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |            |           |
 template0 | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |            |           | =c/postgres          +
           |          |          |                 |            |            |            |           | postgres=CTc/postgres
 template1 | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |            |           | =c/postgres          +
           |          |          |                 |            |            |            |           | postgres=CTc/postgres
(6 rows)

postgres=#
```

Если ошибка:

Caused by: org.hibernate.MappingException: The increment size of the [projects_id_seq] sequence is set to [50] in the entity mapping while the associated database sequence increment size is [1].

меняем сиквенс в бд:

Select sequence_schema, sequence_name, INCREMENT from information_schema.sequences where sequence_name='projects_id_seq';

```
postgres=#
postgres=#
postgres=#
postgres=# Select sequence_schema, sequence_name, INCREMENT from information_schema.sequences where sequence_name='projects_id_seq';
         sequence_schema         |  sequence_name  | increment
---------------------------------+-----------------+-----------
 com_universe_dg_dis_integration | projects_id_seq | 1
(1 row)

postgres=#
```

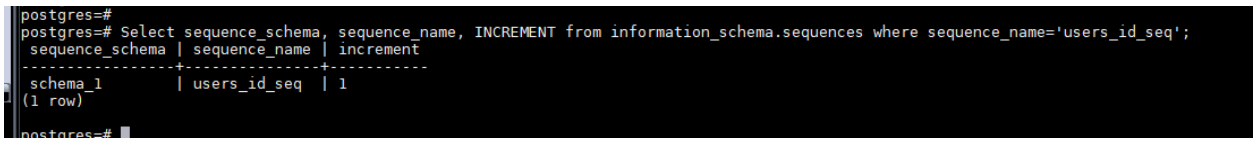
ALTER SEQUENCE projects_id_seq INCREMENT BY 50;

```
postgres=#
postgres=# Select sequence_schema, sequence_name, INCREMENT from information_schema.sequences where sequence_name='projects_id_seq';
         sequence_schema         |  sequence_name  | increment
---------------------------------+-----------------+-----------
 com_universe_dg_dis_integration | projects_id_seq | 1
(1 row)

postgres=# ALTER SEQUENCE projects_id_seq INCREMENT BY 50;
ERROR:  relation "projects_id_seq" does not exist
```

ALTER SEQUENCE com_universe_dg_dis_integration.projects_id_seq INCREMENT BY 50;

```
---------------------------------+-----------------+-----------
 com_universe_dg_dis_integration | projects_id_seq | 1
(1 row)

postgres=# ALTER SEQUENCE projects_id_seq INCREMENT BY 50;
ERROR:  relation "projects_id_seq" does not exist
postgres=# ALTER SEQUENCE com_universe_dg_dis_integration.projects_id_seq INCREMENT BY 50;
ALTER SEQUENCE
postgres=#
```

Select sequence_schema, sequence_name, INCREMENT from information_schema.sequences where sequence_name='users_id_seq';



ALTER SEQUENCE schema_1.users_id_seq INCREMENT BY 50;

users_id_seq

Запустилось

Настройка SSL

1. Нужно добавить контейнер nginx в docker-compose.yml:

Пример:

```
 dm-nginx-dg213:
   image: nginx:latest
   container_name: dm-nginx-dg213
   ports:
    - "4222:443"
    - "9000:8443"
   volumes:
    - ./nginx/nginx.conf:/etc/nginx/nginx.conf
    - ./nginx/ssl/node1.pem:/etc/nginx/ssl/node1.pem
    - ./nginx/ssl/node1-key.pem:/etc/nginx/ssl/node1-key.pem
   restart: always
```

2. Убрать проброс портов наружу с контейнеров моделлера frontend, backend, casdoor.

3. SSL сертификаты:

3.1 если у вас уже есть ssl сертификаты на данном сервере, то вы можете использовать их.

Для этого, например если у вас установлен UDG с SSL на данном сервере вы можете использовать эти же ssl сертификаты для моделлера (в данном случае мы используем уже имеющиеся самоподписные сертификамы UDG node1.pem и ключ node1-key.pem)

Пример нашего docker-compose.yml:

```
services:
  dm-backend-dg213:
    image: ghcr.io/selena-lakehouse/dm-backend:dev-amd64
    restart: unless-stopped
    container_name: dm-backend-dg213
   # ports:
    #  - "8070:8070"
    environment:
      - SERVER_PORT=8070
      - JWT_SECRET=secretdevasdasdasd
      - SPRING_DATASOURCE_URL=jdbc:postgresql://127.0.0.1:5431/postgres?currentSchema=dm
      - SPRING_DATASOURCE_USERNAME=postgres
      - SPRING_DATASOURCE_PASSWORD=password
      - CONFIG_CASDOOR_FRONTEND_URL=https://unidg04-cons.data-integration.ru:9000
      - CONFIG_CASDOOR_REDIRECT_URL=https://unidg04-cons.data-integration.ru:4222/oauth2callback
      - CASDOOR_CLIENT_ID=datamodeller-app
      - CASDOOR_CLIENT_SECRET=d7b4c3iio62900c0d56bcae4005b1efa1861d7d5
      - CASDOOR_ENDPOINT=http://casdoor-dg213:8000

  dm-frontend-dg213:
    image: ghcr.io/selena-lakehouse/dm-frontend:dev-amd64
    restart: unless-stopped
    container_name: dm-frontend-dg213
```

```
    #    ports:
    #      - "4222:8080"
    environment:
      - API_URL=http://dm-backend-dg213:8070

  casdoor-dg213:
    restart: unless-stopped
    image: ghcr.io/selena-lakehouse/auth:1.3.20-amd64
    container_name: casdoor-dg213
    environment:
      - driverName=postgres
      - dataSourceName=user=postgres password=postgres host=127.0.0.1 port=5431 sslmode=disable
dbname=casdoor
      - dbName=casdoor
      - initDataFile=/home/casdoor/init_data.json
    entrypoint: /bin/sh start.sh
    volumes:
      - ./volumes/casdoor:/home/casdoor
      - ./start_casdoor.sh:/start.sh
        #    ports:
    #      - "9000:8000"
        #    depends_on:
        #      - postgres


        #networks:
        #  dm_network:
        #    driver: bridge
  dm-nginx-dg213:
    image: nginx:latest
    container_name: dm-nginx-dg213
    ports:
      - "4222:443"
      - "9000:8443"
```

```
  volumes:
    - ./nginx/nginx.conf:/etc/nginx/nginx.conf
    - ./nginx/ssl/node1.pem:/etc/nginx/ssl/node1.pem
    - ./nginx/ssl/node1-key.pem:/etc/nginx/ssl/node1-key.pem
  restart: always
```

В директории с docker-compose.yml файлом необходимо создать директорию nginx и nginx/ssl

В директории nginx нужно создать файл nginx.conf следующего содержания:

```
user  nginx;
worker_processes  auto;

error_log  /var/log/nginx/error.log notice;
pid        /var/run/nginx.pid;


events {
  worker_connections  1024;
}


http {
  include       /etc/nginx/mime.types;
  default_type  application/octet-stream;

  log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                    '$status $body_bytes_sent "$http_referer" '
                    '"$http_user_agent" "$http_x_forwarded_for"';

  access_log  /var/log/nginx/access.log  main;
```

```
    sendfile        on;
    #tcp_nopush     on;

    keepalive_timeout  65;

    #gzip  on;
map $http_upgrade $connection_upgrade {
    default upgrade;
    '' close;
}

# Frontend @ 4222
server {
    listen 443 ssl http2 default_server;
    listen [::]:443 ssl http2 default_server;
    server_name _;

    ssl_certificate     /etc/nginx/ssl/node1.pem;
    ssl_certificate_key /etc/nginx/ssl/node1-key.pem;

    client_max_body_size 64m;

    location / {
        proxy_pass http://dm-frontend-dg213:8080;
        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;
    }
```

```
}

# Casdoor @ 9000
server {
    listen 8443 ssl http2;
    listen [::]:8443 ssl http2;
    server_name _;

    ssl_certificate     /etc/nginx/ssl/node1.pem;
    ssl_certificate_key /etc/nginx/ssl/node1-key.pem;

    client_max_body_size 32m;

    location / {
        proxy_pass http://casdoor-dg213:8000;
        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;
    }
}

}
```

В секциях # Frontend @ 4222 и # Casdoor @ 9000

нужно указать корректные для вашего развертывания параметры:

proxy_pass http://dm-frontend-dg213:8080;

proxy_pass http://casdoor-dg213:8000;

так же, если вы меняете расположение и\или имена ключей SSL нужно это указать в параметрах:

ssl_certificate /etc/nginx/ssl/node1.pem;

ssl_certificate_key /etc/nginx/ssl/node1-key.pem;

В директорию nginx/ssl нужно положить ваши ключи SSL

В файле start_casdoor.sh, там же, где у вас docker-compose.yml, нужно поменять:

datamodeller_app_url=${DATAMODELLER_APP_URL:-"http://hostname:4222"}

на

datamodeller_app_url=${DATAMODELLER_APP_URL:-"https://hostname:4222"}

hostname- hostname вашего сервера, на котором вы разворачиваете моделлер

4222 – порт, который вы выносите наружу из контейнера nginx по frontend (# Frontend @ 4222)

3.2 Если у вас нет SSL ключей на данном сервере

Генерация сертификата для IP (самоподписанный)

> Let’s Encrypt **не** выдаёт сертификаты для голых IP. Для IP нужен self-signed cert с SAN=IP (браузер покажет предупреждение, это ожидаемо).

```bash
sudo apt update && sudo apt install -y openssl
```

```
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 = <IP>


[req_ext]

subjectAltName = @alt_names


[alt_names]

IP.1 = <IP>

EOF


sudo openssl req -x509 -nodes -days 9999 -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

```


Замените `<IP>` на реальный публичный IP (например, `20.79.153.26`) **в двух местах** (`CN` и `IP.1`).

Теперь в nginx.conf можно указать данные ключи и идти по пунктам инструкции, если у вас уже есть SSL ключи

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