

# Руководство по установке и обновлению

Юниверс MDM HPE 6.11

2024

ООО «Юниверс Дата» оставляет за собой право вносить изменения в настоящий документ без предварительного уведомления.

Данный документ и его отдельные части в любом порядке их расположения не подлежат воспроизведению, публикации и передаче третьим лицам (вне зависимости от конечной цели совершения указанных действий) без письменного разрешения ООО «Юниверс Дата».

Редакция от 18.06.2024.

© ООО «Юниверс Дата», 2015 – 2024. Все права защищены.

# 1. Установка системы

Руководство содержит сведения по установке Юниверс Управление Мастер Данными Высоконагруженная Редакция (далее – Юниверс MDM HPE).

#### Содержание:

| 1.1.  | Репозиторий Docker                        | 3    |
|-------|---|------|
| 1.2.  | Docker через загрузку образов из файла    | 9    |
| 1.3.  | Закрытый контур на Ubuntu                 | 15   |
| 1.4.  | Открытый контур на Ubuntu                 | 26   |
| 1.5.  | Закрытый контур на Astra Linux            | 30   |
| 1.6.  | Открытый контур на Astra Linux            | 47   |
| 1.7.  | Открытый контур на Red OS 8               | 51   |
| 1.8.  | Открытый контур на ALT Linux 10           | 59   |
| 1.9.  | Описание дистрибутива и контрольные суммы | 66   |
| 1.10. | Настройка библиотеки Hunspell             | . 68 |

# 1.1. Репозиторий Docker

# Внимание:

#### Дисклеймер.

Инструкция содержит примерный порядок действий по установке. Инструкция призвана описать логику действий по установке на простом, абстрактном примере. В зависимости от инфраструктуры сервера (кластера) и его конфигурации, вам могут понадобиться дополнительные или иные шаги.

Эта инструкция не должна восприниматься буквально, а использоваться в качестве основы действий.

Рекомендуемый способ установки Юниверс MDM – с использованием Docker.

Также система может устанавливаться вручную. Этот метод установки требует специальных навыков и зависит от множества параметров: деталей проекта, ИТ-инфраструктуры, параметров интеграции и т.д.

#### Примечание:

Эта страница содержит **пример** установки Enterprise редакции через Docker. Для установки необходимо иметь репозиторий с дистрибутивом в виде образов Docker.

## 1.1.1. Подготовка к установке

#### Подготовка сервера:

- Обновите существующий список пакетов Ubuntu. Команда: sudo apt update
- Установите Docker через официальный репозиторий. Документацию см. по ссылке <u>https://docs.docker.com/</u>
- Установите Docker Compose (версия выше 1.29).
- Настройте разрешения для Docker Compose. Пример команды: sudo chmod +x /usr/local/bin/docker-compose
- Установите Git.

## Репозиторий Docker:

Согласно договору поставки Юниверс MDM получите доступ к репозиторию с дистрибутивом вашего продукта:

- Основной путь: получите архив с Docker-образами дистрибутива. Создайте собственный репозиторий через локальный Dockerhub.
- Альтернативный путь: Согласно договору поставки Юниверс Дата получите доступ к репозиторию с дистрибутивом вашего продукта.

## 1.1.2. Установка с Docker

## Чтобы установить Юниверс MDM:

- 1. Убедитесь, что ссылка на репозиторий с дистрибутивом вашего продукта в наличии. О получении ссылки см. выше.
- 2. Клонируйте репозиторий из Gitlab или Docker. Пример команды:

git clone [your-link-universe-platform-deploy.git]

3. Перейдите в каталог с проектом. Пример команды:

cd universe-platform-deploy

4. Авторизуйтесь в репозитории с помощью команды (используя логин / пароль):

```
sudo docker login repo/universe-mdm-deploy
```

5. Для запуска системы используйте команду:

docker-compose up -d

По умолчанию пользовательский интерфейс доступен для просмотра на localhost:8082.

Логин и пароль по умолчанию: admin/ admin. После введения логина и пароля система запросит файл лицензии и смену пароля.

## 1.1.3. Пользовательская установка с Docker Compose

| Примечание:                                     |  |
|---|--|
| Версия docker-compose должна быть 1.29 или выше |  |

При необходимости используйте docker-compose.yaml для создания собственного образа. В файле .env содержится полный перечень доступных переменных. Описание файла <u>см. ниже</u>.

- 1. Создайте файл docker-compose.yaml.
- 2. Создайте папку *hunspell* со словарями. См. также инструкцию по <u>настройке</u> <u>библиотеки Hunspell</u>.
- 3. Вы можете пересоздать docker-контейнер с новыми настройками с помощью команды:

```
docker-compose up -d --build --force-recreate
```

Пример файла docker-compose.yaml:

```
version: '2.4'
services:
 jacoco:
   image: docker.universe-data.ru/universe/jacoco-agent:latest
   profiles: ["jacoco"]
   volumes:
     - jacoco:/jacoco:ro
 setup_json:
   image: stedolan/jq:latest
   entrypoint: >
     sh -c "
     cd /opt/json configs;
    [ ! -z ${OVERRIDE_JSON:-''} ] &&
     jq -s '.[0] * .[1]' customer.json ${OVERRIDE_JSON}
    > ../customer.json ||
    cp customer.json ../customer.json"
   volumes:
     - ./:/opt
 ui:
   image: ${FRONTEND_IMAGE}
   restart: always
   ports:
     - ${FRONTEND_PORT}:80
   networks:
    - mdm network
   links:
```

- mdm

```
volumes:
   - ${FRONTEND UE:-/dev/null}:/usr/share/nginx/html/CUX
   - ./customer.json:/usr/share/nginx/html/customer.json
 environment:
   BACKEND ADDRESS: ${BACKEND ADDRESS}
   CLIENT_MAX_BODY_SIZE: ${CLIENT_MAX_BODY_SIZE}
   PROXY_SEND_TIMEOUT: ${PROXY_SEND_TIMEOUT}
   PROXY READ TIMEOUT: ${PROXY READ TIMEOUT}
   SEND_TIMEOUT: ${SEND_TIMEOUT}
 depends_on:
   setup_json:
     condition: service_completed_successfully
mdm:
 image: ${BACKEND_IMAGE}
 restart: always
 ports:
   - ${BACKEND PORT}:8080
   - ${JACOCO AGENT PORT:-6300}:6300
 networks:
   - mdm network
 environment:
   GUEST_MODE: ${GUEST_MODE}
   POSTGRES_ADDRESS: postgres-mdm:5432
   POSTGRES_USERNAME: ${MDM_POSTGRES_USER}
   POSTGRES_PASSWORD: ${MDM_POSTGRES_PASSWORD}
   DATABASE NAME: ${MDM POSTGRES DB NAME}
   SEARCH CLUSTER ADDRESS: opensearch-mdm:9200
   SEARCH CLUSTER NAME: docker-cluster
   EMAIL_ENABLED: ${RESTORE_EMAIL_ENABLED}
   EMAIL_SERVER_HOST: ${RESTORE_EMAIL_SERVER HOST}
   EMAIL SERVER PORT: ${RESTORE EMAIL SERVER PORT}
   EMAIL USERNAME: ${RESTORE EMAIL USERNAME}
   EMAIL_PASSWORD: ${RESTORE_EMAIL_PASSWORD}
   EMAIL FRONTEND URL: ${RESTORE EMAIL FRONTEND URL}
   EMAIL_SSL_ENABLE: ${RESTORE_EMAIL_SSL_ENABLE}
   EMAIL STARTTLS ENABLE: ${RESTORE EMAIL STARTTLS ENABLE}
   JAVA TOOL OPTIONS: ${JAVA TOOL OPTIONS:-}
 volumes:
   - jacoco:/jacoco:ro
 depends_on:
   postgres-mdm:
     condition: service healthy
   opensearch-mdm:
     condition: service healthy
postgres-mdm:
 image: postgres:12
 restart: always
 environment:
   POSTGRES_DB: ${MDM_POSTGRES_DB_NAME}
   POSTGRES_USER: postgres
   POSTGRES_PASSWORD: postgres
   MDM_POSTGRES_USER: ${MDM_POSTGRES_USER}
```

```
MDM POSTGRES PASSWORD: ${MDM POSTGRES PASSWORD}
   ports:
     - ${POSTGRES OUTER PORT}:5432
   networks:
     - mdm network
   volumes:
     - ./init-db.sh:/docker-entrypoint-initdb.d/initdb.sh
     - mdm-postgres-data:/var/lib/postgresql/data
   command: postgres -c max prepared transactions=300
   healthcheck:
     test: ["CMD-SHELL", "pg_isready -U postgres -d $MDM_POSTGRES_DB_NAME"]
     interval: 10s
     timeout: 1s
     retries: 20
 opensearch-mdm:
   image: opensearchproject/opensearch:2.7.0
   restart: always
   environment:
     - "OPENSEARCH JAVA OPTS=-Xms512m -Xmx512m"
     - "discovery.type=single-node"
     - "DISABLE SECURITY PLUGIN=true"
   volumes:
     - mdm-opensearch-data:/usr/share/opensearch/data
     - ./hunspell:/usr/share/opensearch/config/hunspell/
   ulimits:
     memlock:
       soft: -1
       hard: -1
   ports:
     - ${OPENSEARCH HTTP OUTER PORT}:9200
   networks:
     - mdm network
   healthcheck:
     test: >
       bash -c "curl http://localhost:9200 | grep '¥"cluster_name¥"'"
     interval: 10s
     timeout: 2s
     retries: 20
volumes:
 mdm-postgres-data:
   driver: local
 mdm-opensearch-data:
   driver: local
 jacoco:
   driver: local
networks:
 mdm network:
   driver: bridge
```

## 1.1.3.1. Описание файла .env

## Совет:

Имена параметров могут различаться в зависимости от редакции (SE/EE/CE), но должны совпадать с параметрами, указанными в файле docker-compose.yaml

- **BACKEND\_IMAGE**, **FRONTEND\_IMAGE** ссылки на backend и frontend образы; различаются в зависимости от редакции.
- **OPENSEARCH\_HTTP\_OUTER\_PORT, POSTGRES\_OUTER\_PORT, FRONTEND\_PORT, BACKEND\_PORT** - внешние порты соответствующих контейнеров, т.е. порты, к которым может извне подключиться docker кластера. Значения могут быть произвольными; имена должны соответствовать файлу docker-compose.yaml.
- MDM\_POSTGRES\_USER, MDM\_POSTGRES\_PASSWORD, MDM\_POSTGRES\_DB\_NAME - имя пользователя, пароль и имя БД; различаются в зависимости от редакции; должны соответствовать файлу docker-compose.yaml.
- **GUEST\_MODE** параметр, позволяющий активировать <u>гостевой доступ</u>; доступен только в Enterprise редакции.
- **RESTORE\_EMAIL** параметры, отвечающие за восстановление пароля.
- **CORE\_SECURITY\_TOKEN\_MANUAL\_PROLONGATION\_ENABLED** параметр, отвечающий за продление сессии пользователя.
- Группа параметров **nginx.conf**:
  - **BACKEND\_ADDRESS** значение берется из одноименной секции в файле docker-compose.yaml - в примере используется значение **mdm**.

Пример файла .env для Enterprise редакции:

```
BACKEND_IMAGE=docker.universe-data.ru/unidata-ee/backend:release-6-10-12ed338
e
FRONTEND_IMAGE=docker.universe-data.ru/unidata-ee/frontend:release-6-10-5dd6a
e69
OPENSEARCH_HTTP_OUTER_PORT=19201
MDM_POSTGRES_USER=postgres
MDM_POSTGRES_PASSWORD=postgres
MDM_POSTGRES_DB_NAME=postgres
POSTGRES_OUTER_PORT=15431
FRONTEND_PORT=8082
BACKEND_PORT=9081
```

#### GUEST\_MODE=false

#TIMEZONE=Europe/Moscow

RESTORE\_EMAIL\_ENABLED=false RESTORE\_EMAIL\_SERVER\_HOST=localhost RESTORE\_EMAIL\_SERVER\_PORT=5025 RESTORE\_EMAIL\_USERNAME=universe@example.com RESTORE\_EMAIL\_PASSWORD=password RESTORE\_EMAIL\_FRONTEND\_URL='' RESTORE\_EMAIL\_SSL\_ENABLE=true RESTORE\_EMAIL\_STARTTLS\_ENABLE=false

#nginx.conf
BACKEND\_ADDRESS=http://mdm:8080
CLIENT MAX BODY SIZE=100m

PROXY\_SEND\_TIMEOUT=600s PROXY\_READ\_TIMEOUT=600s SEND\_TIMEOUT=600s

### Settings for cluster (run command: docker-compose -f
docker-compose-cluster.yaml up -d) ###

FRONTEND\_PORT\_2=8083
BACKEND\_PORT\_2=9082
BACKEND\_ADDRESS\_2=http://mdm2:8080
CACHE\_AUTO\_DETECTION\_ENABLED=true
CACHE\_GROUP=unidata
CACHE\_PASSWORD=password
CACHE\_PORT=5701
CACHE\_PORT\_AUTOINCREMENT=false
CACHE\_TCP\_IP\_ENABLED=true
CACHE\_TCP\_IP\_MEMBERS=mdm,mdm2
CACHE\_PUBLIC\_ADDRESS\_2=mdm2:5701
SYSTEM\_NODE\_ID=node0
SYSTEM\_NODE\_ID\_2=node1
### Settings for cluster ###

## 1.1.3.2. Устранение возможных проблем

После установки может возникать ошибка с файлом customer.json (ее видно в консоли разработчика в браузере, клавиша F12). Ошибка возникает не всегда, иногда права устанавливаются верно. В случае ошибки необходимо проверить права на файл, они должны быть 644. Если права отличаются, выполнить команду:

chmod 644 customer.json

# 1.2. Docker через загрузку образов из файла

Внимание:

Дисклеймер.

Инструкция содержит примерный порядок действий по установке. Инструкция призвана описать логику действий по установке на простом, абстрактном примере. В зависимости от инфраструктуры сервера (кластера) и его конфигурации, вам могут понадобиться дополнительные или иные шаги.

Эта инструкция не должна восприниматься буквально, а использоваться в качестве основы действий.

## 1.2.1. Подготовка

## Подготовка сервера:

• Обновите существующий список пакетов Ubuntu. Команда:

```
``sudo apt update``
```

- Установите Docker.
- Установите Docker Compose (версия выше 1.29).
- Настройте разрешения для Docker Compose. Пример команды:

```
``sudo chmod +x /usr/local/bin/docker-compose``
```

#### Дистрибутив:

• Заранее получите и скачайте образы Docker с дистрибутивом продукта Юниверс.

## 1.2.2. Установка

- Скачайте и распакуйте из архива дистрибутива архивы образов Docker. Пример: universe\_mdm\_hpe\_backend\_docker\_v6.11.zip и universe\_mdm\_hpe\_frontend\_docker\_v6.11.zip. Внутри содержатся образы Docker с расширением .tar.
- 2. Распакуйте на сервер содержимое вышеуказанных архивов.
- 3. Запустите Docker.
- 4. Запустите загрузку образов:

docker load < /opt/universe\_mdm\_hpe\_backend\_docker\_v6.11.tar
docker load < /opt/universe\_mdm\_hpe\_frontend\_docker\_v6.11.tar</pre>

- 5. Дождитесь завершения.
- 6. Создайте файлы конфигурации *docker-compose.yml*, *.env*. Содержимое файлов см. <u>ниже</u>.
  - Расположение файлов на сервере может быть любым. Файлы конфигурации должны читаться через Docker compose.

- 7. Укажите имя образа и тег для файла .env. Используйте один из способов:
  - Измените имя и тег в параметрах BACKEND\_IMAGE и FRONTEND\_IMAGE файла .env. Сохраните изменения.
  - Используйте команду. Пример:

docker image tag repo.tddev.ru/unidata-mdm/backend:v6.11
universe\_mdm\_hpe\_backend:v6.11

- 8. Отредактируйте остальные требуемые параметры в файлах *docker-compose.yml*, *.env*. Сохраните изменения.
- 9. Перейдите в каталог с кодом. Пример команды:

cd universe-mdm-deploy

10. Для запуска системы используйте команду:

docker-compose up -d

11. При запуске интерфейса система потребует лицензию. Установите вашу лицензию.

По умолчанию пользовательский интерфейс доступен для просмотра на *localhost:8082*.

Логин и пароль по умолчанию: admin/ admin. После первого входа в систему необходимо сменить пароль.

## 1.2.3. Обязательные файлы конфигурации

Указанные ниже файлы должны быть созданы вручную.

#### Пример файла .env:

```
BACKEND_IMAGE=docker.universe-data.ru/unidata-ee/backend:release-6-10-12ed338
e
FRONTEND_IMAGE=docker.universe-data.ru/unidata-ee/frontend:release-6-10-5dd6a
e69
OPENSEARCH_HTTP_OUTER_PORT=19201
MDM_POSTGRES_USER=postgres
MDM_POSTGRES_PASSWORD=postgres
MDM_POSTGRES_DB_NAME=postgres
POSTGRES_OUTER_PORT=15431
FRONTEND_PORT=8082
BACKEND_PORT=9081
GUEST_MODE=false
```

#### #TIMEZONE=Europe/Moscow

RESTORE\_EMAIL\_ENABLED=false RESTORE\_EMAIL\_SERVER\_HOST=localhost RESTORE\_EMAIL\_SERVER\_PORT=5025 RESTORE\_EMAIL\_USERNAME=universe@example.com RESTORE\_EMAIL\_PASSWORD=password RESTORE\_EMAIL\_FRONTEND\_URL='' RESTORE\_EMAIL\_SSL\_ENABLE=true RESTORE\_EMAIL\_STARTTLS\_ENABLE=false

#nginx.conf
BACKEND\_ADDRESS=http://mdm:8080
CLIENT\_MAX\_BODY\_SIZE=100m
PROXY\_SEND\_TIMEOUT=600s
PROXY\_READ\_TIMEOUT=600s
SEND\_TIMEOUT=600s

### Settings for cluster (run command: docker-compose -f
docker-compose-cluster.yaml up -d) ###
FRONTEND\_PORT\_2=8083
PACKEND\_PORT\_2 0082

```
BACKEND_PORT_2=9082
BACKEND_ADDRESS_2=http://mdm2:8080
CACHE_AUTO_DETECTION_ENABLED=true
CACHE_GROUP=unidata
CACHE_PASSWORD=password
CACHE_PORT=5701
CACHE_PORT_AUTOINCREMENT=false
CACHE_TCP_IP_ENABLED=true
CACHE_TCP_IP_MEMBERS=mdm,mdm2
CACHE_PUBLIC_ADDRESS_mdm:5701
CACHE_PUBLIC_ADDRESS_2=mdm2:5701
SYSTEM_NODE_ID=node0
SYSTEM_NODE_ID_2=node1
### Settings for cluster ###
```

Где BACKEND\_IMAGE и FRONTEND\_IMAGE: имена Docker-образов и теги

версии. Тег указывается после двоеточия. Пример значений:

- BACKEND\_IMAGE=universe\_mdm\_hpe\_backend:v6.11
- FRONTEND\_IMAGE=universe\_mdm\_hpe\_frontend:v6.11

#### Пример файла docker-compose.yml:

```
version: '2.4'
services:
   jacoco:
    image: docker.universe-data.ru/universe/jacoco-agent:latest
   profiles: ["jacoco"]
    volumes:
        - jacoco:/jacoco:ro
   setup_json:
```

```
image: stedolan/jq:latest
 entrypoint: >
   sh -c "
   cd /opt/json configs;
   [ ! -z ${OVERRIDE_JSON:-''} ] &&
   jq -s '.[0] * .[1]' customer.json ${OVERRIDE_JSON}
   > ../customer.json ||
   cp customer.json ../customer.json"
 volumes:
   - ./:/opt
ui:
 image: ${FRONTEND_IMAGE}
 restart: always
 ports:
   - ${FRONTEND PORT}:80
 networks:
   - mdm_network
 links:
   - mdm
 volumes:
   - ${FRONTEND UE:-/dev/null}:/usr/share/nginx/html/CUX
   - ./customer.json:/usr/share/nginx/html/customer.json
 environment:
   BACKEND_ADDRESS: ${BACKEND_ADDRESS}
   CLIENT_MAX_BODY_SIZE: ${CLIENT_MAX_BODY_SIZE}
   PROXY_SEND_TIMEOUT: ${PROXY_SEND_TIMEOUT}
   PROXY READ TIMEOUT: ${PROXY READ TIMEOUT}
   SEND_TIMEOUT: ${SEND_TIMEOUT}
 depends on:
   setup_json:
     condition: service completed successfully
mdm:
 image: ${BACKEND_IMAGE}
 restart: always
 ports:
   - ${BACKEND PORT}:8080
   - ${JACOCO_AGENT_PORT:-6300}:6300
 networks:
   - mdm_network
 environment:
   GUEST MODE: ${GUEST MODE}
   POSTGRES ADDRESS: postgres-mdm:5432
   POSTGRES USERNAME: ${MDM POSTGRES USER}
   POSTGRES PASSWORD: ${MDM POSTGRES PASSWORD}
   DATABASE NAME: ${MDM POSTGRES DB NAME}
   SEARCH CLUSTER ADDRESS: opensearch-mdm:9200
   SEARCH CLUSTER NAME: docker-cluster
   EMAIL_ENABLED: ${RESTORE_EMAIL_ENABLED}
   EMAIL_SERVER_HOST: ${RESTORE_EMAIL_SERVER_HOST}
   EMAIL_SERVER_PORT: ${RESTORE_EMAIL_SERVER_PORT}
   EMAIL_USERNAME: ${RESTORE_EMAIL_USERNAME}
   EMAIL_PASSWORD: ${RESTORE_EMAIL_PASSWORD}
   EMAIL_FRONTEND_URL: ${RESTORE_EMAIL_FRONTEND_URL}
```

```
EMAIL SSL ENABLE: ${RESTORE EMAIL SSL ENABLE}
   EMAIL_STARTTLS_ENABLE: ${RESTORE_EMAIL_STARTTLS_ENABLE}
   JAVA TOOL OPTIONS: ${JAVA TOOL OPTIONS:-}
 volumes:
    - jacoco:/jacoco:ro
 depends on:
   postgres-mdm:
     condition: service_healthy
   opensearch-mdm:
     condition: service healthy
postgres-mdm:
 image: postgres:12
 restart: always
 environment:
   POSTGRES DB: ${MDM POSTGRES DB NAME}
   POSTGRES_USER: postgres
   POSTGRES_PASSWORD: postgres
   MDM POSTGRES USER: ${MDM POSTGRES USER}
   MDM POSTGRES PASSWORD: ${MDM POSTGRES PASSWORD}
 ports:
   - ${POSTGRES OUTER PORT}:5432
 networks:
   - mdm_network
 volumes:
   - ./init-db.sh:/docker-entrypoint-initdb.d/initdb.sh
   - mdm-postgres-data:/var/lib/postgresql/data
 command: postgres -c max prepared transactions=300
 healthcheck:
   test: ["CMD-SHELL", "pg isready -U postgres -d $MDM POSTGRES DB NAME"]
   interval: 10s
   timeout: 1s
   retries: 20
opensearch-mdm:
 image: opensearchproject/opensearch:2.7.0
 restart: always
 environment:
   - "OPENSEARCH JAVA OPTS=-Xms512m -Xmx512m"
   - "discovery.type=single-node"
   - "DISABLE_SECURITY_PLUGIN=true"
 volumes:
   - mdm-opensearch-data:/usr/share/opensearch/data
   - ./hunspell:/usr/share/opensearch/config/hunspell/
 ulimits:
   memlock:
     soft: -1
     hard: -1
 ports:
    - ${OPENSEARCH HTTP OUTER PORT}:9200
 networks:
   - mdm network
 healthcheck:
   test: >
     bash -c "curl http://localhost:9200 | grep '¥"cluster_name¥"'"
```

# 1.3. Закрытый контур на Ubuntu

#### Внимание:

#### Дисклеймер.

Инструкция содержит примерный порядок действий по установке. Инструкция призвана описать логику действий по установке на простом, абстрактном примере. В зависимости от инфраструктуры сервера (кластера) и его конфигурации, вам могут понадобиться дополнительные или иные шаги.

Эта инструкция не должна восприниматься буквально, а использоваться в качестве основы действий.

#### Примечание:

Ниже приведен пример установки системы на сервер Ubuntu 22.04 minimal installation.

#### Предупреждение:

Инструкция актуальна для установки Юниверс MDM версии 6.9 и старше, так как система перешла с Elasticsearch на Opensearch

## Предварительные действия:

- Архив с дистрибутивом предоставляется клиенту при приобретении продукта через менеджера компании "Юниверс Дата".
- Распакуйте архив дистрибутива, который содержит скрипты для установки, в любое место. Содержимое будет храниться в каталоге *MDM\_6.X\_Ubuntu\_22.04\_offline\_install*. Далее этот каталог будет именоваться как <*OFFLINE\_REP*>.
- Скопируйте содержимое *<OFFLINE\_REP>* на целевой сервер.

# 1.3.1. Установка с помощью bash скрипта

В корне директории MDM\_6.X\_Ubuntu\_22.04\_offline\_install находится скрипт **install.sh**, который устанавливает систему автоматически со стандартными настройками.

## Скрипт содержит 2 переменные:

- **POSTGRES\_PASSWORD** берет значение из файла setenv.sh
- **ТОМСАТ\_НОМЕ** указана стандартная директория для установки Universe.

Также скрипт состоит из функций, последовательность которых описана в конце файла:

- checkdirs проверяет все ли директории, необходимые для установки на месте,
- check\_files проверяет наличие необходимых конфигурационных файлов в папке *conf\_files*,
- install\_mc устанавливает Midnight Commander,
- install\_vim устанавливает Vim,
- install\_Opensearch устанавливает и конфигурирует Opensearch,
- install\_postgres устанавливает и конфигурирует postgresql,
- install\_java устанавливает и конфигурирует OpenJDK,
- install\_universe производит все необходимые действия для запуска Universe.

# 1.3.1.1. Перед запуском скрипта

Файлы конфигурации находятся в папке conf\_files. Убедитесь, что вас устраивает стандартная конфигурация.

- 1. Файл setenv.sh содержит основные настройки Universe:
  - export POSTGRES\_ADDRESS="localhost:5432" adpec cepbepa Postgres,
  - export POSTGRES\_USERNAME="postgres" пользователь БД postgres,
  - export POSTGRES\_PASSWORD="notpostgres" пароль пользователя Postgres,
  - export DATABASE\_NAME="universe" имя базы данных,
  - export SEARCH\_CLUSTER\_NAME="universe-os-cluster" имя кластера Opensearch, которое должно соответствовать значению параметра *cluster.name* в *opencsearch.yml*,
  - export SEARCH\_CLUSTER\_ADDRESS="localhost:9200" адрес кластера Opensearch.
- 2. Файл postgresql.conf содержит основные настойки PostgreSQL Из перечня

настроек необходимо изменить следующие:

- max\_prepared\_transactions = 300
- max\_connections = 1000
- port = 5432
- listen\_addresses = '\*'
- 3. Файл *pg\_hba.conf* содержит настройки безопасности PostgreSQL. Можно изменить данную секцию в соответствии с требуемыми настройками безопасности:

```
# Database administrative Login by Unix domain socket
local all postgres peer
# TYPE DATABASE USER ADDRESS METHOD
# "Local" is for Unix domain socket connections only
local all all md5
# IPv4 Local connections:
host all all 127.0.0.1/32 trust
# IPv6 Local connections:
host all all ::1/128 trust
# ALLow replication connections from Localhost, by a user with the
# replication privilege.
local replication all peer
host replication all 127.0.0.1/32 scram-sha-256
host replication all ::1/128 scram-sha-256
```

- 4. Файл *opensearch.yml* содержит конфигурации Opensearch. Вы можете изменить: cluster.name: universe-os-cluster, path.data: /var/lib/opensearch, path.logs: /var/log/opensearch и другие параметры в соответствии с требованиями к настройкам системы.
- 5. Файл *tomcat.service* необходим в создании tomcat daemon. Не нуждается в редактировании, будет автоматически скопирован скриптом.

## 1.3.1.2. Запуск скрипта

1. Предоставьте скрипту права на исполнение:

sudo chmod +x ./install.sh

2. Запустите скрипт от имени супер-пользователя:

sudo ./install.sh

3. По окончании исполнения скрипта Universe будет доступна по адресу: *http://ваш IP:8080/universe-frontend* (логин и пароль: *admin/admin*)

После введения логина и пароля система запросит файл лицензии и смену пароля.

## 1.3.1.2.1. Ручная установка

## 1.3.1.3. Установка Midnight Commander (опционально)

- Все пакеты и зависимости для Midnight Commander располагаются по пути: ./<*OFFLINE\_REP*>/mc
- Для установки Midnight Commander перейдите в указанную директорию и выполните команду:

sudo dpkg -i \*.deb

## 1.3.1.4. Установка Vim (опционально)

- Все пакеты и зависимости для vim располагаются по пути: ./<*OFFLINE\_REP*>/vim
- Для установки vim перейдите в указанную директорию и выполните команду:

sudo dpkg -i \*.deb

## 1.3.1.5. Установка Opensearch

Используйте один из вариантов установки с помощью команд:

```
wget
```

```
https://artifacts.opensearch.org/releases/bundle/opensearch/2.7.0/opensearch-
2.7.0-linux-x64.deb
```

sudo dpkg -i opensearch-2.7.0-linux-x64.deb

Или:

```
curl -o- https://artifacts.opensearch.org/publickeys/opensearch.pgp | sudo gpg
--dearmor --batch --yes -o /usr/share/keyrings/opensearch-keyring
echo "deb [signed-by=/usr/share/keyrings/opensearch-keyring]
https://artifacts.opensearch.org/releases/bundle/opensearch/2.x/apt stable
main" | sudo tee /etc/apt/sources.list.d/opensearch-2.x.list
sudo apt-get update
sudo apt-get install opensearch=2.7.0
```

#### Запуск Opensearch:

```
sudo systemctl enable opensearch
sudo systemctl start opensearch
```

#### Отключение демо-конфигурации безопасности (обязательно):

В дистрибутиве opensearch по умолчанию включена демо-конфигурация безопасности, ее необходимо отключить:

export DISABLE\_SECURITY\_PLUGIN=true
sudo systemctl restart opensearch

Или добавьте plugins.security.disabled: true в /usr/share/opensearch/config/opensearch.yml

#### Установка плагина Analysis-ICU (обязательно)

- Плагин Analysis-ICU для Opensearch располагается по пути: ./<*OFFLINE\_REP*>/*Opensearch/analysis-icu-7.14.0.zip*
- Для установки плагина используйте команду (сработает, если каталог <OFFLINE\_REP> распакована в домашнюю директорию):

sudo /usr/share/opensearch/bin/opensearch-plugin install analysis-icu

#### Установка hunspell словарей

- Словари hunspell располагаются по пути: ./<OFFLINE\_REP>/Opensearch/hunspell
- Для установки скопируйте директорию /<OFFLINE\_REP>/Opensearch/hunspell в /etc/opensearch/:

sudo cp -rv /path/to/hunspell /etc/opensearch

3. Выдайте права для новой директории:

sudo chown -R root:opensearch /etc/opensearch/hunspell/

4. В случае если готовая директория со словарями отсутствует - составьте ее:

```
cd /etc/opensearch/
sudo mkdir hunspell
cd hunspell/
sudo mkdir ru_RU
sudo mkdir en US
cd /etc/opensearch/hunspell/ru RU
sudo wget
https://cgit.freedesktop.org/libreoffice/dictionaries/plain/ru RU/ru RU.
dic
sudo wget
https://cgit.freedesktop.org/libreoffice/dictionaries/plain/ru_RU/ru_RU.
aff
cd /etc/opensearch/hunspell/en_US
sudo wget
https://cgit.freedesktop.org/libreoffice/dictionaries/plain/en/en_US.dic
sudo wget
https://cgit.freedesktop.org/libreoffice/dictionaries/plain/en/en US.aff
sudo chown -R root:opensearch /etc/opensearch/hunspell/
```

См. также инструкцию по настройке библиотеки Hunspell.

#### Конфигурация cluster.name:

1. Откройте файл конфигурации (также доступно в mc):

sudo vi /etc/opensearch/opensearch.yml

- 2. Замените параметр cluster.name, например, на universe-os-cluster
- 3. После внесенных изменений перезапустите Opensearch:

sudo systemctl restart opensearch

#### Проверка запуска доступна с помощью команды:

curl -X GET '127.0.0.1:9200'

## 1.3.1.6. Установка PostgreSQL 12

- 1. Все пакеты необходимые для корректной установки postgres располагаются по пути: /<OFFLINE\_REP>/postgres
- 2. Для установки postgresql перейдите в указанную директорию и выполните команду:

sudo dpkg -i \*.deb

- 3. Основные конфигурационные файлы postgresql находятся по путям: /etc/postgresql/12/main/postgresql.conf и /etc/postgresql/12/main/pg\_hba.conf.
- 4. В файле /etc/postgresql/12/main/postgresql.conf раскомментируйте и измените следующие параметры: #listen\_addresses = 'localhost', max\_connections = 100, #max\_prepared\_transactions = 0 замените на listen\_addresses = '\*', max\_connections = 1000, max\_prepared\_transactions = 300, a port=5433 замените на port=5432.
- 5. Секции файла /etc/postgresql/12/main/pg\_hba.conf должны приобрести следующий вид (допускается изменение параметров под индивидуальные потребности):

```
# Database administrative login by Unix domain socket
local
      all
                     postgres
                                                          peer
# TYPE DATABASE
                     USER
                                   ADDRESS
                                                          METHOD
# "Local" is for Unix domain socket connections only
local
       all
                     all
                                                         md5
# IPv4 local connections:
                                    127.0.0.1/32
                                                          trust
host
       all
                     all
# IPv6 local connections:
host
       all
                     all
                                    ::1/128
                                                          trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
                    all
local
       replication
                                                          peer
host
       replication
                      all
                                    127.0.0.1/32
scram-sha-256
       replication
                      all
host
                                     ::1/128
scram-sha-256
```

6. После установки и настройки измените пароль пользователю postgres

('notpostgres' - пароль пользователя postgres):

```
sudo su
su postgres
psql
alter user postgres with password 'notpostgres'
```

7. Создайте базу данных с помощью команды:

CREATE DATABASE universe;

## 1.3.1.7. Установка орепЈДК

- Все пакеты необходимые для корректной установки openJDK располагаются по пути: ./<*OFFLINE\_REP>/java*
- Для установки openjdk необходимо перейти в эту директорию и выполнить команду:

sudo dpkg -i \*.deb

## 1.3.1.8. Установка Tomcat

1. Создайте пользователя:

sudo useradd -m -U -d /opt/tomcat-9 -s /bin/false tomcat

- 2. Каталог с tomcat находится в /<OFFLINE\_REP>/tomcat-9.
- 3. После выполнения команды будет создан пользователь tomcat с домашним каталогом /opt/tomcat-9. Скопируйте этот каталог в /opt/:

sudo cp -rv /<OFFLINE\_REP>/tomcat-9/\* /opt/tomcat-9

4. Для создания демона скопируйте

файл ./<OFFLINE REP>/conf files/tomcat.service в /etc/systemd/system/:

cp -v MDM\_6.X\_Ubuntu\_22.04\_offline\_install/conf\_files/tomcat.service /etc/systemd/system/

5. Перезагрузите демоны:

systemctl daemon-reload

#### 1.3.1.9. Установка приложения Universe

- 1. Дистрибутив Universe находится по пути /<*OFFLINE\_REP*>/universe-6.x (6.x номер версии системы Universe).
- 2. Для установки нужно перейти в данную директорию:

cd /<OFFLINE\_REP>/universe-6.x/Tomcat/

3. Скопируйте конфигурации:

```
sudo cp -v bin/setenv.sh /opt/tomcat-9/bin/ && ¥
sudo cp -rv conf/universe /opt/tomcat-9/conf/ && ¥
sudo cp -v libs/* /opt/tomcat-9/lib/ &&¥
sudo cp -v webapps/* /opt/tomcat-9/webapps/
```

4. Предоставьте права директории tomcat-9:

chown -R tomcat:tomcat /opt/tomcat-9

5. Предоставьте права на запуск исполняемым файлам:

chmod +x /opt/tomcat-9/bin/\*.sh

#### Настройка системы:

Для корректного запуска системы необходимо измените следующие файлы:

 /opt/tomcat-9/bin/\*\*setenv.sh - приведите к следующему виду (значение настроек описано в секции "Перед запуском скрипта:

```
export JAVA_OPTS="${JAVA_OPTS}
-Dunidata.conf=${CATALINA_HOME}/conf/universe
-Dlogback.configurationFile=${CATALINA_HOME}/conf/universe/logback.xml"
#export JRE_HOME=/usr/Lib/jvm/java-8-oracle
#export JAVA_HOME=/usr/Lib/jvm/java-8-oracle
export POSTGRES_ADDRESS="localhost:5432"
export POSTGRES_USERNAME="postgres"
export POSTGRES_PASSWORD="notpostgres"
export DATABASE_NAME="universe"
export SEARCH_CLUSTER_NAME="universe-os-cluster"
export SEARCH_CLUSTER_ADDRESS="localhost:9200"
```

• /opt/tomcat-9/conf/universe/backend.properties - добавьте настройки пути хранения лога:

```
org.unidata.mdm.system.bitronix.tm.journal.disk.logPart1Filename=${CATAL
INA_HOME}/logs/btm1.tlog
org.unidata.mdm.system.bitronix.tm.journal.disk.logPart2Filename=${CATAL
INA_HOME}/logs/btm2.tlog
```

#### Запуск системы:

1. Перезагрузите tomcat:

sudo systemctl restart tomcat

- 2. После окончания установки перейдите по адресу: *http://BAШ-IP:8080/universe-frontend* (логин и пароль: *admin/admin*)
- 3. Далее выполните вход в систему Юниверс.
- 1.3.1.9.1. Конфигурация безопасности

Конфигурация безопасности необходима для:

- Генерации сертификатов или их получения из авторитетного источника;
- Размещения сертификатов в Opensearch и Юниверс MDM.
- Настройки credentials MDM клиента.

Примечание:

Ниже представлен пример генерации самоподписанных сертификатов

# 1.3.1.10. Генерация сертификатов

```
# Root CA
openssl genrsa -out root-ca-key.pem 2048
openssl req -new -x509 -sha256 -key root-ca-key.pem -subj
"/C=RU/ST=SAINT-PETERSBURG/L=SAINT-PETERSBURG/O=UNIVERSE/OU=MDM/CN=root.dns.a
-record" -out root-ca.pem -days 730
# Admin cert
openssl genrsa -out admin-key-temp.pem 2048
```

openssl pkcs8 -inform PEM -outform PEM -in admin-key-temp.pem -topk8 -nocrypt -v1 PBE-SHA1-3DES -out admin-key.pem openssl reg -new -key admin-key.pem -subj "/C=RU/ST=SAINT-PETERSBURG/L=SAINT-PETERSBURG/O=UNIVERSE/OU=MDM/CN=A" -out admin.csr openssl x509 -req -in admin.csr -CA root-ca.pem -CAkey root-ca-key.pem -CAcreateserial -sha256 -out admin.pem -days 730 # Node cert openssl genrsa -out node-key-temp.pem 2048 openssl pkcs8 -inform PEM -outform PEM -in node-key-temp.pem -topk8 -nocrypt -v1 PBE-SHA1-3DES -out node-key.pem openssl req -new -key node-key.pem -subj "/C=RU/ST=SAINT-PETERSBURG/L=SAINT-PETERSBURG/O=UNIVERSE/OU=MDM/CN=universe-o s-cluster" -out node.csr echo 'subjectAltName=DNS:universe-os-cluster' > node.ext openssl x509 -reg -in node.csr -CA root-ca.pem -CAkey root-ca.key.pem -CAcreateserial -sha256 -out node.pem -days 730 -extfile node.ext # Cleanup rm admin-key-temp.pem rm admin.csr rm node-key-temp.pem rm node.csr rm node.ext

Флаг - subj задает атрибуты x509 сертификата в скрипте (CN: CommonName, OU:

OrganizationalUnit, O: Organization, L: Locality, S: StateOrProvinceName, C:

CountryName). Если его пропустить, задание пройдет интерактивно.

Предупреждение: CN и DNS сертификатов узлов должны совпадать со значением SEARCH\_CLUSTER\_NAME

## 1.3.1.11. Конфигурация Opensearch

Конфигурация Opensearch может находиться по пути:

/usr/share/opensearch/config/opensearch.yml либо etc/opensearch/opensearch.yml.

Ниже представлен пример самой базовой конфигурации безопасности. Для изучения всех возможностей и настроек см. <u>официальную документацию</u>.

Пример конфигурации файла /usr/share/opensearch/config/opensearch.yml:

```
cluster.name: universe-os-cluster
network.host: 0.0.0.0
discovery.type: single-node
plugins.security.ssl.transport.pemcert_filepath: node.pem
plugins.security.ssl.transport.pemkey_filepath: node-key.pem
plugins.security.ssl.transport.pemtrustedcas_filepath: root-ca.pem
plugins.security.ssl.transport.enforce_hostname_verification: false
plugins.security.ssl.http.enabled: true
plugins.security.ssl.http.pemcert filepath: node.pem
```

```
plugins.security.ssl.http.pemkey filepath: node-key.pem
plugins.security.ssl.http.pemtrustedcas filepath: root-ca.pem
plugins security allow default init security index: true
plugins security authcz admin dn:
'emailAddress=my@email.com,CN=A,OU=MDM,O=UNIVERSE,L=SAINT-PETERSBURG,ST=SAINT
-PETERSBURG, C=RU'
plugins.security.nodes_dn:
'CN=universe-os-cluster,OU=MDM,O=UNIVERSE,L=SAINT-PETERSBURG,ST=SAINT-PETERSB
URG, C=RU'
plugins security.enable_snapshot_restore_privilege: true
plugins security check_snapshot_restore_write_privileges: true
plugins.security.restapi.roles_enabled: ["all_access",
"security_rest_api_access"]
cluster.routing.allocation.disk.threshold enabled: false
plugins.security.audit.config.disabled rest categories: NONE
plugins.security.audit.config.disabled_transport_categories: NONE
```

Отформатированную nodes\_dn (admin\_dn — аналогично) можно получить с

помощью команды:

openssl x509 -subject -nameopt RFC2253 -noout -in node.pem

## 1.3.1.12. Конфигурация Юниверс МDМ

1. Создайте truststore, положив туда сертификат открытого ключа узла:

keytool -import -file node.pem -keystore my-truststore.jks

2. Создайте keystore, а также скомбинируйте сертификаты открытого и закрытого ключа клиента, положив туда получившийся .*p12* сертификат:

```
openssl pkcs12 -export -in admin.pem -inkey admin-key.pem -out
volumes/admin.p12 -CAfile root-ca.srl -caname root
keytool -importkeystore -deststorepass my-keystore-pass -destkeypass
my-keystore-pass -destkeystore my-keystore.jks -srckeystore admin.p12
-srcstoretype PKCS12 -srcstorepass my-p12-pass
```

3. Настройте файл backend.properties:

# If any security measures are enabled (credentials, encryption)
org.unidata.mdm.search.security.enabled=true
# Credentials used by MDM to access search cluster
org.unidata.mdm.search.admin.login=admin
org.unidata.mdm.search.admin.password=admin
# Truststore of type JKS, stores public certificates of search nodes for MDM
to verify
org.unidata.mdm.search.truststore.path=/path/to/my-truststore.jks
org.unidata.mdm.search.truststore.password=my-truststore-pass
# Keystore of type JKS, stores public+private .p12 certificate of MDM for
it to establish an encrypted connection
org.unidata.mdm.search.keystore.path=/path/to/my-keystore.jks
org.unidata.mdm.search.keystore.path=/path/to/my-keystore.jks

4. Последний шаг также можно выполнить через соответствующие переменные окружения:

```
SEARCH_SECURITY_ENABLED=true
```

```
SEARCH_ADMIN_LOGIN=admin
SEARCH_ADMIN_PASSWORD=admin
```

```
SEARCH_TRUSTSTORE_PATH=/path/to/my-truststore.jks
SEARCH_TRUSTSTORE_PASSWORD=my-truststore-pass
```

SEARCH\_KEYSTORE\_PATH=/path/to/my-keystore.jks
SEARCH\_KEYSTORE\_PASSWORD=my-keystore-pass

# **1.4.** Открытый контур на Ubuntu

#### Внимание:

#### Дисклеймер.

Инструкция содержит примерный порядок действий по установке. Инструкция призвана описать логику действий по установке на простом, абстрактном примере. В зависимости от инфраструктуры сервера (кластера) и его конфигурации, вам могут понадобиться дополнительные или иные шаги.

Эта инструкция не должна восприниматься буквально, а использоваться в качестве основы действий.

#### Примечание:

Ниже приведен пример установки системы на сервер Ubuntu 22.04 minimal installation.

#### Предупреждение

Инструкция актуальна для установки Юниверс MDM версии 6.9 и старше, так как система перешла с Elasticsearch на Opensearch.

#### Предварительные действия:

Обновите списки репозиториев и установите основные пакеты, позволяющие работать с файлами:

```
sudo apt update && sudo apt upgrade && ¥
sudo apt install mc vim -y
```

# 1.4.1. Установка Opensearch

Рекомендуется устанавливать Opensearch 2.7.0 в закрытом контуре. Подробную инструкцию см. по <u>ссылке</u>.

## 1.4.2. Установка PostgreSQL

1. Предварительно установите обязательные пакеты:

sudo apt install vim curl wget gpg gnupg2 software-properties-common
apt-transport-https lsb-release ca-certificates -y

2. Добавьте официальный репозиторий:

```
curl -fsSL https://www.postgresql.org/media/keys/ACCC4CF8.asc|sudo gpg
--dearmor -o /etc/apt/trusted.gpg.d/postgresql.gpg && sudo sh -c 'echo "deb
http://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" >
/etc/apt/sources.list.d/pgdg.list' && sudo apt update
```

3. Произведите установку:

```
sudo apt install postgresql-12 -y
```

4. PostgreSQL должен запуститься автоматически - проверьте запуск службы:

systemctl status postgresql@12-main.service

- 5. Основные конфигурационные файлы postgresql находятся по пути: /etc/postgresql/12/main/postgresql.conf и /etc/postgresql/12/main/pg\_hba.conf
- 6. В файле /etc/postgresql/12/main/postgresql.conf раскомментируйте и измените следующие параметры: #listen\_addresses = 'localhost', max\_connections = 100, #max\_prepared\_transactions = 0 замените на listen\_addresses = '\*', max\_connections = 1000, max\_prepared\_transactions = 300, a port=5433 замените на port=5432.
- Секции файла /etc/postgresql/12/main/pg\_hba.conf должны приобрести следующий вид (допускается изменение параметров под индивидуальные потребности):

```
# Database administrative login by Unix domain socket
local
      all
                      postgres
                                                           peer
# TYPE DATABASE
                     USER
                                    ADDRESS
                                                           METHOD
# "local" is for Unix domain socket connections only
local
      all
                      all
                                                           md5
# IPv4 local connections:
host
      all
                                    127.0.0.1/32
                                                           trust
                     all
# IPv6 local connections:
      all
                                                           trust
host
                     all
                                    ::1/128
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all host replication all
                                                           peer
                      all
                                     127.0.0.1/32
scram-sha-256
       replication all
host
                                     ::1/128
scram-sha-256
```

8. После установки и настройки измените пароль пользователю postgres

('notpostgres' - пароль пользователя postgres):

```
sudo su
su postgres
psql
alter user postgres with password 'notpostgres';
```

9. Создайте базу данных с помощью команды:

CREATE DATABASE universe;

## **1.4.3.** Установка openJDK

#### Примечание:

```
Ниже приведен пример установки openJDK 11
```

• Выполните следующие команды:

```
sudo apt install default-jre
sudo apt install default-jdk
```

• Проверьте версию:

javac -version

## 1.4.4. Установка Tomcat

- Скачайте архив Tomcat 9 с официального сайта <u>https://dlcdn.apache.org/tomcat/</u>.
- 2. Создайте пользователя:

sudo useradd -m -U -s /bin/false tomcat

3. Распакуйте архив в директорию /opt:

sudo tar -xvf apache-tomcat-9.x.xx.tar.gz -C /opt/

4. Перейдите в папку /opt, переименуйте имя папки и предоставьте права пользователю tomcat:

```
cd /opt && mv apache-tomcat-9.x.xx tomcat-9 && chown -R tomcat:tomcat tomcat-9
```

5. Создайте systemd файл tomcat по пути /etc/systemd/system/tomcat.service:

```
sudo vi /etc/systemd/system/tomcat.service
```

6. Наполните файл следующим содержимым:

```
[Unit]
Description=Apache Tomcat Web Application Container
After=network.target
[Service]
Type=forking
Environment=JAVA HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd64
Environment=CATALINA PID=/opt/tomcat-9/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat-9
Environment=CATALINA_BASE=/opt/tomcat-9
Environment='CATALINA OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'
Environment='JAVA_OPTS=-Djava.awt.headless=true
-Djava.security.egd=file:/dev/./urandom'
ExecStart=/opt/tomcat-9/bin/startup.sh
ExecStop=/opt/tomcat-9/bin/shutdown.sh
User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
Restart=always
[Install]
WantedBy=multi-user.target
```

7. Сохраните и выйдите:

:wq

8. Перезагрузите демоны:

systemctl daemon-reload

## 1.4.5. Установка приложения Universe

Установка Universe производится в офлайн режиме. Подробную инструкцию см. по <u>ссылке</u>.

# 1.5. Закрытый контур на Astra Linux

#### Внимание:

#### Дисклеймер.

Инструкция содержит примерный порядок действий по установке. Инструкция призвана описать логику действий по установке на простом, абстрактном примере. В зависимости от инфраструктуры сервера (кластера) и его конфигурации, вам могут понадобиться дополнительные или иные шаги.

Эта инструкция не должна восприниматься буквально, а использоваться в качестве основы действий.

#### Примечание:

Ниже приведен пример установки системы в закрытом контуре (без интернета) на Astra Linux 1.7.Х "Смоленск". В инструкцию включена настройка SSL

#### Предупреждение:

Инструкция актуальна для установки Юниверс MDM версии 6.9 и старше, так как система перешла с Elasticsearch на Opensearch.

#### Предварительные действия:

- Архив с дистрибутивом предоставляется клиенту при приобретении продукта через менеджера компании "Юниверс Дата".
- Распакуйте архив дистрибутива, который содержит скрипты для установки, в любое место. Содержимое будет храниться в каталоге MDM\_manual\_install\_Astra\_1.7. Далее этот каталог будет именоваться как <0FFLINE REP>.
- Скопируйте содержимое < OFFLINE\_REP> на целевой сервер.

## 1.5.1. Установка ЈАVА

#### Примечание:

Java требуется установить на всех серверах, где планируется запускать Opensearch, Tomcat.

Все пакеты, необходимые для корректной установки орепJDK, располагаются в каталоге *Java*.

#### Для установки Java:

- 1. Перейдите в директорию Java.
- 2. Распакуйте содержимое архива с установочными пакетами.
- 3. В распакованном каталоге выполните команду:

sudo dpkg -i \*.deb

## 1.5.2. Установка Opensearch

#### Для установки Opensearch:

1. Перейдите в каталог Opensearch и используйте команду:

sudo dpkg -i \*.deb

#### 1.5.2.1. Установка словарей

1. Скопируйте директорию ./<*OFFLINE\_REP*>/*Opensearch/hunspell* в /*etc/opensearch/*:

sudo cp -rv ./<OFFLINE\_REP>/Opensearch/hunspell /etc/opensearch

2. Выдайте права для новой директории:

sudo chown -R root:opensearch /etc/opensearch/hunspell/

См. также инструкцию по настройке библиотеки Hunspell.

#### 1.5.2.2. Установка плагина

1. Выполните команду с указанием полного пути до архива analysis-icu-.zip\*:

sudo /usr/share/opensearch/bin/opensearch-plugin install
/<OFFLINE\_REP>/Opensearch/analysis-icu-2.7.0.zip

# если вы в директории с файлом analysis-icu-\*.zip # sudo /usr/share/opensearch/bin/opensearch-plugin install file://`pwd`/analysis-icu-2.7.0.zip

2. Результат выполнения команды будет иметь вид:

Installed analysis-icu with folder name analysis-icu

## 1.5.2.3. Конфигурация Opensearch

1. Очистите файл конфигурации и откройте следующий файл с помощью любого редактора:

```
sudo cp -i /etc/opensearch/opensearch.yml
/etc/opensearch/opensearch.yml_before_ssl && >
/etc/opensearch/opensearch.yml # Οчистка φαŭла
sudo vi /etc/opensearch/opensearch.yml
```

2. Укажите параметр cluster.name, например:

cluster.name: mdm-os-cluster

Имя кластера будет использоваться в настройках приложения для подключения. Каждый параметр в файле указываем с новой строки.

 По умолчанию Opensearch прослушивает только localhost, если приложение Тотсаt устанавливается на другой сервер, и/или Opensearch будет использоваться в кластере, то необходимо разрешить подключения с других интерфейсов, указав параметр:

```
network.host: 0.0.0.0
```

- 4. Также необходимо открыть порт 9200 для нод/приложений, которые будут подключаться к Opensearch.
- 5. Если планируется использование только одной ноды Opensearch укажите параметр:

discovery.type: single-node

6. Укажите каталог для логов и данных:

```
path.data: /var/lib/opensearch
path.logs: /var/log/opensearch
```

7. Если не требуется SSL шифрование и авторизация, то укажите параметр:

```
plugins.security.disabled: true
```

Пример итогового файла opensearch.yml для запуска Opensearch на одной ноде без ssl и авторизации:

```
cluster.name: mdm-os-cluster
network.host: 0.0.0.0
discovery.type: single-node
path.data: /var/lib/opensearch
path.logs: /var/log/opensearch
plugins.security.disabled: true
```

## 1.5.2.4. Настройка кластера Opensearch

#### Примечание:

Приведен пример конфигурации без использования ssl

Для кластерной конфигурации и использования нескольких серверов Opensearch в файле */etc/opensearch/opensearch.yml* на каждом сервере измените ряд настроек:

1. Задайте уникальный node.name для каждой ноды кластера Opensearch:

```
# для первого сервера:
node.name: node01
# для N сервера
# node.name: nodeN
```

2. Перечислите все Hostname или ір серверов, которые планируется соединить в кластер в следующих параметрах:

```
cluster.initial_master_nodes: ["10.10.24.90","10.10.24.91", "10.10.24.92"]
discovery.seed_hosts: ["10.10.24.90", "10.10.24.91", "10.10.24.92"]
```

3. Закомментируйте или уберите следующий параметр, т.к. он противоречит кластерной настройке:

```
#discovery.type: single-node
```

Серверы кластера Opensearch взаимодействуют между собой, используя порт 9300, который должен быть открыт между ними.

## 1.5.2.5. Настройки потребления RAM Opensearch

Необходимо настроить объем выделяемого ОЗУ в файле /etc/opensearch/jvm.options. Действие выполняется на каждом узле кластера.

Значения выделяемой ОЗУ не должно быть более 50% от общего объема RAM (при условии, что на сервере не устанавливаются другие ресурсоемкие приложения) и не более 32 Гб. Хт должен быть равен Хтх. В примере, указанном ниже, значение равно 16 Гб:

| -Xms16g |  |  |  |
|---------|--|--|--|
| -Xmx16g |  |  |  |

## 1.5.3. Настройка SSL Opensearch

#### Совет:

Если не требуется подключение по ssl описанные шаги можно пропустить

Для работы SSL требуются сертификаты, генерация которых происходит на одной ноде, затем они копируются на остальные ноды.

1. На каждом узле удалите или переместите стандартные демо сертификаты в сторонний каталог, т.к. они могут помешать запуску:

cd /etc/opensearch && mkdir -p /opt/\_os\_default\_demo\_certs\_ && mv \*.pem /opt/\_os\_default\_demo\_certs\_

2. Сгенерируйте сертификаты (на одной из нод) и перейдите в каталог для

сертификатов:

mdkir -p /etc/opensearch/ssl && cd /etc/opensearch/ssl

3. Создайте корневой сертификат и сертификат администратора, который потребуется для управления настройками:

```
# Root CA
openssl genrsa -out root-ca-key.pem 2048
openssl req -new -x509 -sha256 -key root-ca-key.pem -subj
"/C=CA/O=ORG/OU=UNIT/CN=root-ca.dns.record" -out root-ca.pem -days 9999
# Admin cert
openssl genrsa -out admin-key-temp.pem 2048
openssl pkcs8 -inform PEM -outform PEM -in admin-key-temp.pem -topk8 -nocrypt
-v1 PBE-SHA1-3DES -out admin-key.pem
openssl req -new -key admin-key.pem -subj "/C=CA/O=ORG/OU=UNIT/CN=A" -out
admin.csr
openssl x509 -req -in admin.csr -CA root-ca.pem -CAkey root-ca-key.pem
-CAcreateserial -sha256 -out admin.pem -days 9999
# очистите промежуточные файлы, которые больше не нужны
rm admin-key-temp.pem
rm admin.csr
# days n - срок действия сертификата в днях
```

4. Создайте сертификат для каждой ноды кластера Opensearch:

```
# Node cert 1
openssl genrsa -out node1-key-temp.pem 2048
openssl pkcs8 -inform PEM -outform PEM -in node1-key-temp.pem -topk8 -nocrypt
-v1 PBE-SHA1-3DES -out node1-key.pem
openssl req -new -key node1-key.pem -subj
"/C=CA/O=ORG/OU=UNIT/CN=opensearch-node-1" -out node1.csr
echo 'subjectAltName=DNS:opensearch-node-1' > node1.ext
openssl x509 -req -in node1.csr -CA root-ca.pem -CAkey root-ca-key.pem
-CAcreateserial -sha256 -out node1.pem -days 9999 -extfile node1.ext
# Node cert 2
openssl genrsa -out node2-key-temp.pem 2048
openssl pkcs8 -inform PEM -outform PEM -in node2-key-temp.pem -topk8 -nocrypt
-v1 PBE-SHA1-3DES -out node2-key.pem
openssl reg -new -key node2-key.pem -subj
"/C=CA/O=ORG/OU=UNIT/CN=opensearch-node-2" -out node2.csr
echo 'subjectAltName=DNS:opensearch-node-2' > node2.ext
openssl x509 -req -in node2.csr -CA root-ca.pem -CAkey root-ca-key.pem
-CAcreateserial -sha256 -out node2.pem -days 9999 -extfile node2.ext
# очистите промежуточные файлы, которые больше не нужны
rm node1-key-temp.pem
rm node1.csr
rm node1.ext
rm node2-key-temp.pem
```

```
rm node2.csr
rm node2.ext
```

- Выше представлен пример для двух нод, для каждой последующей важно:
  - Заменить все названия файлов в команде, содержащие nodeN (т.е. node1-key-temp.pem -> node3-key-temp.pem; node1-key.pem -> node3-key.pem ... и т.д.)
  - Указать в параметре CN= действительный hosname/dns адрес для каждой ноды opensearch: -subj "/C=CA/0=ORG/OU=UNIT/CN=opensearch-node-1"
- 5. Скопируйте на все ноды Opensearch получившиеся файлы в каталог */etc/opensearch/ssl* и выдайте права на каталог и файлы:

```
chown opensearch:opensearch -R /etc/opensearch/ssl
chown +x /etc/opensearch/ssl
chmod 600 /etc/opensearch/ssl/*
chmod 644 /etc/opensearch/ssl/root-ca.pem
```

## 1.5.3.1. Конфигурация файла opensearch.yml (SSL)

1. На каждой ноде очистите файл конфигурации и откройте файл (любым редактором):

```
sudo cp -i /etc/opensearch/opensearch.yml
/etc/opensearch/opensearch.yml_before_ssl && >
/etc/opensearch/opensearch.yml # Ouucmka @aŭna
sudo vi /etc/opensearch/opensearch.yml
```

2. Вставьте следующее содержимое:

```
cluster.name: universe-os-cluster
network.host: opensearch-node-1
node.name: node01
cluster.initial_master_nodes: ["opensearch-node-1","opensearch-node-2"]
discovery.seed_hosts: ["opensearch-node-2"]
# для кластера параметр ниже должен быть удален, или закомментирован
#discovery.type: single-node
plugins.security.ssl.transport.pemcert_filepath:
/etc/opensearch/ssl/node1.pem
plugins.security.ssl.transport.pemkey_filepath:
/etc/opensearch/ssl/node1.key.pem
plugins.security.ssl.transport.pemtrustedcas_filepath:
/etc/opensearch/ssl/root-ca.pem
plugins.security.ssl.transport.enforce_hostname_verification: false
# enable ssl 9200
```

```
plugins.security.ssl.http.enabled: true
plugins.security.ssl.http.pemcert filepath: /etc/opensearch/ssl/node1.pem
plugins.security.ssl.http.pemkey filepath:
/etc/opensearch/ssl/node1-key.pem
plugins security ssl http pemtrustedcas filepath:
/etc/opensearch/ssl/root-ca.pem
plugins.security.allow_default_init_securityindex: true
plugins security authcz admin_dn:
  - 'CN=A,OU=UNIT,O=ORG,C=CA'
plugins security nodes_dn:
 - 'CN=opensearch-node-1,OU=UNIT,O=ORG,C=CA'
 - 'CN=opensearch-node-2,OU=UNIT,O=ORG,C=CA'
plugins.security.enable_snapshot_restore privilege: true
plugins security check snapshot restore write privileges: true
plugins.security.restapi.roles enabled: ["all access",
'security_rest_api_access"]
cluster.routing.allocation.disk.threshold enabled: false
plugins security audit config disabled_rest_categories: NONE
plugins security audit config disabled transport categories: NONE
plugins.security.allow unsafe democertificates: false
path.data: /var/lib/opensearch
path.logs: /var/log/opensearch
```

- 3. Замените описания следующих параметров:
- Общие для кластера параметры в файле:
  - cluster.name должен быть одинаковый для всех узлов кластера
  - cluster.initial\_master\_nodes указываются все ноды кластера
  - network.host: 0.0.0.0 при указании 0.0.0.0 используются все доступные интерфейсы и Ір-адреса, в ином случае можно указать конкретный Ір-адрес текущего узла, доступный для всех нод
  - plugins.security.authcz.admin\_dn указываются атрибуты админского ключа, получить значение в требуемом формате можно командой openssl x509 -subject -nameopt RFC2253 -noout -in admin.pem
  - В параметре plugins.security.nodes\_dn указываются все ноды кластера. Должны быть указаны те атрибуты, которые были использованы в команде при генерации сертификатов:
    - o CN=opensearch-node-1,OU=UNIT,O=ORG,C=CA
    - CN=opensearch-node-2,OU=UNIT,O=ORG,C=CA. Формат отличается от того, что указан при генерации. Чтобы отобразить в требуемом виде, можно выполнить команды для каждого сертификата: openssl x509
       -subject -nameopt RFC2253 -noout -in node1.pem openssl
x509 -subject -nameopt RFC2253 -noout -in node2.pem
openssl x509 -subject -nameopt RFC2253 -noout -in
nodeN.pem

- Уникальные для каждой ноды параметры:
  - Уникальный node.name
  - Для параметров укажите путь к сертификатам для текущей ноды кластера, например:
    - o plugins.security.ssl.http.pemcert\_filepath: /etc/opensearch/ssl/node1.pem
    - o plugins.security.ssl.http.pemkey\_filepath: /etc/opensearch/ssl/node1-key.pem
  - Укажите соседние ноды кластера в параметре discovery.seed\_hosts. Если используются 3 ноды, то для opensearch-node-1 параметр будет выглядеть следующим образом ["opensearch-node-2", "opensearch-node-3"]

# 1.5.3.2. Создание хранилища ключей для использования в приложении при подключении к Opensearch (SSL)

1. Создайте truststore и поместите туда сертификат открытого ключа узла:

```
keytool -import -trustcacerts -file node1.pem -keystore app-truststore.jks
```

- 2. Введите и подтвердите пароль. Например, для MY-TrustStore-pswd введите yes на вопрос Trust this certificate?
- 3. Создайте keystore и пометите туда сертификат открытого ключа узла:

```
openssl pkcs12 -export -in admin.pem -inkey admin-key.pem -out admin.p12
-CAfile root-ca.srl -caname root
keytool -importkeystore -destkeystore app-keystore.jks -srckeystore
admin.p12 -srcstoretype PKCS12
```

4. Введите и подтвердите пароль. Например, MY-p\_12-pswd:

```
keytool -importkeystore -destkeystore app-keystore.jks -srckeystore
admin.p12 -srcstoretype PKCS12
```

- 5. Задайте новый пароль и подтвердить его, он потребуется в конфигурации приложения. Например, MY-keyStore-pswd.
- 6. Введите ранее введенный пароль MY-p\_12-pswd.
  - Команду можно выполнить без интерактивного ввода паролей, передав их через аргументы, например:

keytool -importkeystore -deststorepass MY-keyStore-pswd -destkeypass MY-keyStore-pswd -destkeystore app-keystore.jks -srckeystore admin.p12 -srcstoretype PKCS12 -srcstorepass MY-p\_12-pswd

7. Удалите промежуточный файл:

rm -f admin.p12

Пароли и полученные файлы . jks потребуются при настройке приложения

### 1.5.4. Запуск Opensearch

1. Обновите службы:

sudo systemctl daemon-reload

2. Добавьте Opensearch в автозагрузку:

sudo systemctl enable opensearch.service

3. Запустите Opensearch:

sudo systemctl start opensearch.service

4. Проверьте статус:

```
sudo systemctl status opensearch.service
```

#### 1.5.4.1. Проверка запуска Opensearch

1. Проверьте ноды Opensearch:

curl 'https://localhost:9200' -k -u 'admin:admin'

2. Проверьте статус кластера:

```
curl 'https://localhost:9200/_cluster/health?pretty' -k -u 'admin:admin'
```

Примечание:

Важно, чтобы "status" = "green", а имя статуса и количество нод совпадали с настроенным значением

#### 1.5.4.2. Смена логина/пароля для доступа к Opensearch

Пользователи Opensearch хранятся в файле

/etc/opensearch/opensearch-security/internal\_users.yml. Пароль указан в виде хэша, пример:



```
cd /usr/share/opensearch/plugins/opensearch-security/tools && ¥
./securityadmin.sh -cd /etc/opensearch/opensearch-security -icl -nhnv -h
localhost ¥
    -cacert /etc/opensearch/ssl/root-ca.pem ¥
    -cert /etc/opensearch/ssl/admin.pem¥
    -key /etc/opensearch/ssl/admin-key.pem
```

3. Результат успешного выполнения команды: Done with success

Скрипт записывает данные в индекс для всего кластера, чтобы избежать несогласованности и возможного применения команды с другого узла, следует перенести изменения данного файла на все ноды кластера.

В файле internal\_users.yml также можно переименовать пользователя admin, всех остальных удалить и применить конфигурацию.

# 1.5.5. Установка PostgreSQL 12

Все пакеты необходимые для корректной установки располагаются в директории Postgresql.

#### Для установки postgresql:

1. Перейдите в директорию Postgresql и выполните команду:

sudo dpkg -i \*.deb

- 2. Основные конфигурационные файлы postgresql находятся в каталоге:
- /etc/postgresql/12/main/postgresql.conf
- /etc/postgresql/12/main/pg\_hba.conf
- 3. В файле /etc/postgresql/12/main/postgresql.conf задайте перечисленные параметры следующим образом. Также замените port=5433 на port=5432:

```
listen_addresses = '*'
max_connections = 1000
max_prepared_transactions = 300
```

4. В конец файла /etc/postgresql/12/main/pg\_hba.conf добавьте строку, которая разрешит подключение к хосту по паролю, для всех баз:

host all all all scram-sha-256

5. Перезагрузите postgresql для применения настроек (*donyckaemcs изменение параметров под индивидуальные потребности*):

systemctl restart postgresql-12

6. Продолжите настройку в консоли. Зайдите в базу данных под пользователем:

```
sudo su
su postgres
psql
```

7. Задайте пароль пользователя:

ALTER USER postgres WITH PASSWORD 'notpostgres\_change\_me'

8. Создайте базу данных для приложения (доступен выбор собственного логического наименования, которое будет использовано в дальнейшем):

CREATE DATABASE universe;

9. Поменяйте timezone (например, MSK+3):

```
ALTER SYSTEM SET timezone TO 'W-SU';
ALTER SYSTEM SET random_page_cost TO 1.1;
```

10. Раскомментируйте и задайте размер буфера. При ОЗУ 1 ГБ и более значение shared buffers составляет 25% от объема памяти:

ALTER SYSTEM SET shared buffers TO '4GB'

11. Перезагрузите postgresql для применения дополнительных настроек:

systemctl restart postgresql-12

12. Проверьте подключение к базе и запрос пароля:

psql -U postgres -h localhost

#### 1.5.6. Установка Tomcat

Для запуска Тотсат требуется Java (описание установки см. выше).

1. Создайте пользователя для запуска tomcat:

```
sudo useradd -r tomcat -s /sbin/nologin
```

- 2. Распакуйте дистрибутив из каталога apache-tomcat-9.0.\*.tar.gz в каталог /opt/.
- 3. Переименуйте каталог /opt/apache-tomcat-9.0.\* в /opt/tomcat-9 (версия может отличаться).
- 4. Удалите стандартные файлы и каталоги Manager App, которые могут содержаться в дистрибутиве tomcat из каталога /opt/tomcat-9/webapps:

```
rm -rf /opt/tomcat-9/webapps/*
```

5. Выдайте права на директорию tomcat для пользователя tomcat:

```
chown -R tomcat:tomcat /opt/tomcat-9
```

6. Добавьте сервис, создав файл tomcat.service:

```
sudo vi /etc/systemd/system/tomcat.service
```

со следующим содержимым, задав параметры потребления RAM в

CATALINA\_OPTS:

```
[Unit]
Description=Apache Tomcat Web Application Container
After=network.target
[Service]
Type=forking
Environment=JAVA_HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd64
Environment=CATALINA PID=/opt/tomcat-9/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat-9
Environment=CATALINA_BASE=/opt/tomcat-9
Environment='CATALINA_OPTS=-Xms1024M -Xmx2048M -server -XX:+UseParallelGC'
Environment='JAVA_OPTS=-Djava.awt.headless=true
-Djava.security.egd=file:/dev/./urandom'
WorkingDirectory=/opt/tomcat-9/
ExecStart=/opt/tomcat-9/bin/startup.sh
ExecStop=/opt/tomcat-9/bin/shutdown.sh
User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
Restart=always
[Install]
WantedBy=multi-user.target
```

7. Примените конфигурацию сервисов:

```
systemctl daemon-reload
```

# 1.5.7. Настройка SSL/TLS Tomcat

#### Примечание:

В приведенном ниже примере используется имя переменной **\$CATALINA\_BASE** для ссылки на базовый каталог, по отношению к которому разрешается большинство относительных путей. Если вы не настроили Tomcat для нескольких экземпляров, задав каталог CATALINA\_BASE, то для **\$CATALINA\_BASE** будет установлено значение **\$CATALINA\_HOME** — каталога, в котором был установлен Tomcat.

#### Чтобы установить и настроить поддержку SSL/TLS на Tomcat:

1. Создайте файл *keystore* для хранения закрытого ключа и самоподписного сертификата сервера, выполнив команду:

\$JAVA\_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA

- 2. Укажите значение пароля "changeit".
- Раскомментируйте запись SSL HTTP/1.1 Connector в файле \$CATALINA\_BASE/conf/server.xml и измените ее, как описано ниже.

## 1.5.7.1. Подготовка хранилища ключей сертификатов

#### Примечание:

В настоящее время Tomcat работает только с хранилищами ключей формата JKS, PKCS11 или PKCS12. Каждая запись в хранилище ключей идентифицируется строкой псевдонима. Например, спецификация PKCS11 требует, чтобы псевдонимы были чувствительны к регистру. Чтобы избежать проблем, связанных с чувствительностью псевдонимов к регистру, не рекомендуется использовать псевдонимы, отличающиеся только регистром.

#### Совет:

Чтобы импортировать существующий сертификат в хранилище ключей JKS, ознакомьтесь с оф.документацией (в пакете документации JDK) о keytool. Обратите внимание, что OpenSSL часто добавляет читаемые комментарии перед ключом, но keytool этого не поддерживает. Если в вашем сертификате есть комментарии перед ключевыми данными, удалите их перед импортом сертификата с помощью keytool.

1. Чтобы импортировать сертификат, подписанный вашим собственным СА, в хранилище ключей PKCS12 с помощью OpenSSL - выполните команду:

```
openssl pkcs12 -export -in mycert.crt -inkey mykey.key
        -out mycert.p12 -name tomcat -CAfile myCA.crt
        -caname root -chain
```

- Для более сложных случаев см. <u>оф.документацию OpenSSL</u>.
- 2. Чтобы создать новое хранилище ключей JKS с нуля, содержащее один

самоподписной сертификат, выполните следующие действия в командной строке терминала:

\$JAVA\_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA

В качестве безопасного алгоритма следует предпочесть алгоритм RSA, который также обеспечивает общую совместимость с другими серверами и компонентами.

3. Используемая команда создаст новый файл с именем ".keystore" в домашнем каталоге пользователя, в котором она запущена. Чтобы указать другое местоположение или имя файла, добавьте к команде keytool, указанной выше, параметр -keystore, за которым следует полное имя пути к вашему файлу keystore. Также необходимо отразить это новое местоположение в файле конфигурации server.xml, как описано далее. Например:

\$JAVA\_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA
 -keystore /path/to/my/keystore

- 4. После выполнения команды введите пароль хранилища ключей ("changeit" по умолчанию). При желании вы можете указать собственный пароль. Также необходимо указать пароль в конфигурационном файле *server.xml*, как описано далее.
- 5. При необходимости добавьте информацию (компания, имя контактного лица и т.д.) для отображения пользователям, которые попытаются получить доступ к защищенной странице в приложении.
- 6. Введите пароль ключа, который является паролем именно для этого сертификата (в отличие от любых других сертификатов, хранящихся в том же файле keystore). Подсказка keytool сообщит вам, что при нажатии клавиши ENTER автоматически используется тот же пароль для ключа, что и для хранилища ключей. Вы можете использовать тот же пароль или выбрать собственный. Если вы выберете пароль, отличный от пароля хранилища ключей, вам также нужно будет указать пользовательский пароль в файле конфигурации server.xml.
- 7. В результате действий будет сконфигурирован файл хранилища ключей с сертификатом, который может быть использован сервером.

Более подробно см. в официальной документации Tomcat.

# 1.5.7.2. Настройка файла конфигурации

1. Откройте файл server.xml в текстовом редакторе (файл находится в папке

conf).

2. В файле найдите блок:

```
<!--
Define a SSL/TLS HTTP/1.1 Connector on port 8443
This connector uses the NIO implementation with the JSSE engine.
When using the JSSE engine, the JSSE configuration attributes
must be used. -->
```

 Ниже найденного блока пропишите конфигурацию коннектора: укажите порт 8443 (параметр port), правильное название файла хранилища ключей (параметр keystoreFile) и пароль (параметр keystorePass).

Пример конфигурации:

```
<Connector URIEncoding="UTF-8" port="8080" acceptCount="100"
enableLookups="false" maxThreads="150" redirectPort="8443" />
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
maxThreads="150" scheme="https" secure="true"
keystoreFile="/usr/share/tomcat/key/server.keystore"
keystorePass="uWePKrfemC"/>
```

4. Сохраните изменения в файле и перезапустите Tomcat.

## 1.5.8. Установка приложения Universe

Для запуска приложения требуется Tomcat и Java (описание установки см. выше).

#### 1.5.8.1. Начало установки

- 1. Установка .war файлов: в каталоге Application\_Tomcat содержатся 2 архива: frontend или backend. В каждом архиве содержится .war файлы, которые необходимо скопировать в директорию /opt/tomcat-9/webapps/.
- 2. Установка дополнительных параметров и библиотек: в архиве backend содержится каталог Tomcat, содержимое которого необходимо скопировать в каталог /opt/tomcat-9/. Находясь в распакованном каталоге архива backend, перейдите в каталог tomcat и скопируйте файлы в директорию сервиса:

```
sudo cp -v bin/setenv.sh /opt/tomcat-9/bin/ && ¥
sudo cp -rv conf/universe /opt/tomcat-9/conf/ && ¥
sudo cp -v libs/* /opt/tomcat-9/lib/
```

3. Предоставьте права на файлы для сервиса:

```
chown -R tomcat:tomcat /opt/tomcat-9
chmod +x /opt/tomcat-9/bin/*.sh
```

## 1.5.8.2. Настройка приложения

1. Основные параметры задаются в переменных в файле setenv.sh. Отредактируйте файл:

```
vi /opt/tomcat-9/bin/setenv.sh
```

2. Установите/добавьте следующие переменные, установив значения соответствующие логическому имени переменной:

```
# имеющийся параметр JAVA_OPTS HE затрагиваются
# укажите параметры подключение к базе
export POSTGRES_ADDRESS="localhost:5432"
export POSTGRES_USERNAME="postgres"
export POSTGRES_PASSWORD="notpostgres_change_me"
export DATABASE_NAME="universe"
# укажите параметры подключение к Opensearch:
export SEARCH_CLUSTER_NAME="universe-os-cluster"
export SEARCH_CLUSTER_NAME="universe-os-cluster"
export SEARCH_CLUSTER_ADDRESS="localhost:9200"
# при использовании кластера Opensearch в переменной SEARCH_CLUSTER_ADDRESS
перечисляются через запятую все ноды (hostname или ip), пример:
#
SEARCH_CLUSTER_ADDRESS=opensearch-node-1:9200, opensearch-node-2:9200, ope
nsearch-node-3:9200
```

# 1.5.8.3. Настройка приложения для подключения к Opensearch через SSL

1. Создайте каталог для сертификатов:

```
sudo mkdir /opt/tomcat-9/ssl
```

2. Скопируйте созданные на предыдущих шагах .jks файлы в созданный каталог:

```
cp -v *¥.jks /opt/tomcat-9/ssl
chown -R tomcat:tomcat /opt/tomcat-9/ssl
```

3. Основные параметры задаются в переменных в файле *setenv.sh*. Отредактируйте файл:

```
vi /opt/tomcat-9/bin/setenv.sh
```

4. Установите/добавьте следующие переменные, введя значения, соответствующие логическому имени переменной:

```
export SEARCH CLUSTER NAME="universe-os-cluster"
# в качестве адреса требуется использовать dns имя, которые прописано в
сертификате в CN
export SEARCH CLUSTER ADDRESS="opensearch-node-1:9200"
# при использовании кластера Opensearch в переменной SEARCH_CLUSTER_ADDRESS
перечисляются через запятую все ноды (hostname или ip), пример:
#
SEARCH_CLUSTER_ADDRESS=opensearch-node-1:9200, opensearch-node-2:9200, ope
nsearch-node-3:9200
export SEARCH SECURITY ENABLED=true
# ичетные данные для авторизации в Opensearch
export SEARCH ADMIN LOGIN=admin
export SEARCH ADMIN PASSWORD=admin
# укажите сформированные для Opensearch jks файлы и пароли от них
export SEARCH TRUSTSTORE PATH=/opt/tomcat-9/ssl/app-truststore.jks
export SEARCH TRUSTSTORE PASSWORD=MY-TrustStore-pswd
export SEARCH_KEYSTORE_PATH=/opt/tomcat-9/ssl/app-keystore.jks
export SEARCH_KEYSTORE_PASSWORD=MY-keyStore-pswd
```

#### 1.5.8.4. Кластерная настройка приложения

1. Если планируется использование нескольких серверов Tomcat, то для

настройки кластерной конфигурации приложения в файле

*/opt/tomcat-9/conf/universe/backend.properties* на каждом сервере требуется изменить ряд настроек:

vi /opt/tomcat-9/conf/universe/backend.properties

2. Следующие параметры одинаковые для каждого узла приложения:

```
# включите pacnpedeленный кэш
org.unidata.mdm.system.cache.tcp-ip.enabled=true
# nepeчислите все ноды tomcat, ip или hostname
org.unidata.mdm.system.cache.tcp-ip.members=server-192-168-106-110,serve
r-192-168-106-111
# стандартный порт, при необходимости можно заменить, должен быть
открыт для всех узлов кластера приложения
org.unidata.mdm.system.cache.port=5701
```

3. Параметр ниже должен быть уникальный для каждого узла в кластере приложения, пример значения параметра:

```
org.unidata.mdm.system.node.id=node1
#org.unidata.mdm.system.node.id=nodeN # для остальных N серверов
```

4. Пример сообщения в логе (*logs/catalina.out*), позволяющее определить то, что приложения объединились в кластер:

#### 1.5.8.5. Запуск приложения

• Управление осуществляется через сервис, на каждом узле tomcat:

```
sudo systemctl start tomcat
sudo systemctl status tomcat
# sudo systemctl restart tomcat
# sudo systemctl stop tomcat
```

• Логи приложения находятся в каталоге /opt/tomcat-9/logs/.

После окончания установки выполните вход в систему Юниверс.

# 1.6. Открытый контур на Astra Linux

#### Внимание:

#### Дисклеймер.

Инструкция содержит примерный порядок действий по установке. Инструкция призвана описать логику действий по установке на простом, абстрактном примере. В зависимости от инфраструктуры сервера (кластера) и его конфигурации, вам могут понадобиться дополнительные или иные шаги.

Эта инструкция не должна восприниматься буквально, а использоваться в качестве основы действий.

#### Примечание:

Ниже приведен пример установки системы на Astra Linux 1.7.Х "Смоленск".

Предупреждение:

Инструкция актуальна для установки Юниверс MDM версии 6.9 и старше, так как система перешла с Elasticsearch на Opensearch.

#### Предварительные действия:

• Обновите списки репозиториев и установите основные пакеты, позволяющие работать с файлами:

sudo apt update && sudo apt upgrade &&  $\tt X$  sudo apt install mc vim -y

## 1.6.1. Установка ЈАVА

1. Выполните следующую команду:

sudo apt install default-jdk

2. Проверьте версию (open JDK 11):

javac -version

## 1.6.2. Установка Opensearch

Рекомендуется устанавливать Opensearch 2.7.0 в закрытом контуре. Подробную инструкцию см. по ссылке.

# 1.6.3. Установка PostgreSQL

1. Настройте репозитории. Содержание файла /etc/apt/sources.list должно приобрести следующий вид:

```
# Astra Linux repository description https://wiki.astralinux.ru/x/0oLiC
# Основной репозиторий
deb https://dl.astralinux.ru/astra/stable/1.7 x86-64/repository-main/
1.7 x86-64 main contrib non-free
# Оперативные обновления основного репозитория
deb https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-update/
1.7 x86-64 main contrib non-free
# Базовый репозиторий
deb https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-base/
1.7 x86-64 main contrib non-free
# Расширенный репозиторий
deb https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-extended/
1.7_x86-64 main contrib non-free
# Расширенный репозиторий (компонент astra-ce)
deb https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-extended/
1.7 x86-64 astra-ce
```

2. После обновления файла выполните команду:

sudo apt update && sudo apt install postgresql-12 -y

3. Проверьте, запущен ли демон после установки:

systemctl status postgresql

• Расположение основных конфигурационных файлов:

/etc/postgresql/12/main/postgresql.conf
/etc/postgresql/12/main/pg\_hba.conf

# 1.6.3.1. Настройка PostgreSQL

1. Выполните вход в сессию служебного пользователя postgres:

sudo su postgres

2. Установите пароль администратора СУБД:

psql -c "alter user postgres with password '<указать\_пароль>';"

- 3. Вместо текста <пароль> установите свой пароль
  - Пароль заключается в одинарные кавычки;
  - Вся команда заключается в двойные кавычки.
- 4. Создайте базу данных universe:

psql -c "CREATE DATABASE universe;"

5. Завершить работу в сессии служебного пользователя postgres:

exit

#### 1.6.3.2. Haстройка postgresql.conf и pg\_hba.conf

1. Настройте postgresql.conf в текстовом редакторе:

```
vi /etc/postgresql/12/main/postgresql.conf
```

2. Проверьте значения и раскомментируйте следующие параметры (являются параметрами по умолчанию):

```
max_prepared_transactions = 300
max_connections = 1000
port = 5432
listen_addresses = '*'
```

3. Настройте postgresql.conf в текстовом редакторе:

```
vi /etc/postgresql/12/main/pg_hba.conf
```

4. Секции файла /etc/postgresql/12/main/pg\_hba.conf должны приобрести следующий вид:

```
# Database administrative login by Unix domain socket
local
      all
                      postgres
                                                            peer
# TYPE DATABASE
                      USER
                                     ADDRESS
                                                            METHOD
# "Local" is for Unix domain socket connections only
local
      all
                      all
                                                           md5
# IPv4 local connections:
                                     127.0.0.1/32
host
       all
                                                            trust
                     all
# IPv6 local connections:
host
       all
                    all
                                     ::1/128
                                                            trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
       replication all replication all
local
                                                            peer
host
                       all
                                      127.0.0.1/32
scram-sha-256
       replication
                       all
host
                                      ::1/128
scram-sha-256
```

5. Для применения изменений перезапустите PostgreSQL:

```
sudo systemctl restart postgresql
```

- 6. Настройте активные сетевые экраны, разрешив доступ к сетевому порту postgresql (по умолчанию порт 5432).
  - Для сетевого экрана ufw:

sudo ufw allow 5432/tcp

• Для сетевого экрана firewalld:

sudo firewall-cmd --add-service=postgresql --zone=internal --permanent

Возможная проблема при добавлении порта в ufw (может возникнуть при dist update без перезагрузки): ERROR: Couldn't determine iptables version

Решение:

```
sudo update-alternatives --set iptables /usr/sbin/iptables-legacy
sudo shutdown -r now
sudo ufw allow ssh
sudo ufw allow 5432/tcp
sudo ufw enable
sudo ufw status ##Προβερκα cmamyca ufw
```

# 1.6.4. Установка Tomcat

- Скачайте архив Tomcat 9 с официального сайта <u>https://dlcdn.apache.org/tomcat/</u>.
- 2. Создайте пользователя:

sudo useradd -m -U -s /bin/false tomcat

- 3. Каталог с tomcat находится в /<OFFLINE\_REP>/tomcat-9. После выполнения команды выше будет создан пользователь tomcat с домашним каталогом /opt/tomcat-9.
- 4. Скопируйте папку в /opt/:

sudo cp -rv /<OFFLINE\_REP>/tomcat-9/\* /opt/tomcat-9

5. Для создания демона скопируйте

файл ./<OFFLINE\_REP>/conf\_files/tomcat.service B /etc/systemd/system/:

cp -v MDM\_6.X\_Astra\_1.7.X/conf\_files/tomcat.service /etc/systemd/system/

6. Перезагрузите демоны:

systemctl daemon-reload

#### 1.6.5. Установка приложения Universe

Установка Universe производится в офлайн режиме. Подробную инструкцию см. по <u>ссылке</u>.

# 1.7. Открытый контур на Red OS 8

#### Внимание

#### Дисклеймер.

Инструкция содержит примерный порядок действий по установке. Инструкция призвана описать логику действий по установке на простом, абстрактном примере. В зависимости от инфраструктуры сервера (кластера) и его конфигурации, вам могут понадобиться дополнительные или иные шаги.

Эта инструкция не должна восприниматься буквально, а использоваться в качестве основы действий.

#### Примечание:

Ниже приведен пример установки системы на РЕД ОС 8 в стандартной редакции и конфигурации "Сервер".

#### Предупреждение:

Инструкция актуальна для установки Юниверс MDM версии 6.9 и старше, так как система перешла с Elasticsearch на Opensearch.

#### Шаги установки:

- <u>Настройка ОС</u>
- Установка пакетов:

- Установка ОрепЈДК,
- <u>Установка Opensearch</u>,
- <u>Vctahobka PostgreSQL</u>,
- Установка Тотсаt,
- <u>Установка системы Universe</u>.

# 1.7.1. Настройка ОС

Для удобства установки необходимо настроить на целевой машине демон SSH (в конфигурации "Сервер" данный компонент настроен по умолчанию).

Пользователь должен иметь права администратора при использовании sudo (по умолчанию при создании пользователя во время установки OC).

# 1.7.2. Установка пакетов

1. Зайдите под гоот пользователем:

sudo su

2. Выполните обновление системы после свежей установки:

dnf update -y

3. Выполните установку Midnight Commander и Vim (опционально):

dnf install mc vim -y

# 1.7.2.1. Установка OpenJDK

1. Выполните установку OpenJDK:

```
dnf install java-11-openjdk.x86_64 -y
```

2. Проверьте установку пакета OpenJDK:

java -version

Корректный вывод выглядит следующим образом:

```
openjdk version "11.0.17" 2022-10-18 LTS
OpenJDK Runtime Environment (Red_OS-11.0.17.0.8-3.red80.1) (build 11.0.17+8-LTS)
OpenJDK 64-Bit Server VM (Red_OS-11.0.17.0.8-3.red80.1) (build 11.0.17+8-LTS,
mixed mode, sharing)
```

# 1.7.2.2. Установка Opensearch

Рекомендуется устанавливать Opensearch версии 2.7.0.

1. Скачайте пакет Opensearch и установите с помощью команды:

```
wget
https://artifacts.opensearch.org/releases/bundle/opensearch/2.7.0/opense
arch-2.7.0-linux-x64.rpm && dnf install ./opensearch-2.7.0-linux-x64.rpm -y
```

- 2. Скопируйте плагины для Opensearch в удобное место на сервере (в примере используется домашняя директория */home/test*).
- 3. Перейдите в директорию /home/test:

cd /home/test

4. Распакуйте архив:

unzip plugins.zip

5. Скопируйте содержимое распакованной директории hunspell в каталог настроек Opensearch:

```
cp -rv ./hunspell /etc/opensearch/
```

6. Задайте корректного владельца скопированной директории:

chown -R opensearch:opensearch /etc/opensearch/hunspell/

7. Создайте копию конфигурационного файла opensearch.yml:

cp /etc/opensearch/opensearch.yml /etc/opensearch/opensearch.yml\_bak

8. Очистите содержимое оригинального файла:

echo > /etc/opensearch/opensearch.yml

9. Откройте файл для редактирования:

vim /etc/opensearch/opensearch.yml

10. Наполняем файл следующим содержимым:

```
cluster.name: universe-os-cluster
network.host: 0.0.0.0
discovery.type: single-node
path.data: /var/lib/opensearch
path.logs: /var/log/opensearch
plugins.security.disabled: true
```

11. Сохраните изменения и выйдите:

:wq

12. Настройте количество выделяемой оперативной памяти для Opensearch. Для этого откройте файл /etc/opensearch/jvm.options:

vim /etc/opensearch/jvm.options

- 13. Измените значение следующих параметров в файле: -Xms1g и -Xmx1g например, на -Xms4g и -Xmx4g.
- 14. Сохраните изменения и выйдите:

:wq

15. Запустите Opensearch:

systemctl start opensearch

16. Установите analysis-icu-2.7.0:

/usr/share/opensearch/bin/opensearch-plugin install file:/home/test/analysis-icu-2.7.0.zip

где <u>\*</u>/home/test/analysis-icu-2.7.0.zi\*р замените на свой путь до analysis-icu-2.7.0.zip

17. Перезапустите Opensearch:

systemctl restart opensearch

18. Убедитесь, что запуск произошел успешно:

systemctl status opensearch

19. Добавьте службу в автозагрузку:

systemctl enable opensearch

Примечание:

- Рекомендуется выделять количество оперативной памяти не менее 25% от объема виртуальной машины. 25% - это оптимальное число, но если Opensearch будет расположен на отдельной ноде, то можно выделить 90% объема.
- Архив analysis-icu-2.7.0.zip должен быть распакован из архива plugins.zip.

# 1.7.2.3. Установка PostgreSQL

1. Установите необходимые пакеты PostgreSQL с помощью команды:

```
dnf install postgresql14.x86_64 postgresql14-contrib.x86_64
postgresql14-devel.x86_64 postgresql14-server.x86_64 -y
```

2. Добавьте службу в автозагрузку:

```
sudo systemctl enable postgresql-14
```

3. Инициализируйте базу данных:

```
sudo postgresql-14-setup initdb
```

Основные конфигурационные файлы postgresql находятся по пути:

- /var/lib/pgsql/14/data/postgresql.conf
- /var/lib/pgsql/14/data/pg\_hba.conf
- 4. В файле /var/lib/pgsql/l4/data/postgresql.conf раскомментируйте и измените следующие параметры: #listen\_addresses = 'localhost', max\_connections = 100, #max\_prepared\_transactions = 0 Ha listen\_addresses = '\*', max\_connections = 1000, max\_prepared\_transactions = 300.

Секции файла /var/lib/pgsql/14/data/pg\_hba.conf должны приобрести следующий вид (допускается изменение параметров под индивидуальные потребности):

| <i># Databas</i><br>local a | se administrat <sup>.</sup><br>11 | ive login by Un<br>postgres | ix domain socket       | peer     |  |
|-----------------------------|-----------------------------------|-----------------------------|------------------------|----------|--|
|                             |                                   |                             |                        | •        |  |
| # TYPE D                    | DATABASE                          | USER                        | ADDRESS                | METHOD   |  |
| # "local"                   | ' is for Unix d                   | domain socket c             | onnections only        |          |  |
| local a                     | 11                                | all                         |                        | md5      |  |
| # IPv4 Lo                   | ocal connection                   | ns:                         |                        |          |  |
| host a                      | 11                                | all                         | 127.0.0.1/32           | trust    |  |
| # IPv6 Lo                   | ocal connection                   | ns:                         |                        |          |  |
| host a                      | 11                                | all                         | ::1/128                | trust    |  |
| # Allow r                   | replication con                   | nnections from              | localhost, by a user w | nith the |  |
| # replication privileae.    |                                   |                             |                        |          |  |
| local r                     | eplication                        | all                         |                        | peer     |  |
| host r                      | eplication                        | all                         | 127.0.0.1/32           |          |  |
| scram-sha-256               |                                   |                             |                        |          |  |
| host r                      | eplication                        | all                         | ::1/128                |          |  |
| scram-sha-256               |                                   |                             |                        |          |  |

5. Запустите postgresql:

```
systemctl start postgresql-14
```

6. Поменяйте пароль пользователю postgres:

```
sudo su
su postgres
psql -U postgres -h localhost
alter user postgres with password 'notpostgres';
```

где notpostgres - пароль пользователя postgres.

7. Создайте базу данных с помощью команды:

CREATE DATABASE universe;

### 1.7.2.4. Установка Tomcat

1. Скачайте архив Tomcat 9 (ссылка из примера может устареть, в этом случае найдите актуальную версию на <u>официальном сайте</u>):

```
wget
https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.87/bin/apache-tomca
t-9.0.87.tar.gz
```

2. Распакуйте скачанный архив в директорию /opt с помощью команды:

```
sudo tar -xvf ./apache-tomcat-9.0.87.tar.gz -C /opt
```

3. Выполните вход под root пользователем:

sudo su

4. Создайте пользователя tomcat:

```
sudo useradd -m -U -s /bin/false tomcat
```

5. Переименуйте *apache-tomcat-9.х* и дайте права пользователю tomcat на этот каталог:

cd /opt && mv apache-tomcat-9.x tomcat-9 ; chown -R tomcat:tomcat tomcat-9

9.х - актуальная версия tomcat. **Важно**: команда может не сработать, если версия tomcat различается - сверьте версию со скачанным архивом.

6. Создайте systemd файл tomcat по пути /etc/systemd/system/tomcat.service:

sudo vim /etc/systemd/system/tomcat.service

7. Вставьте в файл следующее содержимое:

```
[Unit]
Description=Apache Tomcat Web Application Container
After=network.target
```

```
[Service]
Type=forking
Environment=JAVA HOME=/usr/lib/jvm/java-11-openjdk-11.0.17.0.8-3.red80.1
.x86 64
Environment=CATALINA PID=/opt/tomcat-9/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat-9
Environment=CATALINA_BASE=/opt/tomcat-9
Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'
Environment='JAVA OPTS=-Djava.awt.headless=true
-Djava.security.egd=file:/dev/./urandom'
ExecStart=/opt/tomcat-9/bin/startup.sh
ExecStop=/opt/tomcat-9/bin/shutdown.sh
User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
Restart=always
[Install]
WantedBy=multi-user.target
```

```
Важно: замените путь до JAVA на свой, если он отличается, в строке
Environment=JAVA_HOME=/usr/lib/jvm/java-11-openjdk-11.0.17.0.8-3.red80.1
.x86_64
```

8. Сохраните изменения и выйдите:

:wq

9. Перечитайте демоны:

systemctl daemon-reload

10. Добавьте Тотсат в автозагрузку:

systemctl enable tomcat

# 1.7.3. Установка системы Universe

Дистрибутив системы Universe состоит из двух архивов (6.x - актуальная версия системы):

- universe\_mdm\_hpe\_backend\_v6.x.zip
- universe\_mdm\_hpe\_frontend\_v6.x.zip
- 1. Скопируйте дистрибутив Universe на сервер в удобное для вас место (в примере используется */home/test*).
- 2. Распакуйте дистрибутив с помощью команды (*6.х* актуальная версия системы):

mkdir universe\_frontend universe\_backend ; unzip universe\_mdm\_hpe\_backend\_v6.x.zip -d ./universe\_backend && unzip universe\_mdm\_hpe\_frontend\_v6.x.zip -d ./universe\_frontend

3. Скопируйте файлы frontend:

cp universe\_frontend/universe-frontend.war /opt/tomcat-9/webapps/

4. Скопируйте файлы backend и выдайте права с помощью одной команды:

```
cp -v universe_backend/universe-backend.war /opt/tomcat-9/webapps/ && ¥
cp -v universe_backend/Tomcat/bin/setenv.sh /opt/tomcat-9/bin/ && ¥
chmod +x /opt/tomcat-9/bin/* && ¥
cp -rv universe_backend/Tomcat/conf/universe /opt/tomcat-9/conf/ && ¥
cp -v universe_backend/Tomcat/libs/* /opt/tomcat-9/lib/ && ¥
chown -R tomcat:tomcat /opt/tomcat-9
```

5. Заполните /opt/tomcat-9/bin/setenv.sh, приведя файл к следующему виду:

```
vim /opt/tomcat-9/bin/setenv.sh
export JAVA_OPTS="${JAVA_OPTS}
-Dunidata.conf=${CATALINA_HOME}/conf/universe
-Dlogback.configurationFile=${CATALINA_HOME}/conf/universe/logback.xml"
#export JRE_HOME=/usr/Lib/jvm/java-8-oracle
#export JAVA_HOME=/usr/Lib/jvm/java-8-oracle
export POSTGRES_ADDRESS="localhost:5432"
export POSTGRES_USERNAME="postgres"
export POSTGRES_PASSWORD="notpostgres"
export DATABASE_NAME="universe"
export SEARCH_CLUSTER_NAME="universe-os-cluster"
export SEARCH_CLUSTER_ADDRESS="localhost:9200"
```

где: POSTGRES\_ADDRESS - адрес ноды PostgreSQL; POSTGRES\_USERNAME - имя пользователя PostgreSQL: POSTGRES\_PASSWORD - пароль пользователя PostgreSQL; DATABASE\_NAME - имя базы данных; SEARCH\_CLUSTER\_NAME - имя кластера Opensearch; SEARCH\_CLUSTER\_ADDRESS - перечисление участников кластера Opensearch.

6. Сохраните изменения:

:wq

7. Откройте для редактирования /opt/tomcat-9/conf/universe/backend.properties:

vim /opt/tomcat-9/conf/universe/backend.properties

 Добавьте настройки пути хранения лога (в любое место файла в новой строке): org.unidata.mdm.system.bitronix.tm.journal.disk.logPart1Filename=\${CATAL INA\_HOME}/logs/btm1.tlog org.unidata.mdm.system.bitronix.tm.journal.disk.logPart2Filename=\${CATAL INA\_HOME}/logs/btm2.tlog

9. Запустите tomcat:

systemctl start tomcat

После запуска Тотсаt должно пройти некоторое время, после чего система Юниверс будет доступна по адресу *http://<your-server-ip>:8080/universe-frontend*.

После окончания установки выполните вход в систему.

# **1.8.** Открытый контур на ALT Linux 10

#### Внимание:

#### Дисклеймер.

Инструкция содержит примерный порядок действий по установке. Инструкция призвана описать логику действий по установке на простом, абстрактном примере. В зависимости от инфраструктуры сервера (кластера) и его конфигурации, вам могут понадобиться дополнительные или иные шаги.

Эта инструкция не должна восприниматься буквально, а использоваться в качестве основы действий.

#### Примечание:

Ниже приведен пример установки системы на ALT Linux 10 в конфигурации "Сервер".

#### Предупреждение:

Инструкция актуальна для установки Юниверс MDM версии 6.9 и старше, так как система перешла с Elasticsearch на Opensearch.

#### Шаги установки:

- Настройка ОС
- Установка пакетов:
  - <u>Установка OpenJDK</u>,
  - <u>Установка Opensearch</u>,
  - <u>Установка PostgreSQL</u>,
  - <u>Установка Tomcat</u>,
  - <u>Установка системы Universe</u>.

#### 1.8.1. Настройка ОС

Для удобства установки необходимо настроить на целевой машине демон SSH (в

конфигурации "Сервер" данный компонент настроен по умолчанию).

#### 1.8.2. Установка пакетов

1. Установка должна производиться пользователем с правами администратора (root):

su root

2. Выполните обновление системы после свежей установки:

apt-get update -y

3. Выполните установку Midnight Commander и Vim (опционально):

apt-get install mc vim -y

#### 1.8.2.1. Установка OpenJDK

1. Выполните установку OpenJDK:

apt-get install java-11-openjdk -y

2. Проверьте установку пакета OpenJDK:

java -version

Корректный вывод должен выглядеть так:

```
openjdk version "11.0.22.1" 2024-01-16
OpenJDK Runtime Environment 18.9 (build 11.0.22.1+1)
OpenJDK 64-Bit Server VM 18.9 (build 11.0.22.1+1, mixed mode, sharing)
```

#### **1.8.2.2. Установка Opensearch**

Рекомендуется устанавливать актуальную версию Opensearch 2.7.0.

1. Предварительно установите обязательные пакеты:

apt-get install wget dnf

2. Скачайте пакет Opensearch и установите его:

```
wget
https://artifacts.opensearch.org/releases/bundle/opensearch/2.7.0/opense
arch-2.7.0-linux-x64.rpm && dnf install ./opensearch-2.7.0-linux-x64.rpm -y
```

- 3. Скопируйте плагины для Opensearch в удобное место на сервере, в примере это будет домашняя директория */home/user*.
- 4. Перейдите в директорию /home/user, или ту, куда была произведена загрузка:

cd /home/user

5. Распакуйте архив:

unzip plugins.zip

6. Скопируйте содержимое распакованной директории hunspell в каталог настроек Opensearch:

```
cp -rv ./hunspell /etc/opensearch/
```

7. Задайте корректного владельца скопированной директории:

chown -R opensearch:opensearch /etc/opensearch/hunspell/

8. Создайте копию конфигурационного файла opensearch.yml:

cp /etc/opensearch/opensearch.yml /etc/opensearch/opensearch.yml\_bak

9. Откройте файл /etc/opensearch/opensearch.yml для редактирования и наполните его следующим содержимым:

```
cluster.name: universe-os-cluster
network.host: 0.0.0.0
discovery.type: single-node
path.data: /var/lib/opensearch
path.logs: /var/log/opensearch
plugins.security.disabled: true
```

10. Сохраните изменения и выйдите:

:wq

11. Настройте количество выделяемой оперативной памяти для Opensearch. Для этого откройте файл /etc/opensearch/jvm.options:

vim /etc/opensearch/jvm.options

- 12. Измените значение следующих значений в файле: Xmslg и Xmxlg, например на: Xms4g и Xmx4g.
- 13. Сохраните изменения и выйдите:

:wq

14. Запустите Opensearch:

systemctl start opensearch

15. Проверьте его состояние:

systemctl status opensearch

16. Установите *analysis-icu-2.7.0*. Для этого в директории с загруженным архивом *analysis-icu-2.7.0.zip* воспользуемся командой:

/usr/share/opensearch/bin/opensearch-plugin install analysis-icu

17. Перезапустите Opensearch:

systemctl restart opensearch

18. Убедитесь в успешном запуске:

systemctl status opensearch

19. Добавьте службу в автозагрузку:

systemctl enable /usr/lib/systemd/system/opensearch.service

Проверка запуска доступна с помощью команды:

curl -X GET '127.0.0.1:9200'

Примечание:

- Рекомендуется выделять количество оперативной памяти не менее 25% от объема виртуальной машины. 25% - это оптимальное число, но если Opensearch будет расположен на отдельной ноде, то можно выделить 90% объема.
- Архив analysis-icu-2.7.0.zip должен быть распакован из архива plugins.zip.

# 1.8.2.3. Установка PostgreSQL

1. Установите необходимые пакеты PostgreSQL с помощью команды:

```
apt-get install postgresql14 postgresql14-server postgresql14-contrib
postgresql14-server-devel -y
```

2. Добавьте службу в автозагрузку:

systemctl enable postgresql

3. Инициализируйте базу данных:

/etc/init.d/postgresql initdb

Основные конфигурационные файлы postgresql находятся по пути:

- /var/lib/pgsql/data/postgresql.conf
- /var/lib/pgsql/data/pg\_hba.conf

4. В файле /var/lib/pgsql/14/data/postgresql.conf раскомментируйте и измените следующие параметры: #listen\_addresses = 'localhost', max\_connections = 100, #max\_prepared\_transactions = 0, port=5433 Ha listen\_addresses = '\*', max\_connections = 1000, max\_prepared\_transactions = 300, port=5432.

Секции файла /var/lib/pgsql/14/data/pg\_hba.conf должны приобрести следующий вид (допускается изменение параметров под индивидуальные потребности):

| # Database administrative login by Unix domain socket              |                 |                 |                 |        |  |
|--|-----------------|-----------------|-----------------|--------|--|
| TOCAT  | all             | postgres        |                 | peer   |  |
| # TYPE   | DATABASE        | USER            | ADDRESS         | METHOD |  |
| # "Loca  | l" is for Unix  | domain socket c | onnections only |        |  |
| local  | all             | all             |                 | md5    |  |
| # IPv4   | local connectio | ns:             |                 |        |  |
| host   | all             | all             | 127.0.0.1/32    | trust  |  |
| # IPv6   | local connectio | ns:             |                 |        |  |
| host   | all             | all             | ::1/128         | trust  |  |
| # Allow replication connections from localhost, by a user with the |                 |                 |                 |        |  |
| # replication privilege.   |                 |                 |                 |        |  |
| local  | replication     | all             |                 | peer   |  |
| host   | replication     | all             | 127.0.0.1/32    |        |  |
| scram-sha-256  |                 |                 |                 |        |  |
| host   | replication     | all             | ::1/128         |        |  |
| scram-s  | ha-256          |                 |                 |        |  |

5. Запустите postgresql:

```
systemctl start postgresql-14
```

6. Поменяйте пароль пользователю postgres:

```
su - postgres -s /bin/bash
psql
alter user postgres with password 'notpostgres';
```

где notpostgres - пароль пользователя postgres.

7. Создайте базу данных с помощью команды:

CREATE DATABASE universe;

#### 1.8.2.4. Установка Tomcat

1. Скачайте архив Tomcat 9 (ссылка из примера может устареть, в этом случае найдите актуальную версию на <u>официальном сайте</u>):

wget

https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.87/bin/apache-tomca t-9.0.87.tar.gz

2. Распакуйте скачанный архив в директорию /opt с помощью команды:

tar -xvf ./apache-tomcat-9.0.87.tar.gz -C /opt

3. Выполните вход под гоот пользователем:

su-

4. Создайте пользователя tomcat:

useradd -m -U -s /bin/false tomcat

5. Выйдите из него обратно в пользователя root:

exit

6. Переименуйте *apache-tomcat-9.x* и дайте права пользователю tomcat на этот каталог:

```
cd /opt && mv apache-tomcat-9.x tomcat-9 ; chown -R tomcat:tomcat tomcat-9
```

9.х - актуальная версия tomcat. **Важно**: команда может не сработать, если версия tomcat различается - сверьте версию со скачанным архивом.

7. Создайте systemd файл tomcat по пути /etc/systemd/system/tomcat.service:

vim /etc/systemd/system/tomcat.service

8. Вставьте в файл следующее содержимое:

```
[Unit]
Description=Apache Tomcat Web Application Container
After=network.target
[Service]
Type=forking
Environment=JAVA HOME=/usr/lib/jvm/java-11-openjdk-11.0.22.0.7-0.x86 64
Environment=CATALINA_PID=/opt/tomcat-9/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat-9
Environment=CATALINA_BASE=/opt/tomcat-9
Environment='CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'
Environment='JAVA OPTS=-Djava.awt.headless=true
-Djava.security.egd=file:/dev/./urandom'
ExecStart=/opt/tomcat-9/bin/startup.sh
ExecStop=/opt/tomcat-9/bin/shutdown.sh
User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
Restart=always
```

[Install] WantedBy=multi-user.target

**Важно**: замените путь до JAVA на свой, если он отличается, в строке Environment=JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-11.0.17.0.8-3.red80.1 .x86\_64

9. Сохраните изменения и выйдите:

:wq

10. Перечитайте демоны:

systemctl daemon-reload

11. Добавьте Тотсат в автозагрузку:

systemctl enable tomcat

## 1.8.3. Установка системы Universe

Дистрибутив системы Universe состоит из двух архивов (6.x - актуальная версия системы):

- universe\_mdm\_hpe\_backend\_v6.x.zip
- universe\_mdm\_hpe\_frontend\_v6.x.zip
- 1. Скопируйте дистрибутив Universe на сервер в удобное для вас место (в примере используется */home/user*).
- 2. Распакуйте дистрибутив с помощью команды (*6.х* актуальная версия системы):

```
mkdir universe_frontend universe_backend && unzip
universe_mdm_hpe_backend_v6.10.2.zip -d ./universe_backend && unzip
universe_mdm_hpe_frontend_v6.10.2.zip -d ./universe_frontend
```

3. Скопируйте файлы frontend:

cp universe\_frontend/universe-frontend.war /opt/tomcat-9/webapps/

4. Скопируйте файлы backend и выдайте права с помощью одной команды:

```
cp -v universe_backend/universe-backend.war /opt/tomcat-9/webapps/ && ¥
cp -v universe_backend/Tomcat/bin/setenv.sh /opt/tomcat-9/bin/ && ¥
chmod +x /opt/tomcat-9/bin/* && ¥
cp -rv universe_backend/Tomcat/conf/universe /opt/tomcat-9/conf/ && ¥
cp -v universe_backend/Tomcat/libs/* /opt/tomcat-9/lib/ && ¥
chown -R tomcat:tomcat /opt/tomcat-9
```

5. Заполните /opt/tomcat-9/bin/setenv.sh, приведя файл к следующему виду:

```
vim /opt/tomcat-9/bin/setenv.sh
export JAVA_OPTS="${JAVA_OPTS}
-Dunidata.conf=${CATALINA_HOME}/conf/universe
-Dlogback.configurationFile=${CATALINA_HOME}/conf/universe/logback.xml"
#export JRE_HOME=/usr/Lib/jvm/java-8-oracle
#export JAVA_HOME=/usr/Lib/jvm/java-8-oracle
export POSTGRES_ADDRESS="localhost:5432"
export POSTGRES_USERNAME="postgres"
export POSTGRES_PASSWORD="notpostgres"
export DATABASE_NAME="universe"
export SEARCH_CLUSTER_NAME="universe-os-cluster"
export SEARCH_CLUSTER_ADDRESS="localhost:9200"
```

где: POSTGRES\_ADDRESS - адрес ноды PostgreSQL; POSTGRES\_USERNAME - имя пользователя PostgreSQL: POSTGRES\_PASSWORD - пароль пользователя PostgreSQL; DATABASE\_NAME - имя базы данных; SEARCH\_CLUSTER\_NAME - имя кластера Opensearch; SEARCH\_CLUSTER\_ADDRESS - перечисление участников кластера Opensearch.

6. Сохраните изменения:

:wq

7. Откройте для редактирования /opt/tomcat-9/conf/universe/backend.properties:

vim /opt/tomcat-9/conf/universe/backend.properties

8. Добавьте настройки пути хранения лога (в любое место файла в новой строке):

```
org.unidata.mdm.system.bitronix.tm.journal.disk.logPart1Filename=${CATAL
INA_HOME}/logs/btm1.tlog
org.unidata.mdm.system.bitronix.tm.journal.disk.logPart2Filename=${CATAL
INA_HOME}/logs/btm2.tlog
```

9. Запустите tomcat:

systemctl start tomcat

После запуска Тотсаt должно пройти некоторое время, после чего система Юниверс будет доступна по адресу *http://<your-server-ip>:8080/universe-frontend*.

После окончания установки выполните вход в систему.

# 1.9. Описание дистрибутива и контрольные суммы

# 1.9.1. Дистрибутив

Дистрибутив предоставляется клиенту при приобретении продукта через

менеджера компании "Юниверс Дата".

Внутри редакции дистрибутив делится на сборку для ручной установки и для Docker; на frontend и backend.

Шаблон имени файла дистрибутива: Universe [product] [edition] [version\_tag].

Примечание: Ниже приведены примеры описаний файлов для редакции Enterprise с использованием *"ее"* в имени.

# 1.9.1.1. Файлы frontend

Файл **universe\_mdm\_hpe\_frontend\_6.x.0.zip**. Используется для ручной установки системы. Содержимое:

- version.txt имя продукта, редакция и тег версии.
- universe-frontend.war пакет приложения Java, содержащий frontend-часть приложения и файлы конфигурации.
  - customer.json файл конфигурации frontend. Для изменения конфигурации, создавайте пользовательский customer.json и переопределяйте путь к нему.
  - react-resources/images/ каталог с изображениями системы (лого, фон на экране логина и т.д.).
- universe-frontend.sha256 контрольная сумма файла universe-frontend.war.

Файл **universe\_mdm\_hpe\_frontend\_docker\_6.x.0.zip**. Используется для установки системы через Docker. Содержимое:

- version.txt имя продукта, редакция и тег версии.
- universe\_mdm\_hpe\_frontend\_docker\_6.x.0.tar архив с набором образов для Docker.
- universe\_mdm\_hpe\_frontend\_docker\_6.x.0.sha256 контрольная сумма файла universe\_mdm\_hpe\_frontend\_docker\_6.x.0.tar.

# 1.9.1.2. Файлы backend

Файл **universe\_mdm\_hpe\_backend\_6.x.0.zip**. Используется для ручной установки системы. Содержимое:

- version.txt имя продукта, редакция и тег версии.
- Tomcat дополнительные файлы для Tomcat (после установки Tomcat эти файлы копируются на сервер в соответствующие каталоги).

- Tomcat/conf/universe/backend.properties основные параметры системы.
- Tomcat/conf/universe/logback.xml логи системы.
- Tomcat/conf/universe/enterprise-pipelines.json потоки выполнения системы.
- universe-backend.war пакет приложения Java, содержащий backend-часть приложения.
- universe-backend.sha256 контрольная сумма файла universe-backend.war.

Файл **universe\_mdm\_hpe\_backend\_docker\_6.x.0.zip**. Используется для установки системы через Docker. Содержимое:

- version.txt имя продукта, редакция и тег версии.
- universe\_mdm\_hpe\_backend\_docker\_6.x.0.tar архив с набором образов для Docker.
- universe\_mdm\_hpe\_backend\_docker\_6.x.0.sha256 контрольная сумма файла universe\_mdm\_hpe\_backend\_docker\_6.x.0.tar.

# 1.9.2. Контрольные суммы

Файлы контрольных сумм генерируются для каждого архива дистрибутива, а также для пакетов внутри дистрибутива. Например, если в поставку входит архив universe\_mdm\_hpe\_frontend\_6.x.0.zip, то вместе с ним идет файл universe\_mdm\_hpe\_frontend\_6.x.0.sha256. Внутри архива дополнительно контрольная сумма для файла universe-frontend.war.

Пример содержимого контрольной суммы: b1311d2ccf985e552b71019707144631e3672fe4789666b20cc8f67f369d3974 universe\_mdm\_hpe\_frontend\_6.x.0.zip

# 1.10. Настройка библиотеки Hunspell

Библиотека Hunspell может дополняться под нужды пользователя.

Словарь hunspell состоит из двух файлов:

- Первый файл (.*dic*) содержит список "корней" слов (стемов) со специальными метками, флагами.
- Второй файл (.aff) содержит группы аффиксов, обозначенные флагами.

Файлы можно редактировать, добавляя свои слова и правила.

Описание формата см. в официальной документации.

Пример: слово "процессный".

- 1. Откройте файл словаря *ru\_RU.dic*, кодировка KOI8-R.
- 2. Добавьте строку "процессный/А":
  - А прописывается латиницей. Это флаг, обозначающий к какой группе аффиксов (второй файл .aff) относится данное слово (Рисунок 1).
- 3. Сохраните файл в домашней директории OpenSearch в ./hunspell.
- 4. Перезапустите OpenSearch.

| 🔚 черновик | 2.txt 🗵 🔚 новый 2 🗵 🔚 новый 5 🗵 🔚 ru_RU.dic 🗵 |
|------------|---|
| 146247     | ёршик/К                                       |
| 146248     | ёрш   |
| 146249     | ёрзавший/А                                    |
| 146250     | ëpsatb/BLMP                                   |
| 146251     | ёрзание/Ј                                     |
| 146252     | ёрзающий/А                                    |
| 146253     | ёрнический/А                                  |
| 146254     | ёрничавший/А                                  |
| 146255     | ёрничать/BLMP                                 |
| 146256     | ёрничающий/А                                  |
| 146257     | ëphur/K                                       |
| 146258     | емче  |
| 146259     | ёмкость/N                                     |
| 146260     | ёмкостный/А                                   |
| 146261     | ёмкий/АЅ                                      |
| 146262     | ёлочный/А                                     |
| 146263     | ënouka/I                                      |
| 146264     | ёлка/I  |
| 146265     | ёкнувший/А                                    |
| 146266     | ëkhytb/LMR                                    |
| 146267     | ёкавший/А                                     |
| 146268     | ёкать/LMP                                     |
| 146269     | ёканье/Ј                                      |
| 146270     | ёкающий/А                                     |
| 146271     | процессный/А                                  |
|            |   |
|            |   |

Рисунок 1 – Пример добавления слова "процессный"

# 2. Обновление системы

Руководство содержит сведения по обновлению Юниверс Управление Мастер Данными Высоконагруженная Редакция (далее – Юниверс MDM HPE) с предыдущей релизной версии на текущую.

После каждого обновления системы следует переиндексировать все данные (записи, бизнес-процессы, связи, черновики, классификацию, таблицы сопоставления). Используйте <u>операцию reindexDataJob</u>.

#### Содержание:

| 2.1. | Обновление системы с помощью Docker          | 70 |
|------|--|----|
| 2.2. | Обновление системы вручную                   | 72 |
| 2.3. | Рекомендации по работе с БД                  | 89 |
| 2.4. | Рекомендации по работе с форматами даты      | 90 |
| 2.5. | Настройка в Tomcat редиректа с http на https | 91 |

# 2.1. Обновление системы с помощью Docker

### 2.1.1. Предварительные условия

- Сервер, на котором размещена обновляемая версия Юниверс MDM, имеет доступ к интернету.
- Установлен Git.
- Установлен Docker Compose (версия выше 1.29).
- Проект Юниверс МDM был клонирован на сервер.

# 2.1.2. Обновление системы

#### Предупреждение:

В версии 6.9 был осуществлен переход с Elasticsearch на OpenSearch. Если обновлять ветку, имея поднятый docker, то старый контейнер elasticsearch будет препятствовать дальнейшим запускам. Поэтому рекомендуется выполнить нижеописанные действия

- 1. Перед обновлением остановите работу образа Docker Compose через команду docker-compose down.
  - Поднять контейнеры после обновления возможно через команду: docker-compose up --force-recreate -d --remove-orphans. Последняя опция очищает Elasticsearch контейнер.

- 2. Войдите в репозиторий Gitlab (доступ к репозиторию с дистрибутивом вашего продукта должен быть получен заранее согласно договору поставки).
- 3. Переключитесь на ветку release/6.11.
- 4. Извлеките проект. Команда git pull.
- 5. Выполните миграцию на Opensearch с потерей или сохранением данных (см. инструкцию ниже).
- 6. Перейдите в каталог с кодом. Пример команды:

```
cd universe-platform-deploy
```

- 7. При необходимости в файле .*env*, расположенном в корне проекта, измените параметры для БД и параметры портов. Пример файла .*env* смотрите по <u>ссылке</u>.
- 8. Для запуска системы используйте команду:

docker-compose up -d

## 2.1.2.1. Переход на OpenSearch с потерей данных

1. Поскольку данные не будут перенесены, выполните операцию <u>переиндексации данных</u>, а при наличии правил качества - операцию <u>переприменения данных</u>.

#### Важно:

• Если в качестве хранилища для аудита выбран поисковой движок, журнал событий будет утерян — его невозможно восстановить из БД, даже если она также указана в качестве хранилища.

#### 2.1.2.2. Переход на OpenSearch с переносом данных

- B docker-compose конфигурации контейнера Opensearch укажите существующий volume с данными Elasticsearch mdm-elasticsearch-data:/usr/share/opensearch/data.
- 2. В случае если необходимо избавиться от упоминания Elastic, то можно оставить volume по умолчанию (mdm-opensearch-data), выполнив одно из двух действий:
  - Клонировать volume. Пример:

```
docker run --rm ¥
   -v mdm-elasticsearch-data:/original ¥
   -v mdm-opensearch-data:/migration ¥
   ubuntu:latest ¥
   bash -c "cp -R /original/* /migration/"
```

• Переименовать volume:

```
mv /var/lib/docker/volumes/mdm-elasticsearch-data
/var/lib/docker/volumes/mdm-opensearch-data
```

# 2.2. Обновление системы вручную

#### Предупреждение:

В версии 6.9 был осуществлен переход с Elasticsearch на OpenSearch. В связи с этим в инструкции по обновлению системы добавились дополнительные действия по переходу на Opensearch и переносу данных, после которых следует обновление системы

#### Содержание:

| 2.2.1. | Переход на Opensearch вручную | 72 |
|--------|-------------------------------|----|
| 2.2.2. | Подготовка к обновлению       | 77 |
| 2.2.3. | Процедура обновления ПО       | 81 |

# 2.2.1. Переход на Opensearch вручную

#### 2.2.1.1. Переход с потерей данных

- 1. Установите Opensearch согласно инструкции по установке.
- 2. Перенесите настройки из файла elasticsearch.yml в opensearch.yml.
- 3. При необходимости обновите параметры MDM: org.unidata.mdm.search.cluster.nodes, org.unidata.mdm.search.cluster.address.
- Поскольку данные не будут перенесены, выполните операцию переиндексации данных, а при наличии правил качества - операцию переприменения данных.

#### Важно:

• Если в качестве хранилища для аудита выбран поисковой движок, журнал событий будет утерян — его невозможно восстановить из БД, даже если она также указана в качестве хранилища.
#### 2.2.1.2. Переход с переносом данных

Перед обновлением системы Юниверс необходимо сделать слепок индексов Elasticsearch. Рекомендуется выполнить этот шаг, чтобы иметь бэкап состояния поискового узла. Подробности см. в <u>официальной документации Elaticsearch</u>.

#### 2.2.1.2.1. Слепок индексов

- 1. Снимите слепок Elasticsearch согласно инструкции "<u>Резервное копирование</u> поискового индекса".
- 2. Далее выполните обновление согласно инструкции "Переход с потерей данных" (см. выше).
- 3. Используйте слепок для восстановления данных в opensearch. Укажите путь к хранилищу в *opensearch.yml*:

path.repo: /mount/backups

4. Добавьте репозиторий в конфигурацию кластера. Пример выполнения из консоли с помощью утилиты *curl*:

```
$ curl -XPUT 'localhost:9200/_snapshot/universe_indices_backup' -d
'{ "type" : "fs", "settings" : { "compress" : true, "location": "<путь к
директории репозитория из настроек>/universe_indices_backup" } }'
```

• Если путь к директории репозитория был указан в path.repo, его не требуется включать в вышеуказанную команду. В таком случае команда будет иметь вид:

```
curl -XPUT 'localhost:9200/_snapshot/universe_indices_backup' -d '{ "type" :
    "fs", "settings" : { "compress" : true, "location" :
    "universe_indices_backup" }}'
```

 Проверьте успешность выполнения предыдущего шага (в результате выполнения команды на экране должны отобразится текущие настройки репозитория):

```
$ curl -XGET 'localhost:9200/_snapshot/universe_indices_backup'
```

6. Закройте текущие индексы:

```
$ curl -XPOST 'localhost:9200/_all/_close'
```

Количество частей (Shards) индекса в снапшоте и в кластере должны совпадать, в противном случае восстановление завершится с ошибкой. Требование не распространяется на пустые кластеры.

7. В случае, если имя текущих снапшотов неизвестно, используйте команду:

\$ curl -XGET 'localhost:9200/\_snapshot/\_all'

8. Восстановите снапшот. Пример команды для снапшота с именем *snapshot\_1*:

```
$ curl -XPOST
'localhost:9200/_snapshot/universe_indices_backup/snapshot_1/_restore'
```

#### Примечания:

Представленный вариант применим в т.ч, если Elasticsearch развернут в docker — достаточно пробросить локальное хранилище и конфигурационный файл через docker volume.

Фрагмент docker-compose для Opensearch:

```
volumes:
- mdm-opensearch-data:/usr/share/opensearch/data
- ./hunspell:/usr/share/opensearch/config/hunspell/
- ./opensearch.yml:/usr/share/opensearch/config/opensearch.yml
- ./backups:/mount/backups
```

## 2.2.1.2.2. Миграция кластера

Представленный вариант подразумевает ручной перенос настроек и данных для каждого узла.

#### Доступно два варианта:

- Остановить кластер, выполнить перенос для каждого узла и запустить весь кластер (cluster restart upgrade);
- Проделать процесс для каждого узла отдельно в этом случае поисковый движок будет активен во время обновления (rolling upgrade).
- 1. Установите Opensearch согласно <u>инструкции по установке</u>, при этом директории Elasticsearch не должны быть затерты.
- 2. Отключите аллоцирование шардов:

```
curl -X PUT "localhost:9200/_cluster/settings" -H 'Content-Type:
application/json' -d'
{
    "persistent": {
        "cluster.routing.allocation.enable": "primaries"
    }
}
```

3. Остановите Elasticsearch на одном узле (rolling upgrade) или на всех (cluster restart upgrade):

sudo systemctl stop elasticsearch.service

- 4. Выполните переход на одном узле (rolling) или на всех (cluster restart):
  - Скопируйте данные и логи Elasticsearch (содержимое директорий data и logs) в аналогичные директории Opensearch. Например, sudo ср -r /usr/share/elasticsearch/data /usr/share/opensearch/data && chown -R opensearch:opensearch
  - Портируйте настройки из elasticsearch.yml в opensearch.yml.
- 5. Запустите OpenSearch на одном узле (rolling) или на всех (cluster restart):

```
sudo systemctl start opensearch
```

- 6. Rolling повторите шаги 3–5 до полного перехода всего кластера на OpenSearch.
- 7. Включите аллоцирование шардов обратно:

```
curl -X PUT "localhost:9200/_cluster/settings" -H 'Content-Type:
application/json' -d'
{
   "persistent": {
      "cluster.routing.allocation.enable": "all"
   }
}
```

# Шаг 3 может быть выполнен автоматически через утилиту opensearch-upgrade:

- 1. Установите Opensearch согласно инструкции по установке.
- 2. Отключите аллоцирование шардов:

```
curl -X PUT "localhost:9200/_cluster/settings" -H 'Content-Type:
application/json' -d'
{
    "persistent": {
        "cluster.routing.allocation.enable": "primaries"
    }
}
```

- 3. Проверьте, что заданы переменные окружения: ES\_HOME, ES\_PATH\_CONF, OPENSEARCH\_HOME, OPENSEARCH\_PATH\_CONF.
- 4. Исполните утилиту от имени пользователя, запустившего Elasticsearch:

```
/usr/share/opensearch/bin/opensearch-upgrade
```

5. Остановите Elasticsearch на узле:

```
sudo systemctl stop elasticsearch.service
```

6. Запустите Opensearch на узле:

sudo systemctl start opensearch

- 7. Повторите шаги 3-6 до полного перехода на Opensearch.
- 8. Включите аллоцирование шардов обратно:

```
curl -X PUT "localhost:9200/_cluster/settings" -H 'Content-Type:
application/json' -d'
{
   "persistent": {
      "cluster.routing.allocation.enable": "all"
   }
}
```

#### 2.2.1.2.3. Сохранение данных аудита

Для сохранения данных аудита при переходе на Opensearch проделайте следующие действия:

- 1. Остановите приложение, если оно запущено, чтобы в данные не вносились изменения.
- 2. Coздайте индекс default\_default\_audit read-only с помощью запроса:

```
PUT
http://OPEN_SEARCH_HOST:OPEN_SEARCH_PORT/default_default_audit/_settings
{
    "settings": {
        "index.blocks.write": true
    }
}
```

 Клонируйте индекс default\_default\_audit с данными во временный индекс default\_default\_audit\_old\_index\_copy с помощью запроса:

```
PUT
http://OPEN_SEARCH_HOST:OPEN_SEARCH_PORT/default_default_audit/_clone/de
fault_default_audit_old_index_copy/
```

- 4. Проверьте, что индекс default\_default\_audit\_old\_index\_copy создался и содержит все нужные данные.
- 5. Удалите индекс default\_default\_audit с помощью запроса:

DELETE http://OPEN\_SEARCH\_HOST:OPEN\_SEARCH\_PORT/default\_default\_audit

- 6. Запустите сервер с приложением и дождитесь выполнения запуска. Должен создаться новый индекс default\_default\_audit.
- 7. Копируйте данные из копии индекса default\_default\_audit\_old\_index\_copy в

новый индекс default\_default\_audit.

- Если данных в журнале много, то копирование может занять значительное время. Данные будут постепенно переноситься в журнал, работа в системе в это время будет доступна.
- 8. Выполните запрос:

```
POST http://OPEN_SEARCH_HOST:OPEN_SEARCH_PORT/_reindex/
```

• В теле запроса должен присутствовать следующий JSON:

```
{
    "source": {
        "index": "default_default_audit_old_index_copy"
    },
    "dest": {
        "index": "default_default_audit"
    }
}
```

- 9. Проверьте, что данные перенеслись в индекс default\_default\_audit.
- 10. Зайдите в раздел "Журнал" и проверьте наличие в журнале записей до обновления.
- 11. Опционально: Если шаги выполнены корректно, то можно удалить копию индекса default\_default\_audit\_old\_index\_copy с помощью запроса:

```
DELETE
http://OPEN_SEARCH_HOST:OPEN_SEARCH_PORT/default_default_audit_old_index
_copy
```

#### 2.2.2. Подготовка к обновлению

#### Примечание:

Перед началом резервного копирования экспортируйте модель данных и остановите сервер приложений Tomcat

#### 2.2.2.1. Резервное копирование компонентов

2.2.2.1.1. Резервное копирование базы данных

Резервное копирование базы данных выполняется стандартными средствами СУБД (pg\_dump, pg\_restore, pg admin и т.п.). Пример создания резервной копии с помощью утилиты pg dump:

1. Остановите сервер Tomcat:

systemctl stop tomcat

- Уточните имя базы данных в файле настроек <TOMCAT\_HOME>/conf/Catalina/localhost/universe-backend.xml. Имя содержится в строке вида (в примере указано имя universe): Resource .... url="jdbc:postgresql://localhost:5432/universe" .../>
- 3. Выполните команду:

pg\_dump -U \${POSTGRES\_USERNAME} -h \${POSTGRES\_ADDRESS} -Fc \${DATABASE\_NAME} > ./\${DATABASE\_NAME}\$(date +%F\_%H%M).dump

• Узнать путь к БД можно в файле ./bin/setenv.sh или с помощью команды:

echo \$POSTGRES\_ADDRESS\$DATABASE\_NAME

4. Запустите сервер Tomcat:

systemctl start tomcat

Параметры команды могут отличаться. Необходимо убедиться, что в резервную копию были включены все схемы базы universe.

2.2.2.1.2. Резервное копирование поискового индекса

Резервное копирование поискового индекса осуществляется стандартными средствами Opensearch. Подробнее см. в <u>официальной документации</u> (подходит и для Opensearch). Перед выполнением операции необходимо изменить настройки Opensearch в соответствии с конфигурацией целевой системы.

#### Для создания резервной копии поискового индекса:

 Укажите путь к репозиторию снапшотов в секции Paths (файла <OPENSEARCH\_HOME>/config/opensearch.yml). Путь должен быть доступен всем узлам кластера:

path.repo : /mount/backups

2. Перезапустите Opensearch. Пример команды:

systemctl restart opensearch

3. Создайте каталог для репозитория:

mkdir -p /mount/backups

4. Перейдите в созданный каталог:

cd /mount/

5. Назначьте права на запись всем пользователям:

chmod a+w ./backups

- 6. Перезапустите узлы кластера, если используется кластерная конфигурация (в противном случае шаг можно пропустить).
- 7. Добавьте репозиторий в конфигурацию кластера. Пример выполнения из консоли с помощью утилиты curl:

```
$ curl -XPUT 'localhost:9200/_snapshot/universe_indices_backup' -d '{ "type" :
"fs", "settings" : { "compress" : true, "location": "<путь к директории репозитория
из настроек>/universe_indices_backup" } }'
```

Если путь к директории репозитория был указан в path.repo, его не требуется включать в вышеуказанную команду. В таком случае команда будет иметь вид:

```
curl -XPUT 'localhost:9200/_snapshot/universe_indices_backup' -d '{ "type" :
"fs", "settings" : { "compress" : true, "location" : "universe_indices_backup" }}'
```

8. Проверьте успешность выполнения предыдущего шага (в результате выполнения команды на экране должны отобразится текущие настройки репозитория):

```
$ curl -XGET 'localhost:9200/_snapshot/universe_indices_backup'
```

9. Создайте снапшот всех активных индексов кластера (в примере используется имя snapshot\_1):

```
$ curl -XPUT
'localhost:9200/_snapshot/universe_indices_backup/snapshot_1?wait_for_co
mpletion=true'
```

В случае успешного завершения процедуры создания снапшота на экране отобразится сообщение вида "state": "SUCCESS".

10. Для получения дополнительных сведений о состоянии снапшота выполните команду:

\$ curl -XGET 'localhost:9200/\_snapshot/universe\_indices\_backup/snapshot\_1'

2.2.2.1.3. Резервное копирование файлов конфигурации

- 1. Скопируйте содержимое:
- Каталога <TOMCAT HOME>/webapps;
- Каталога <TOMCAT\_HOME>/universe-integration;
- Каталога, указанного в параметре -Duniverse.conf файла конфигурации tomcat.conf.

Примечание:

По умолчанию параметр -Duniverse.conf расположен в файле <TOMCAT\_HOME>/conf/tomcat.conf

2. Укажите путь к файлу logback.xml в <TOMCAT\_HOME>/conf/tomcat.conf:

Dlogback.configurationFile=/opt/conf/universe/logback.xml в настройках \$TOMCAT\_HOME/conf/tomcat.conf

Если в файл universe-backend.xml, расположенный в <TOMCAT\_HOME>/ conf/Catalina/localhost, вносились изменения - создайте его резервную копию для последующего <u>обновления</u>.

#### 2.2.2.2. Восстановление резервных копий

2.2.2.2.1. Восстановление резервной копии базы данных Восстановление резервной копии БД осуществляется стандартными средствами СУБД (pg\_restore, pg admin и т.д.). Далее представлен пример восстановления резервной копии средствами pg\_restore.

1. Остановите сервер Tomcat:

```
systemctl stop tomcat
```

2. Создайте пустую БД. Для этого используйте команды консоли:

```
su postgres
psql -U postgres
create database <имя базы>;
¥q
```

Либо используйте SQL-запрос:

```
CREATE DATABASE <имя базы> WITH OWNER = postgres
```

В качестве альтернативы может быть выполнена очистка существующей БД.

3. Выполните команду:

```
$ pg_restore -d universe -p 5432 -h localhost -U <username> -- jobs=<количество
параллельных потоков, каждый занимающий одно соединение + 1
административное соединение> <путь и имя файла>
```

4. Запустите сервер Tomcat:

systemctl start tomcat

2.2.2.2.2. Восстановление резервной копии поискового индекса

Восстановление резервных копий поискового индекса осуществляется

стандартными средствами Opensearch. Подробнее см. в <u>официальной</u>

документации (подходит и для Opensearch).

#### Для восстановления резервной копии поискового индекса:

1. Закройте текущие индексы:

\$ curl -XPOST 'localhost:9200/\_all/\_close'

Количество частей (Shards) индекса в снапшоте и в кластере должны совпадать, в противном случае восстановление завершится с ошибкой. Требование не распространяется на пустые кластеры.

2. В случае, если имя текущих снапшотов неизвестно, используйте команду:

```
$ curl -XGET 'localhost:9200/_snapshot/_all'
```

3. Восстановите снапшот. Пример команды для снапшота с именем snapshot\_1:

```
$ curl -XPOST
'localhost:9200/_snapshot/universe_indices_backup/snapshot_1/_restore'
```

Снапшот представляет собой слепок состояния поискового индекса. Таких снапшотов может быть несколько – для хранения нескольких состояний Opensearch.

#### 2.2.3. Процедура обновления ПО

#### 2.2.3.1. PostgreSQL

#### Примечание:

Ниже приведен пример обновления PostgreSQL с версии 9.6 на версию 13.2 и примеры команд для OC CentOS 7. Если ваше окружение отличается, то используйте официальную документацию PostgreSQL

#### Совет:

Рекомендуется использовать программы pg\_dump и pg\_dumpall из более новой версии PostgreSQL

Настоящая инструкция предполагает, что текущая установка находится в каталоге

/usr/local/pgsql, а область данных – в /usr/local/pgsql/data. Замените пути соответствующим образом.

Пример обновления PostgreSQL с помощью pg\_dumpall:

1. Создайте резервную копию базы данных (шаг можно пропустить, если резервная копия была сделана ранее):

```
pg_dumpall > outputfile.sql
```

2. Создайте резервную копию файлов конфигурации pg\_hba.conf и postgresql.conf из <<u>POSTGRESQL9.6\_HOME>/data</u>.

3. Остановите postgresql-9.6:

systemctl stop postgresql-9.6

4. Переименуйте каталог, если postgresql-9.6 был установлен в каталоге по умолчанию:

mv /usr/local/pgsql /usr/local/pgsql.old

5. Установка Postgresql-13 будет производиться из исходного кода. Установите git:

yum install git

6. Скачайте исходный код из официального репозитория (каталог postgres создается автоматически в активном каталоге):

git clone https://github.com/postgres/postgres

7. Перейдите в созданный каталог postgres и переключитесь на ветку репозитория с версией 13:

```
cd postgres/
git checkout REL_13_STABLE
```

8. Для обеспечения корректности конфигурирования и сборки установите следующее:

```
yum install readline-devel zlib-devel flex bison uuid-devel libxml2
libxml2-devel systemd systemd-devel
```

9. Запустите конфигурирование:

./configure --with-uuid=ossp --with-libxml --with-systemd

- Дополнительные параметры --with-uuid=ossp и --with-libxml нужны для установки необходимых в дальнейшем расширений, параметр --with-systemd необходим для управления запуском Postgresql с помощью systemctl (описано ниже в п. 23).
- 10. Запустите сборку: make
- 11. Под учетной записью пользователя запустите автотесты (данный функционал не работает под учетной записью root): make check
- 12. Запустите установку: sudo make install
- 13. Создайте каталог, где будут храниться файлы БД:

mkdir /usr/local/pgsql/data

14. Назначьте права пользователю на этот каталог:

chown postgres /usr/local/pgsql/data/

15. Перейдите в каталог:

cd /usr/local/pgsql/bin

16. Выполните следующее:

export PATH=/usr/local/pgsql/bin/:\$PATH
export PGDATA=/usr/local/pgsql/data/

17. Под учетной записью пользователя инициализируйте БД:

```
initdb -k -D /usr/local/pgsql/data/
```

- 18. Переместите в <POSTGRESQL13\_HOME>/data сохраненные файлы конфигурации pg\_hba.conf и postgresql.conf.
- 19. Запустите postgresql13:

pg\_ctl start

20. Установите расширения для БД:

```
a. cd .../postgres/contrib
b. make
c. make install
```

21. Восстановите данные из созданной резервной копии (пример восстановления с помощью <u>pg\_restore</u>):

```
/usr/local/pgsql/bin/psql -d postgres -f outputfile.sql
```

22. Остановите postgresql13:

pg\_ctl stop

23. Для автозапуска и управления работой приложения с помощью команд systemctl добавьте следующий системный файл

/etc/systemd/system/postgresql-13.service с содержимым:

```
[Unit]
Description=PostgreSQL database server
Documentation=man:postgres(1)
[Service]
Type=notify
User=postgres
ExecStart=/usr/local/pgsql/bin/postgres -D /usr/local/pgsql/data
ExecReload=/bin/kill -HUP $MAINPID
KillMode=mixed
KillSignal=SIGINT
TimeoutSec=0
[Install]
WantedBy=multi-user.target
```

24. Выполните следующее:

```
systemctl enable postgresql-13 && systemctl start postgresql-13
```

25. Проверьте статус:

systemctl status postgresql-13

В случае ошибки запуска (статус failed) перезагрузите операционную систему.

26. Удалите postgresql-9.6 из автозагрузки:

systemctl disable postgresql-9.6

27. При необходимости удалите postgresql-9.6:

```
yum remove pgdg-centos96 postgresql96 postgresql96-server postgresql96-libs
postgresql96-devel postgresql96-contrib postgresql96-jdbc
```

28. Проверьте версию установленного PostgreSQL:

postgres --version

Далее запустите миграцию БД от 9.6 к 13:

1. Остановите сервис:

systemctl stop postgresql-9.6

2. Проверьте возможность обновления:

```
su postgres
cd /var/lib/pgsql
/usr/pgsql-13/bin/pg_upgrade --old-bindir=/usr/pgsql-9.6/bin/ --
new-bindir=/usr/pgsql-13/bin/ --old- datadir=/var/lib/pgsql/9.6/data/
--new- datadir=/var/lib/pgsql/13/data/ --check
```

3. Если в ответном сообщении отображено *Clusters are compatible*, то запустите миграцию:

```
su postgres
cd /var/lib/pgsql
/usr/pgsql-13/bin/pg_upgrade --old-bindir=/usr/pgsql-9.6/bin/ --
new-bindir=/usr/pgsql-13/bin/ --old- datadir=/var/lib/pgsql/9.6/data/
--new- datadir=/var/lib/pgsql/13/data/
```

4. После успешного завершения версия 9.6 может быть удалена:

```
yum remove pgdg-centos96 postgresql96 postgresql96-server postgresql96-libs
postgresql96-devel postgresql96-contrib postgresql96-jdbc postgresql-jdbc
```

5. Повторно проверьте версию установленного PostgreSQL:

```
yum list installed | egrep "postgres|pgdg"
```

#### 2.2.3.2. Файлы приложения и конфигурации

Предварительно: убедитесь, что созданы все требуемые резервные копии

<u>конфигурации</u>.

Примечание:

При копировании новых библиотек необходимо удалять старые, т.к в некоторых случаях при обновлении системы используется не новая, а старая версия библиотеки, вследствие чего в работе системы возникают ошибки

- 1. Удалите все файлы из директорий:
  - <TOMCAT\_HOME>/work;
  - <TOMCAT\_HOME>/temp;
  - <TOMCAT\_HOME>/webapps;
- 2. Обновите конфигурационные файлы:
- Скопируйте из <TOMCAT\_HOME>/conf/Catalina/localhost дистрибутива в директорию <UNIVERSE\_CONF\_DIR> ранее созданную резервную копию файла universe-backend.xml. Пример команды:

```
cp <UNIVERSE_PACKAGE_DIR>/conf/Catalina/localhost/universe-backend.xml
<TOMCAT_HOME>/conf/Catalina/localhost/
```

 Скопируйте из <TOMCAT\_HOME>/conf/universe/ дистрибутива в директорию <UNIVERSE\_CONF\_DIR>/ ранее созданную резервную копию файла backend.properties. Пример команды:

```
cp <UNIVERSE_PACKAGE_DIR>/conf/universe/backend.properties
<TOMCAT_HOME>/conf/universe/
```

• В файле backend.properties закомментируйте параметр universe.licensing.gpg.license.file для корректного обновления лицензии через интерфейс системы. После запуска обновленной системы необходимо активировать действующую <u>лицензию</u> повторно.

• Скопируйте из дистрибутива в директорию <TOMCAT\_HOME>/conf/universe/ актуальный файл external-routing.xml, либо внесите в него актуальные изменения:

cp <UNIVERSE\_PACKAGE\_DIR>/conf/universe/external-routing.xml <TOMCAT\_HOME>/conf/universe/

- При необходимости восстановите из резервных копий файлов персональные параметры конфигурации.
- Скопируйте содержимое директории <TOMCAT\_HOME>/webapps/ и <TOMCAT\_HOME>/lib/ (WAR-файлы приложения, библиотеки) из установочного пакета при помощи команд:

```
cp <UNIVERSE_PACKAGE>/Tomcat/lib/* <TOMCAT_HOME>/lib/
cp <UNIVERSE_PACKAGE>/Tomcat/webapps/universe-*.war <TOMCAT_HOME>/webapps/
```

- 4. Убедитесь, что все требуемые для проекта файлы интеграции в <TOMCAT\_HOME>/universe\_integration актуальны. При необходимости воспользуйтесь резервной копией каталога, созданной ранее.
- 5. При переходе на новую версию PostgreSQL удалите устаревшие стандартные .lib файлы. Пример файла: postgresql-13-1200.jdbc41.lib.
- 6. Создайте резервные копии конфигурационных файлов (если они изменялись например, изменения могли быть внесены в файл

<TOMCAT\_HOME>/webapps/universe-frontend/customer.json), которые

расположены по адресам:

- <TOMCAT\_HOME>/webapps/universe-backend;
- <TOMCAT\_HOME>/webapps/universe-frontend.
- 7. Удалите индексы Elasticsearch:

\$ curl -XDELETE 'http://localhost:9200/\_all'

8. Перезапустите Tomcat:

systemctl restart tomcat

Примечание:

Процедуру обновления программных компонентов необходимо выполнить на каждом узле кластера, если Юниверс MDM установлена в кластерной конфигурации

#### 2.2.3.3. Apache Tomcat

#### Для обновления Tomcat с версии 7 до версии 9:

1. Остановите Tomcat 7 и удалите из автозагрузки:

systemctl stop tomcat && systemctl disable tomcat

2. Создайте системного пользователя:

useradd -m -U -d /opt/tomcat -s /bin/false tomcat

3. Скачайте и распакуйте архив (актуальную версию можно найти на <u>официальном сайте</u>). Пример команды (9.*х* - актуальная версия Tomcat):

```
wget
https://downloads.apache.org/tomcat/tomcat9/v9.x/bin/apache-tomcat-9.x.t
ar.gz
tar -zxpvf apache-tomcat-9.x.tar.gz -C /opt/(или другой необходимый каталог)
```

4. Переместите в каталог /opt/tomcat (или другой необходимый):

mv apache-tomcat-9.x tomcat-9

5. Измените владельца каталога на tomcat:

chown -R tomcat: /opt/tomcat-9

6. Создайте файл системного модуля:

nano /etc/systemd/system/tomcat.service

7. Запишите в файл следующее:

```
[Unit]
Description=Tomcat 9 servlet container
After=network.target
[Service]
Type=forking
User=tomcat
Group=tomcat
Environment="JAVA_HOME=/usr/lib/jvm/jre"
Environment="JAVA OPTS=-Djava.security.egd=file:///dev/urandom"
Environment="CATALINA BASE=/opt/tomcat-9"
Environment="CATALINA HOME=/opt/tomcat-9"
Environment="CATALINA PID=/opt/tomcat-9/temp/tomcat.pid"
Environment="CATALINA OPTS=-Xms512M -Xmx1024M -server - XX:+UseParallelGC"
ExecStart=/opt/tomcat-9/bin/startup.sh
ExecStop=/opt/tomcat-9/bin/shutdown.sh
[Install]
WantedBy=multi-user.target
```

8. Выполните перезагрузку для обновления конфигураций:

systemctl daemon-reload

- 9. Исправьте системные переменные. CATALINA\_BASE=директория Tomcat 9 (экземпляры Tomcat), CATALINA\_HOME=директория tomcat 9 (общие файлы Tomcat).
- 10. Перенесите содержимое каталогов из Tomcat 7 в Tomcat 9:
  - TOMCAT7\_HOME/conf/Catalina -> TOMCAT9\_HOME/conf/Catalina
  - TOMCAT7\_HOME/conf/universe -> TOMCAT9\_HOME/conf/universe
  - TOMCAT7\_HOME/bin/setenv.bat(sh) -> TOMCAT9\_HOME/bin/setenv.bat(sh)
  - TOMCAT7\_HOME/lib/postgresql-x.x.x -> TOMCAT9\_HOME/lib/postgresqlx.x.x
- Пример команды:

cp <TOMCAT9\_HOME>/conf/universe <TOMCAT7\_HOME>/conf/universe/

- 11. В файле <TOMCAT9\_HOME>/conf/tomcat.conf укажите пути каталогов Tomcat 9 для: CATALINA\_HOME, CATALINA\_TMPDIR, Duniverse.conf, Dlogback.configurationFile.
- 12. Перенесите дополнительные .jar файлы, добавленные в проект, если такие имеются:
  - TOMCAT7\_HOME/lib/\*.jar -> TOMCAT9\_HOME/lib/
  - TOMCAT7\_HOME/webapps/\*.war -> TOMCAT9\_HOME/webapps/
  - TOMCAT7\_HOME/universe-integration -> TOMCAT9\_HOME/universe-integration
- Пример команды:

cp <TOMCAT9\_HOME>/lib/\*.jar <TOMCAT7\_HOME>/lib

13. Сделайте сценарии внутри bin исполняемыми:

sh -c 'chmod +x /opt/tomcat/latest/bin/\*.sh'

14. Добавьте Тотсат 9 в автозагрузку и запустите:

systemctl enable tomcat && systemctl start tomcat

15. Проверьте статус:

systemctl status tomcat

#### 2.2.3.4. База данных

#### Обновление базы данных:

1. Скопируйте директорию database из дистрибутива в любой подкаталог

относительно конечного каталога установки дистрибутива Юниверс.

- 2. Укажите требуемые параметры системы в файле init\_env.sh (init\_env.bat для Windows) из директории database.
- 3. Запустите скрипт обновления схемы данных в БД:

```
./update_database.sh
```

- В случае возникновения проблем с обновлением используйте следующие скрипты:
  - Для просмотра статуса миграций: ./check\_database.sh (.bat для Windows).
  - Для устранения ошибок миграций: ./repair\_database.sh (.bat для Windows).

Если проблема coxpansetcs после выполнения repair\_database.sh, то необходимо обратиться в техподдержку.

#### 2.2.3.5. Поисковой индекс

#### Примечание:

В версии 6.9 был осуществлен переход с Elasticsearch на OpenSearch. Инструкцию по переходу см. <u>по ссылке</u>

# 2.3. Рекомендации по работе с БД

При обновлении параметров *host, port, dbname, username, password* базы данных необходимо вносить исправления:

- в таблицу org\_unidata\_mdm\_data.nodes\_info в БД.
- в файле конфигурации .env.

Это актуально для таких случаев, как перенос БД на другой сервер, переименования БД или изменение логина/пароля к БД.

В файле .env/setenv.sh параметры переменных, взятых из файла backend.properties.

Названия параметров для 5.х:

```
POSTGRES_USERNAME="postgres" - пользователь БД PostgreSQL.
POSTGRES_ADDRESS="localhost:5432" - адрес сервера PostgreSQL.
POSTGRES_PASSWORD="notpostgres" - пароль пользователя PostgreSQL.
DATABASE_NAME="universe" - имя базы данных.
```

```
Названия параметров для 6.х:
```

MDM\_POSTGRES\_USER=postgres - пользователь БД PostgreSQL. MDM\_POSTGRES\_PASSWORD=postgres - пароль пользователя PostgreSQL. MDM\_POSTGRES\_DB\_NAME=postgres - имя базы данных. POSTGRES\_OUTER\_PORT=15431 - внешний порт

# 2.4. Рекомендации по работе с форматами даты

#### Примечание:

Рекомендуется выбирать формат даты до начала наполнения системы данными. Если формат необходимо исправить, когда в системе уже есть данные, то такое изменение будет доступно только посредством вмешательства в базу данных.

В Юниверс MDM 6.х принимаются периоды актуальности в форматах:

- yyyy-MM-dd'T'HH:mm:ss.SSS
- yyyy-MM-dd'T'HH:mm:ss.SSS'Z'
- yyyy-MM-dd'T'HH:mm:ss.SSS+00:00

В Юниверс MDM 5.х используется только формат ISO 8601 (уууу-MM-dd'T'HH:mm:ss.SSS).

Формат даты задается в файле backend.properties.

Имя параметров Юниверс MDM 6.х:

- org.unidata.mdm.data.validity.period.start
- org.unidata.mdm.data.validity.period.end

Имя параметров Юниверс MDM 5.х:

- unidata.validity.period.start
- unidata.validity.period.end

Тредупреждение

Параметры ...start и ...end должны иметь одинаковый формат даты, иначе система не запустится

#### 2.4.1. Переход с 5.х на 6.х

При миграции с версий Юниверс MDM 5.х на Юниверс MDM 6.х необходимо выполнять несколько условий:

- Форматы даты должны быть одинаковы для обеих систем.
- Чтобы не было смещения даты и разбиения индексов, необходимо, чтобы в 5.х и в 6.х версиях совпадали числа периодов, заданные в backend.properties, например:

- 1900-01-01T00:00:00.000 start
- 9999-12-31T23:59:59.999 end

Примечание:

Для MDM 6.х рекомендуемый правый конец даты: 2500 год. Если для миграции используется другой период, то после миграции необходимо установить лимит в конец 2500 года (2500-12-31T23:59:59.999Z)

# 2.5. Настройка в Tomcat редиректа с http на https

Настройка редиректа с http на https может понадобиться в случае, если сервера Юниверс MDM изолированы, и открыт только порт 443 (стандартный для https). При такой конфигурации Tomcat не может принимать http-запросы.

Ниже приведен пример конфигурации, в котором в server.xml открывается порт 80 (стандартный для http) с редиректом на 443, а в настройках web.xml добавляется конфигурация, чтобы любой запрос обрабатывался как CONFIDENTIAL. При такой настройке любой запрос к backend будет переадресован на 443 порт с https.

Добавьте коннекторы в файл server.xml:

```
<Connector port="80" protocol="HTTP/1.1"

connectionTimeout="20000" redirectPort="443"/>

<Connector port="443" protocol="org.apache.coyote.http11.Http11NioProtocol"

maxThreads="150" scheme="https" secure="true" SSLEnabled="true" >

<SSLHostConfig>

<Certificate certificateKeystoreFile="${path_to_keystore}"

certificateKeystorePassword="${keystore_password}"

type="RSA" />

</SSLHostConfig>

</Connector>
```

Добавьте в конец файла web.xml секцию:

Добавьте в файл **docker-compose.yaml** новые пути в секцию volumes. Эти пути позволяют подложить файлы **server.xml** и **web.xml** в Tomcat:

#### volumes:

\${BACKEND\_INTEGRATION:-./universe-integration}:/usr/local/tomcat/universe-int
egration

PATH\_TO\_MODIFIED\_SERVER.xml/server.xml:/usr/local/tomcat/conf/server.xmlPATH\_TO\_MODIFIED\_WEB.xml/web.xml:/usr/local/tomcat/conf/web.xml

# 3. Системные требования

# 3.1. Операционная система

#### Рекомендуемые ОС для сервера системы:

- 1. Ubuntu Server 20.04 LTS. *Рекомендуемая*. Совместимость регулярно тестируется для MDM.
- 2. ALT Server 10.1 (Mendelevium). *Рекомендуемая для ФСТЭК-версии*. Совместимость регулярно тестируется для MDM.
- 3. Astra Linux Special Edition (релиз "Смоленск"). *Рекомендуемая для ФСТЭК-версии*. Совместимость регулярно тестируется для MDM.
- 4. Ред ОС 8. Рекомендуемая для ФСТЭК-версии.

**Поддерживаемые ОС для сервера системы** (полная совместимость не гарантируется):

- 1. Red Hat Enterprise Linux 8.0.
- 2. Windows Server 2008 R2.
- 3. Linux CentOS 8. Производитель более не поддерживает (с декабря 2021).

# 3.2. Зависимости

- 1. PostgreSQL Pro или PostgreSQL. Применяется версия: 12.
- 2. Apache Tomcat. Применяется версия: 9. ФСТЭК-версия: Libercat.
- 3. **Opensearch**. Применяется версия: 2.7.х.
- 4. **JDK**. Применяется версия: 11. ФСТЭК-версия: AxiomJDK.
- 5. **OrientDB**. Применяется версия: 3.2.23.

# 3.3. Браузеры

Поддерживаются последние версии следующих браузеров:

- 1. Google Chrome (версия 120 и новее) рекомендовано;
- Mozilla Firefox (версия 100 и новее). При использовании разрешите продуктам Юниверс Дата использовать всплывающие окна. Снимите флаг Preferences > Search Results > Block pop-up windows, или там же настройте Exceptions.
- 3. Любой браузер на движке Chromium версии 120 и новее (Microsoft Edge, Яндекс.Браузер и т.п.).

### Рекомендуемое разрешение экрана (для клиента):

- 1. 1600х900 и выше (Не рекомендуется использовать разрешение более 3200х1800, т.к. начнутся проблемы отображения элементов).
- Разрешение 1366х768 не поддерживается, но работа с ним допустима.
   Возможны некорректные отображения элементов интерфейса в разрешении 1366х768 (15.6' ноутбук). Части интерфейса могут наезжать друг на друга.