

DRE Advanced Media Platform Office

Руководство по установке

Индекс	POM-IG
Конфиденциальность	Публичный - L0
Ревизия	1.0
Статус	Согласован

Содержание

1. Аннотация	3
2. Общие сведения	4
2.1. Назначение	4
2.2. Требования к квалификации установщика	4
3. Системные требования	5
3.1. Аппаратное Обеспечение	5
3.2. Программное Обеспечение	5
4. Предварительные действия	7
4.1. Установка PostgreSQL	7
4.2. Настройка PostgreSQL	10
4.3. Установка и настройка Redis (open source)	11
5. Установка и настройка Системы	12
5.1. Развёртывание внутри Kubernetes	12
5.2. Необходимые доступы	12
5.3. Состав репозитория	12
5.3.1. Описание helmfile	12
5.3.2. Описание values	13
5.4. Настройка CD для продукта, опубликованного в Releases	13
5.5. Настройка переменных окружения	13
5.6. Настройка production.yaml	13
5.7. Настройка versions.yaml	14
5.8. Развертывание системы	14
6. Администрирование системы	15

1. Аннотация

Документ содержит руководство по установке и первоначальной настройке системы "DRE Advanced Media Platform Office" (далее - POM или Система).

Документ предназначен для сотрудников отдела мониторинга и инсталляции, а также для других технических специалистов, в обязанности которых входит установка и первоначальная настройка POM.

i Данный документ опубликован исключительно с целью изучения системных требований для установки продукта, а также ознакомления с последовательностью и деталями процесса установки. Реальная установка продукта производится с использованием внутренних репозиториев ООО "Цифра", доступ к которым предоставляется заказчику по запросу.

2. Общие сведения

2.1. Назначение

DRE Advanced Media Platform Office (далее - ПОМ или ПО) предназначен для централизованного учёта данных об аккаунтах оператора платформы DRE Advanced Media Platform. Обеспечивает выполнение следующих функций: регистрация и авторизация аккаунтов; работа с аккаунтом по коду домена: удаление, получение данных; работа с аккаунтами с токеном: удаление, получение данных, выход пользователя из аккаунта; работа с подписками: создание, удаление подписки, получение списка подписок аккаунта и их данных; работа с услугами: получение списка услуг, получение информации об услуге, определение использования услуги или тарифа в подписках; работа с платежами: покупка подписки, проверка и обновление статуса покупки, получение списка платежей и их данных; получение списка аккаунтов по заданным параметрам поиска; получение данных аккаунта; интеграция с системами отправки SMS-сообщений пользователям.

Язык программирования: Go, SQL.

Объем программы для ЭВМ: 125,5 МБ.

2.2. Требования к квалификации установщика

Для установки системы сотрудник обязан:

- иметь базовые представления и практические навыки работы с системой оркестрации Kubernetes и пакетным менеджером Helm.
- иметь навыки работы с ОС семейства Linux, а именно:
 - установка пакетов;
 - создание и настройка сетевых подключений;
 - запуск служб, настройка автозапуска служб;
 - установка и настройка PostgreSQL;
 - создание и работа с БД под управлением PostgreSQL.
- иметь знания о DNS.
- иметь базовые представления и практические навыки работы с Git.
- иметь навыки работы с ssl-сертификатами.

3. Системные требования

3.1. Аппаратное Обеспечение

Рекомендации по минимальной конфигурации серверного обеспечения:

Число пользователей - до 10 000:

1. Сервер с 4 ГБ RAM и 4 ядрами
2. Сервер с 8 ГБ RAM и 8 ядрами.


Число пользователей - до 5 000 000:

1. 2 x Сервер с 8 ГБ RAM и 8 ядрами.
2. 5 x Сервер с 16 ГБ RAM и 4 ядрами.
3. 2 x Сервер с 16 ГБ RAM и 8 ядрами.
4. 1 x Сервер 4 GB RAM и 4 ядрами.


3.2. Программное Обеспечение

Для установки необходимо **предварительно** выполнить следующие условия:

1. Установить и настроить кластер Kubernetes. Так как развертывание производится в кластере k8s, то необходим config file для доступа к кластеру.
 - Если пользователь выполнял развертывание Kubernetes самостоятельно, то он сам должен создать config file (см. документацию Kubernetes).
 - Если Kubernetes был развернут сторонними людьми, то необходимо получить config file у администратора кластера.

 Установкой и администрированием системы Kubernetes занимаются сотрудники Отдела поддержки инфраструктуры.

2. Установить kubectl (ссылка предоставляется по запросу заказчика).
3. Установить helm (ссылка предоставляется по запросу заказчика).

 Helm необходим в случае ручной установки элементов системы в кластер k8s. При стандартной установке с использованием gitlab-ci использование helm не предполагается.


4. Развернуть DNS-сервер, преобразование имен DNS-зоны настроить на мастера k8s (созданы А записи на зону DNS).
5. Развернуть кластер БД (ссылка предоставляется по запросу заказчика). Можно работать как в Standalone режиме, так и в кластерном режиме.
6. Для корректной работы системы необходим доступ к следующим ресурсам:
 - chartmuseum (ссылка и права доступа предоставляются по запросу заказчика)
 - gitlab(ссылка и права доступа предоставляются по запросу заказчика)
7. Необходим доступ к репозиторию (ссылка и права доступа предоставляются по запросу заказчика), содержащему helmfile для развертывания ПОМ. Helm файл содержит инструкции, с помощью которых осуществляются настройки устанавливаемых компонентов. Сами компоненты поставляются в виде

чартов (charts), из которых развертываются Kubernetes Pods. Данные берутся из gitlab (ссылка и права доступа предоставляются по запросу заказчика).

4. Предварительные действия

Для POM используется база данных Redis (open source). Для использования этой базы требуется установить и настроить Redis (open source).

Сама БД развертывается с помощью CI/CD (аналогично другим компонентам POM).


 Можно работать как в Standalone режиме, так и в кластерном режиме.

Установка и настройка PostgreSQL описаны в разделах "[Установка PostgreSQL](#)" и "[Настройка PostgreSQL](#)".
Рекомендуется использовать расширение TimescaleDB для работы с Time-series - партиционирование таблиц, функции агрегации и т.д.

После установки PostgreSQL выполнить действия:

1. Развернуть пулер соединений перед ним (*перед кем?*) (опционально для тестовых зон).
2. Установить расширение TimescaleDB.
3. Создать БД для POM.
4. Создать пользователя с доступом только к этой БД.
5. Записать артефакты конфигурации для последующей настройки POM:
 - IP-адрес и порт мастера (пулера).
 - Имя БД.
 - Логин и пароль пользователя.
 - Имя схемы, используемой в БД (по умолчанию - "public").

4.1. Установка PostgreSQL

 **ВНИМАНИЕ!** Для работы системы POM требуется PostgreSQL версии 12 или выше.
Ниже приведен пример установки PostgreSQL на сервер без развертывания и настройки кластера БД.

1. Перед установкой рекомендуем обновить текущие системные пакеты, если это новый экземпляр сервера:

```
sudo apt update
sudo apt -y install vim bash-completion wget
sudo apt -y upgrade
```

Установите дополнительные пакеты (локаль):

```
locale -a
sudo locale-gen ru_RU.UTF-8
sudo dpkg-reconfigure locales
```

Выполните перезагрузку:

```
sudo reboot
```

2. Добавьте репозиторий PostgreSQL 15:

- a. Перед настройкой репозитория APT импортируйте ключ GPG, используемый для подписи пакетов:

```
sudo apt update
sudo apt -y install gnupg2
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
```

- b. После импорта ключа GPG добавьте содержимое репозитория в ОС:

```
echo "deb http://apt.postgresql.org/pub/repos/apt/ `lsb_release -cs`-pgdg main" |sudo tee /etc
/apt/sources.list.d/pgdg.list
```

- c. Добавленный репозиторий содержит много различных пакетов, включая сторонние дополнения. Они включают:

- PostgreSQL-клиент.
- PostgreSQL.
- libpq-DEV.
- PostgreSQL-сервер-DEV.
- Пакеты pgadmin.
- PostgreSQL timescaledb.

- d. Cat файл, созданный для проверки его содержимого:

```
$ cat /etc/apt/sources.list.d/pgdg.list
deb http://apt.postgresql.org/pub/repos/apt/ buster-pgdg main
```

3. Установка пакетов PostgreSQL 15:

- a. Обновите список пакетов и установите серверные и клиентские пакеты PostgreSQL 15:

```
sudo apt update
sudo apt -y install postgresql-15 postgresql-client-15 postgresql-15-cron
```

- b. Запустите сервер базы данных, используя следующую команду:

```
sudo pg_ctlcluster 15 main start
```

- c. Подтвердите статус службы и используемый файл конфигурации:

```
$ sudo pg_ctlcluster 15 main status
pg_ctl: server is running (PID: 4209)
/usr/lib/postgresql/15/bin/postgres "-D" "/var/lib/postgresql/15/main" "-c" "config_file=/etc
/postgresql/15/main/postgresql.conf"
```

- d. Можно использовать команду *systemctl* для проверки статуса службы. В случае успешной установки выводится сообщение примерно следующего вида:


```

$ systemctl status postgresql.service
postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Sun 2019-10-06 10:23:46 UTC; 6min ago
   Main PID: 8159 (code=exited, status=0/SUCCESS)
   Tasks: 0 (limit: 2362)
   CGroup: /system.slice/postgresql.service
Oct 06 10:23:46 debian systemd[1]: Starting PostgreSQL RDBMS...
Oct 06 10:23:46 debian systemd[1]: Started PostgreSQL RDBMS.

$ systemctl status [email protected]
[email protected] - PostgreSQL Cluster 15-main
   Loaded: loaded (/lib/systemd/system/[email protected]; indirect; vendor preset: enabled)
   Active: active (running) since Sun 2019-10-06 10:23:49 UTC; 5min ago
   Main PID: 9242 (postgres)
   Tasks: 7 (limit: 2362)
   CGroup: /system.slice/system-postgresql.slice/[email protected]
          9242 /usr/lib/postgresql/15/bin/postgres -D /var/lib/postgresql/15/main -c
config_file=/etc/postgresql/15main/postgresql.conf
          9254 postgres: 15/main: checkpointer
          9255 postgres: 15/main: background writer
          9256 postgres: 15/main: walwriter
          9257 postgres: 15/main: autovacuum launcher
          9258 postgres: 15/main: stats collector
          9259 postgres: 15main: logical replication launcher
Oct 06 10:23:47 debian systemd[1]: Starting PostgreSQL Cluster 15-main...
Oct 06 10:23:49 debian systemd[1]: Started PostgreSQL Cluster 15-main.

$ systemctl is-enabled postgresql
enabled

```

- e. Во время установки автоматически создается пользователь postgres. Это суперадминистратор, который имеет полный доступ ко всему PostgreSQL.
4. Проверка соединения с PostgreSQL, настройка пользователя:
- a. Во время установки пользователь postgres создается автоматически. Этот пользователь имеет полный доступ superadmin ко всему экземпляру PostgreSQL.

```
sudo su - postgres
```

- b. (Необязательно) замените пароль пользователя на более надежный:

```
psql -c "alter user postgres with password 'NEW_PASSWORD' "
```

- c. Запускаем PostgreSQL с помощью команды:

```
$ psql
```

- d. Получить информацию о подключении, как показано ниже:

```

$ psql
psql (15.0 (Ubuntu 15.0-1.pgdg18.04+1))
Type "help" for help.

postgres=# \conninfo
You are connected to database "postgres" as user "postgres" via socket in "/var/run/postgresql"
at port "5432".

```

- е. Убедиться, что сервис PostgreSQL запускается при загрузке системы, можно с помощью команд:

```
$ systemctl status postgresql.service
$ systemctl status postgresql@15-main.service
$ systemctl is-enabled postgresql
```

4.2. Настройка PostgreSQL

i Данный раздел следует использовать только в случае установки БД в режиме Standalone.

Следующие действия выполняются на сервере, где будут развернуты базы данных, только после установки пакета postgresql-15.

1. Открыть конфигурационный файл postgresql.conf для редактирования:

```
sudo nano /etc/postgresql/15/main/postgresql.conf
```

2. В файле выполнить следующее:

- а. Изменить значение параметра listen_addresses, как показано ниже, и раскомментировать соответствующую строку:

```
listen_addresses = '*'           # what IP address(es) to listen on;
```

- б. Для настройки автовакуума рекомендуются приведенные ниже значения (использовались при тестировании):

```
autovacuum = on
#log_autovacuum_min_duration = 0
autovacuum_max_workers = 10
autovacuum_naptime = 1s
autovacuum_vacuum_threshold = 50
autovacuum_analyze_threshold = 50
autovacuum_vacuum_scale_factor = 0.01
autovacuum_analyze_scale_factor = 0.02
```

3. Открыть конфигурационный файл pg_hba.conf для редактирования:

```
sudo nano /etc/postgresql/15/main/pg_hba.conf
```

4. Необходимо, чтобы к postgres могли подсоединиться любые процессы с локальной машины и компьютеры из локальной сети (например, с ip 192.168.x.x). Также необходимо указать настройки IPv6. Таким образом, файл может выглядеть следующим образом (рекомендуется задавать уровень доступа менее открытый, чем trust):

```
# "local" is for Unix domain socket connections only
local        all
all
# IPv4 local connections:
host         all                all                127.0.0.1
/32          md5
host         all                all                172.17.0.0
/16          md5
host         all                all                192.168.0.0
/16          md5
# IPv6 local connections:
host         all                all                ::1
/128        md5
```

5. Назначьте права для пользователей владельцев баз: *GRANT CREATE ON DATABASE namestand_pom_usd TO usdadmin*
6. Настройте доступ к базам данных, указанным в таблице ниже:

Название БД	Администратор БД
usd-namestand_pom_usd	usdadmin

7. После внесения изменений перезапустить PostgreSQL:

```
sudo /etc/init.d/postgresql restart
```

4.3. Установка и настройка Redis (open source)

В соответствии с Системными требованиями, для корректной работы системы ПОМ требуется поднять базу данных Redis (open source). Все экземпляры БД должны быть доступны из кластера. Параметры работы с экземплярами БД настраиваются в конфигурационных файлах сервисов ПОМ (см. раздел [Администрирование системы](#)). Конфигурацию самих экземпляров БД менять не требуется.

5. Установка и настройка Системы

Необходимо выполнить установку системы, как описано ниже.

5.1. Развёртывание внутри Kubernetes

Ссылка на проект для развёртывания ПОМ в kubernetes предоставляется заказчику по запросу.

5.2. Необходимые доступы

Для развёртывания проекта требуется получить доступы к следующим проектам в gitlab:

- pom (ссылка и права доступа предоставляются по запросу заказчика);
- pom-based-stand (ссылка и права доступа предоставляются по запросу заказчика);
- cd-templates (ссылка и права доступа предоставляются по запросу заказчика).

5.3. Состав репозитория

В рамках оптимизации процессов разработки и развёртывания приложений был внедрен новый метод деплоя. Рекомендуется ознакомиться с документацией по новому методу (ссылка предоставляется по запросу заказчика), чтобы обеспечить его правильное использование в процессе разработки.

Состав репозитория ПОМ (ссылка и права доступа предоставляются по запросу заказчика)

- helmfile.yaml - конфиг helmfile;
- values - конфигурационные шаблоны helmfile для генерации values.yaml файлов для каждого helm chart;
- default.yaml - значения по умолчанию;
- versions.yaml - файл, содержащий последние стабильные версии сервисов.

5.3.1. Описание helmfile

Helm-файл содержит инструкции, с помощью которых осуществляются настройки устанавливаемых компонентов ПОМ. Сами компоненты поставляются в виде образов (images), из которых разворачиваются Docker-контейнеры. helmfile.yaml имеет строгую типизированную yaml структуру и имеет несколько основных и дополнительных секций.

В helmfile.yaml содержится следующая информация:

```
repositories:
  - name: < >
    url: <url , chart'>

releases:
  - name: < chart'>
    namespace: < namespace'>
    chart: < chart' >
    version: < chart'>
    condition: < chart'>
    labels: # chart'a
      stage: <, chart'a>
    values: #
      - "./values/< chart'>/values.yaml.gotmpl"
```

Ссылка на актуальный helmfile.yaml предоставляется по запросу заказчика.

5.3.2. Описание values

В директории values хранятся конфигурационные шаблоны helmfile для генерации values.yaml файлов для каждого helm chart.

Фактические значения для шаблонов хранятся в файле default.yaml, что позволяет адаптировать развертывание под конкретное окружение, используя одинаковые шаблоны.

Если значение отсутствует в default.yaml, то оно может быть задано в качестве дефолтного внутри шаблона.

Ссылка на актуальный default.yaml предоставляется по запросу заказчика.

5.4. Настройка CD для продукта, опубликованного в Releases

Ссылка на документ с описанием настроек предоставляется по запросу заказчика.

5.5. Настройка переменных окружения

Настройка переменных осуществляется в gitlab.

В боковом меню выбрать **Settings** (на панели слева) -> **CI/CD** -> **Environment variables**. Отредактировать переменные.


Таблица с описанием используемых переменных Gitlab

Переменная	Описание
POSTGRES_LOGIN	Имя администратора PostgreSQL БД
POSTGRES_PASSWORD	Пароль администратора PostgreSQL БД
USDDB_LOGIN	Логин для подключения к usd DB (по умолчанию - usdadmin)
USDDB_PASSWORD	Пароль для подключения к usd DB (по умолчанию - usdadmin)
GLOBAL_DB_HOST	Адрес хоста
GLOBAL_DB_PORT	Порт
GLOBAL_DB_SYNC_PORT	Порт синхронной реплики
GLOBAL_DB_ASYNC_PORT	Порт асинхронной реплики
DOMAIN_ZONE	Адрес доменной зоны
SIGN	Имя БД, если не задано то создается база со стандартным для сервиса именем

 **ВАЖНО!** Environment variables имеют более высокий приоритет, чем переменные, заданные в файлах.

5.6. Настройка production.yaml

В файле production.yaml нужно настроить параметры, которые используются в промышленной эксплуатации в соответствии с требуемой конфигурацией. Файл production.yaml переопределяет переменные в файле default.yaml.

 Настройка файлов values.yaml.gotmpl осуществляется только разработчиками продукта.

5.7. Настройка versions.yaml

versions.yaml - файл с версиями компонентов. В файле versions.yaml необходимо установить версию продукта /продуктов, до которой необходимо обновиться.

5.8. Развертывание системы

Развертывание ПОМ осуществляется в Gitlab CI/CD:

1. В боковом меню выбрать **Build** (на панели слева) -> **Pipelines**.
2. Нажать lint (кнопка ">>"). Во всплывающем окне нажать кнопку Play.
3. Нажать common (кнопка ">>"). helmfil.yaml имеет строгую типизированную yaml структуру и имеет несколько основных и дополнительных секций.

Во всплывающем окне нажать кнопку Play. 4. Дождаться окончания операции.

Примечание. Развертывание системы (с помощью CI/CD) выполняется в ДВА этапа (эти два этапа могут быть разнесены по времени):

1. Установка / Обновление базы данных ПОМ DB.
2. Установка / Обновление сервисов и служб, входящих в состав ПОМ.

6. Администрирование системы

Информация об администрировании POM приведена в отдельном документе, ссылка на который предоставляется заказчику по запросу.

© ООО "Цифра", 2024

Документация "DRE Advanced Media Platform Office. Руководство по установке" является объектом авторского права. Воспроизведение всего произведения или любой его части воспрещается без письменного разрешения правообладателя.