

DRE Advanced Media Platform META DATA SERVER

Руководство по установке

Индекс	MDS-IG
Конфиденциальность	Публичный - L0
Ревизия	3.0
Статус	Согласован

Содержание

1. Аннотация	3
2. Общее описание	4
3. Минимальные системные требования	5
4. Установка сервиса	6
4.1. Настройка работы с DRE Account Manager	6
4.2. Настройка Apache Pulsar	6
4.3. Настройка БД для микросервисов MDS	7
4.4. Установка	7
4.4.1. Завершение установки	10
4.4.1.1. Настройка сервиса Metadata Manager	10
4.4.1.2. Настройка сервиса User Content Advisor	10
4.4.1.3. Настройка сервиса Search Engine	10

1. Аннотация

Документ предназначен для технических специалистов, занимающихся установкой, настройкой и поддержкой сервиса. Документ рассчитан на инженеров, обладающих специальными навыками и знаниями в области инсталляции программного обеспечения.

2. Общее описание

DRE Advanced Media Platform META DATA SERVER (далее - MDS) - сервер, осуществляющий сбор и хранение метаданных к контенту, предоставляемому DRE Advanced Media Platform. Сбор метаданных происходит автоматически с внешних источников, их редактирование возможно и автоматически, и вручную. Также MDS участвует в формировании витрины контента на телеэкране для абонента ТВ. MDS взаимодействует с подсистемой шифрования и оплаты, что позволяет создавать и редактировать коммерческие пакеты и типы покупок через веб-интерфейс сервера.

3. Минимальные системные требования

Для установки сервиса необходимо наличие не менее 3 серверов без раздела подкачки swap с разными именами (hostname): master, node1, node2. Общее количество серверов должно быть нечетным.

Сервера должны удовлетворять следующим требованиям:

1. Операционная система ubuntu-18.04-server-amd64 (с установленным пакетом sudo).
2. Многоядерный центральный процессор с тактовой частотой каждого ядра 2 ГГц (не менее 20-ти ядер).
3. Объем оперативной памяти 64 ГБ.
4. Не менее 2-ух жестких дисков емкостью не менее чем по 500 ГБ. Рекомендуется наличие на каждой ноде, помимо основного дискового пространства с ОС, одного диска SSD или NVMe и девяти дисков HDD (SATA, SAS), не собранных в RAID и не форматированных.
5. Два интерфейса Ethernet 100 и 1000 Base-T с поддерживаемой пропускной способностью 100 и 1000 Мбит/сек соответственно. Один предназначен для сети поддержки, второй используется для вывода генерируемого транспортного потока.
6. Свободное место для папки временных файлов /tmp - 10 ГБ.

Установка должна производиться с дополнительного Ubuntu-сервера, не имеющего отношения к будущему кластеру. Требования к объему ресурсов дополнительного сервера отсутствуют.

Рекомендуемая ОС - Ubuntu с 18.04 с актуальным HWE ядром.

4. Установка сервиса

Для установки MDS необходимо предварительно установить и запустить Apache Pulsar и Account Manager.

4.1. Настройка работы с DRE Account Manager

После установки продукта DRE Account Manager необходимо через WEB UI DRE Account Manager вручную создать сервисы с фиксированными кодами:

- metadatamanager;
- metadatastorage;
- metadataenhancers;
- logmanagement;
- imageservice;
- sts;
- scheduleservice.

Процесс создания сервиса в WEB UI описан в руководстве пользователя DRE Account Manager.

4.2. Настройка Apache Pulsar

Для установки MDS необходимо предварительно установить и запустить Apache Pulsar.

Далее, в Apache Pulsar необходимо создать основного владельца (tenant) и окружения (namespace) для микросервисов. Сделать это можно через консоль в Kubernetes.

В версии Apache Pulsar ниже версии 2.8.1 обнаружен баг, нарушающий работу микросервиса STS, так как полностью очищает старые данные при закрытии ledger в Pulsar (см. <https://github.com/apache/pulsar/issues/11241>). Как временное решение, до обновления версии Apache Pulsar, необходимо выставить бесконечное хранение данных в неймспейсе mds/sts.

В namespace pulsar открыть консоль пода pulsar-toolset и выполнить:

```
sh
# MDS
bin/pulsar-admin tenants create mds
# MS
bin/pulsar-admin namespaces create mds/metadatastorage
bin/pulsar-admin namespaces set-deduplication mds/metadatastorage --enable
# MM
bin/pulsar-admin namespaces create mds/metadatamanager
bin/pulsar-admin namespaces set-deduplication mds/metadatamanager --enable
# SS
bin/pulsar-admin namespaces create mds/schedule-service
bin/pulsar-admin namespaces set-deduplication mds/schedule-service --enable
# STS
bin/pulsar-admin namespaces create mds/sts
bin/pulsar-admin namespaces set-retention mds/sts --size -1 --time -1
```

4.3. Настройка БД для микросервисов MDS

Перед установкой MDS необходимо создать и настроить базы данных для следующих микросервисов MDS: Account Manager, Log Management, Image Service, Metadata Storage, Schedule Service, Metadata Manager, Metadata Enchancers, Settings and Translation Service, User Content Advisor. Пример настройки pgbouncer для баз данных приведен в Руководстве администратора (предоставляется по запросу заказчика).

Используемые БД должны быть развернуты с параметрами:

```
Collate='en_US.UTF-8' Ctype='en_US.UTF-8'
```

1. Скопируйте файлы numbers.syn, part_of_speech_russian.stop, ru_ru.affix, ru_ru.dict из репозитория MDS \Services\MDS\5.2\5.2 Rev 1_distr, поместите в директорию сервера базы данных /usr/share/postgresql /<VER>/tsearch_data/ (путь может отличаться на разных системах).
2. Выполнить команды из-под учетной записи с привилегией суперпользователя в разрезе соответствующей базы данных микросервиса:

а. Для БД MetadataStorage:

```
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT,INSERT,DELETE,UPDATE ON TABLES TO <metadata_storage_user>;
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT USAGE,SELECT ON SEQUENCES TO <metadata_storage_user>;
```

б. Для БД MetadataManager:

```
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT SELECT,INSERT,DELETE,UPDATE ON TABLES TO <metadata_manager_user>;
ALTER DEFAULT PRIVILEGES IN SCHEMA public GRANT USAGE,SELECT ON SEQUENCES TO <metadata_manager_user>;
```

4.4. Установка

Для установки сервиса в имеющийся настроенный кластер Kubernetes используется процесс CI/CD, настраиваемый с помощью GitLab. Весь процесс описан в документе в Git, доступ к документу ограничен и предоставляется по запросу заказчика.

Конфигурирование сервиса MDS осуществляется путем изменения значений параметров в helm-файле.

Ниже приведены примеры параметров компонентов MDS, обязательных для переопределения в helmfile (по сравнению со значениями по умолчанию). Набор параметров и выставляемых значений может меняться в соответствии с требованиями и задачами заказчика. Описание специфических параметров для MDS приведено в Руководстве администратора (предоставляется по запросу заказчика).

Параметр	Описание	Пример или рекомендуемое значение
	##### KEEPALIVED #####	
Values: Ips:	Виртуальные IP-адреса нод кластера.	- 192.168.11.245 - 192.168.11.246 - 192.168.11.247

Values: Servers:	Имена хостов нод кластера, начиная с мастер-ноды с указанием физического сетевого интерфейса для работы виртуальной сети и IP-адреса ноды (через символ #)	- master#ens18#192.168.13.45 - node1#ens18#192.168.13.46 - node2#ens18#192.168.13.47 - node3#ens18#192.168.13.48 - node4#ens18#192.168.13.49
##### ROOK CLUSTER #####		
Dashboard: HostName:	Доменное имя для Ceph Dashboard.	ceph.cs.kube.tz.cas
Objects: FileSystem: FileBrowser: HostName:	Доменное имя для Ceph File Browser.	fb.cs.kube.tz.cas
Storage: Nodes: - name:	Имена хостов нод хранилища Ceph.	"node6-hard"
devices: - name: FullPath: config: storeType:	Настройки файловых хранилищ элементов кластера. Имя узла. Путь к хранилищу. Тип хранилища.	"sdc" "/dev/sdc" bluestore
##### MDS NGINX #####		
version	Версия чарта.	
values: ingress: hostname:	Доменное имя прокси-сервиса MDS.	mds.gs-labs.tv
##### ACCOUNT-MANAGER#####		
(аналогично для metadata-manager, metadata-storage, metadata enhancer, search engine. log management service, image service, schedule-service)		
version:	Версия сервиса.	
values: Replicas:	Количество реплик (контейнеров) сервиса.	1

Variables:	Имя БД для подключения.	account-manager
DbName:		
Variables:	Имя или адрес сервера БД.	127.0.0.1
DbHost:		
Variables:	Имя пользователя для подключения к БД.	user
DbUser:		
Variables:	Пароль для подключения к БД.	passwd
DbPassword:		
EnvVars: <variable>:	Значение переменной окружения, соответствующей параметру из настроек компонента (см. рук-во администратора); значение задается как строка.	"365"
##### MDS FRONTEND (аналогично для mds frontend-react) #####		
version:	Версия сервиса.	
values: ingress: hostname:	Доменное имя сервиса.	uimds.gs-labs.tv
values: EnvVars: API_HOST_ACCOUNT API_HOST_LOG API_HOST_IMAGE_SERVICE API_HOST_STORAGE API_HOST_SCHEDULE API_HOST_MANAGER API_HOST_ENHANCER API_HOST_STS	Доменное имя сервисов account manager, metadata manager, metadata storage, metadata enhancers, search engine, image service	mds.gs-labs.tv
##### user-vod-purchases ##### (аналогично для channel-source-distribution)		

version:	Версия сервиса.	
DRM_IP:	Адрес DRM сервера.	http://drm-svc:80
##### vod-source-distribution #####		
version:	Версия сервиса.	
SVOD_EXCLUDE_PLATFORM_IDS	Перечень кодов платформ, для которых SVOD-предложения всегда недоступны.	["ANDROID", "IOS", "TVOS"]
DRM_IP:	Адрес DRM сервера.	http://drm-svc:80
##### movie-recommendations #####		
version:	Версия сервиса.	

4.4.1. Завершение установки

4.4.1.1. Настройка сервиса Metadata Manager

После установки MDS отправьте на Metadata Manager запрос, осуществляющий запуск задачи синхронизации данных между компонентами MDS (указав токен авторизации):

```
POST /api/v1/vod/internal/metadata/resync-shows/
```

Результат выполнения синхронизации отображается в логах подов mm-celery-external-sync.

Доступные параметры запроса:

- fresh_start=true - запуск для всего контента из manager_title (по умолчанию false);
- clear_unrelated_titles=true - удалить контент из manager_title, который не используется (по умолчанию false);
- sync_only_fields=countries,genres,staff,tags,ratings - список полей контента, которые необходимо синхронизировать (по умолчанию все).

4.4.1.2. Настройка сервиса User Content Advisor

После установки MDS необходимо выполнить следующую команду в контейнере user-content-advisor-app:

```
python uca.py full-sync metadata
```

4.4.1.3. Настройка сервиса Search Engine

После установки MDS, для начального заполнения индексов необходимо выполнить следующие команды в поде search-engine-app:

```
python es.py reindex --recreate person
python es.py reindex --recreate metadata
python es.py reindex --recreate program
python es.py reindex --recreate feed
```

Выполнение команд наполнит индекс поиска по персонам/метадате/программам/фидам и в дальнейшем будет обновлять его при получении сообщений из Apache Pulsar.

Также необходимо выполнить команду для синхронизации сортировок программ:

```
python es.py reindex --recreate orderingconf
```

© ООО "Цифра", 2017-2024

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