

DRE Advanced Media Platform SCRAMBLER

Руководство по установке

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

Содержание

1. Аннотация	3
2. Требования к квалификации установщика	4
3. Системные требования	5
3.1. Предварительные действия	5
3.1.1. Установка PostgreSQL	5
3.1.2. Настройка PostgreSQL	8
4. Установка и настройка системы	10
4.1. Процедура установки	10
4.1.1. Развёртывание внутри Kubernetes	10
4.1.1.1. Необходимые доступы	10
4.1.1.2. Состав репозитория	10
4.1.1.3. Описание helmfile	10
4.1.1.4. Описание templates	10
4.1.2. Настройка CD для продукта, опубликованного в Releases	11
4.1.3. Альтернативная развертка scr_manager в качестве контейнера	11
4.1.4. Настройка и развёртывание Scrambler	15
4.1.4.1. CD для артефактов БД	15
4.1.4.2. CD для KeyDB	15
4.1.4.3. CD для бновления support_server	16
4.1.4.4. Описание параметров scr_manager	16
4.1.4.5. Описание параметров key_db	19
4.1.4.6. Описание параметров ags	21
4.1.4.7. Описание параметров ags_web	22
4.1.4.8. Описание параметров ems	23
4.1.4.9. Описание параметров cron	24
4.1.4.10. Описание параметров scrambler_ui	25
4.1.4.11. Описание параметров секции db	25
4.1.4.12. Настройка переменных окружения	26
4.2. Подготовка локального сервера для запуска Scrambler Task	26
4.2.1. Установка пакетов Docker	26
4.2.2. Добавление пользователя в группу docker	26
4.2.3. Установка nvidia-container-toolkit	26
4.2.4. Обновление драйверов	27
4.2.5. Запуск docker daemon	28
4.2.6. Mount папки output_dir	28

1. Аннотация

Данный документ содержит руководство по установке и первоначальной настройке DRE Advanced Media Platform SCRAMBLER (далее - SCRAMBLER или Система), а также описание системных требований для компонентов.

Документ предназначен для технических специалистов, в обязанности которых входит установка и первоначальная настройка системы SCRAMBLER.

Перед установкой системы рекомендуется изучить технические особенности построения и функционирования SCRAMBLER. Данная информация содержится в документах "Scrambler. Общее описание" и "Scrambler. Техническое описание", входящих в комплект поставки.

Данное описание является документом для внутреннего пользования, т.е. распространяется среди сотрудников GS Labs и партнеров компании.



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. Требования к квалификации установщика

Для установки системы сотрудник обязан:

- иметь навыки работы с ОС Ubuntu, 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 базовые представления и практические навыки работы с Git.
- иметь базовые представления и практические навыки работы с Docker.

3. Системные требования

Для установки необходимо предварительно выполнить следующие требования:

- Установлен и настроен кластер Kubernetes.
 - Так как развертывание производится в кластере k8s, то необходим config file для доступа к кластеру.
 1. Если пользователь выполнял развертывание Kubernetes самостоятельно, то он сам должен создать config file (см. документацию Kubernetes).
 2. Если Kubernetes был развернут сторонними людьми, то необходимо получить config file у администратора кластера.
- Установлен kubectl.
- Установлен helm.
- Развернут DNS-сервер, преобразование имен dns зоны настроено на мастера k8s (созданы A записи на зону dns).
- Для корректной работы Scrambler требуется Redis база данных(устанавливается в кластере);
- Для корректной работы системы Scrambler требуется развернуть кластер БД.
- Для корректной работы системы Scrambler необходим доступ к следующим ресурсам:
 - chartmuseum cas-dep (ссылка предоставляется по запросу заказчика)
 - chartmuseum svc-dep (ссылка предоставляется по запросу заказчика)
 - gitlab (ссылка предоставляется по запросу заказчика)
- Необходим доступ к репозиторию, содержащему helmfile для развертывания Scrambler. Helm файл содержит инструкции, с помощью которых осуществляются настройки устанавливаемых компонентов (Manager, web, ags и т.д.). Сами компоненты поставляются в виде образов (images), из которых разворачиваются Docker-контейнеры. Данные берутся из репозитория git.

Для сохранения результатов обработки требуется наличие удаленного NFS-сервера.


Также для корректной работы Scrambler необходимо наличие минимум одного сервера для запуска Scrambler Task:

- Операционная система ubuntu-20.04-server-amd64 (с установленным пакетом sudo).
- Docker версии 20.10.17 и выше.
- Процессор - не менее 4 ядер, не менее 4GB ОЗУ.
- При необходимости транскодирования на GPU, требуется так же наличие видеокарты nvidia не младше 600 серии и драйвер не ниже версии 520.56.06.

Сервер(-а) необходимо устанавливать в локальной сети, защищенной от доступа извне.

3.1. Предварительные действия

3.1.1. Установка PostgreSQL

 Если установка БД производится "с нуля", то необходимо развернуть кластер БД (ссылка на документ предоставляется по запросу заказчика).

Ниже приведен пример установки PostgreSQL на сервер без развертывания и настройки кластера БД.

1. (Рекомендуется) 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 АРТ 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:
 - i. PostgreSQL-клиент.
 - ii. PostgreSQL.
 - iii. libpq-DEV.
 - iv. PostgreSQL-сервер-DEV.
 - v. Пакеты pgadmin.
- 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-14 postgresql-client-15
```

- 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/15/main/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: 15/main: 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 создаётся пользователь `postgres`. Это пользователь со статусом `superadmin`, 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 14.0-1.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".
```

e. 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
```

3.1.2. Настройка PostgreSQL



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.

Открыть конфигурационный файл postgresql.conf для редактирования:

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

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

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

Открыть конфигурационный файл pg_hba.conf для редактирования:

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

Необходимо, чтобы к postgres могли подключиться 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
/0          md5
host        all                all                172.17.0.0/0
host        all                all                192.168.0.0/0
md5
# IPv6 local connections:
host        all                all                ::1
/128          md5
```

При работе Scrambler требуются подключения к базам данных, приведенным в таблице ниже. Необходимо настроить к ним доступ:

Название БД	Администратор БД
screrrmap	errmap
scr	scradmin

После внесения изменений перезапустить PostgreSQL:

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

4. Установка и настройка системы

4.1. Процедура установки

4.1.1. Развёртывание внутри Kubernetes

Ссылка на проект для развёртывания в kubernetes предоставляется по запросу заказчика.

4.1.1.1. Необходимые доступы

Для развёртывания проекта scrambler потребуется получить доступы к проектам scrambler, scrambler-based-stand, cd-templates. Ссылка на проекты предоставляется по запросу заказчика.

4.1.1.2. Состав репозитория

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

4.1.1.3. Описание helmfile

Helm файл содержит инструкции, с помощью которых осуществляются настройки устанавливаемых компонентов Scrambler. Сами компоненты поставляются в виде образов (images), из которых разворачиваются Docker-контейнеры.

В helmfile.yaml содержится следующая информация:

```
repositories:  
- name: <название репозитория>  
  url: <url репозитория, в котором хранятся chart'ы>  
  
releases:  
- name: <название chart'a>  
  namespace: <название namespace'a>  
  chart: <путь до chart'a из репозитория>  
  version: <версия chart'a>  
  condition: <условие деплоя chart'a>  
  labels: # Список меток для деплоя chart'a  
    stage: <стадия, на которой происходит развертка chart'a>  
  values: # Указывается путь до конфигурационных шаблонов  
    - "./values/<название chart'a>/values.yaml.gotmpl"
```

4.1.1.4. Описание templates

В данной директории 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.

4.1.2. Настройка CD для продукта, опубликованного в Releases

Ссылка на документ с описанием настроек предоставляется по запросу заказчика.

4.1.3. Альтернативная развертка `scg_manager` в качестве контейнера

ВНИМАНИЕ! Для работы с мультикаст live потоком `scg_manager` необходимо устанавливать на сервер, сеть которого имеет беспрепятственный доступ до хоста, раздающего мультикаст поток, а не в кластер. В качестве такого сервера может быть использован сервер для запуска Scrambler Task.

Для установки `scg_manager` используется утилита `docker-compose`. Установка пакетов `Docker-compose` выполняется при наличии доступа в Интернет. Инструкция по установке `Docker-compose` описана на официальном сайте: <https://docs.docker.com/compose/install/#install-compose>. Актуальный `docker-compose` файл, необходимый для поднятия контейнера `scg_manager` можно просмотреть по ссылке, предоставляемой заказчику по требованию.

Ниже представлена таблица, которая описывает параметры, используемые `scg_manager` сервером при деплое в качестве `docker`-контейнера на Scrambler Task сервер.

ВНИМАНИЕ! Формат `yaml/yaml` имеет специфические требования к синтаксису (например, в `yaml/yaml` файле нельзя использовать табуляцию). В связи с этим настоятельно рекомендуется проверить файл на корректность синтаксиса.

Ниже выделены параметры, которые с большей вероятностью будут уникальными в рамках каждого стенда.

Параметр	Описание
<code>version:</code>	Версия файла. В разных версиях файлов <code>docker-compose</code> некоторые опции могут быть доступны, отсутствовать, задаваться по-другому либо не работать с определенными версиями файла.
<code>services:</code>	Перечень <code>docker</code> -контейнеров, работой которых управляет <code>docker-compose</code> . Параметры каждого контейнера задаются в отдельной секции.
<code>scrambler_manager:</code>	Настройки Scrambler Manager:

image:	Образ для контейнера. Можно указать либо репозиторий / тег, либо идентификатор частичного изображения. В стандартном развертывании указываются ссылки на репозитории, указанные Производителем по запросу Заказчика. Примечание: если образ не существует, Compose пытается его вытащить (docker pull), если только вы не указали сборку, и в этом случае он строит её с использованием указанных опций и помечает её указанным тегом.
container_name:	Имя настраиваемого контейнера. В результате контейнер будет иметь указанное имя, в противном случае docker-compose генерирует имя по умолчанию. Примечание: поскольку имена Docker-контейнеров должны быть уникальными, то в случае задания этого параметра вы не сможете масштабировать службу за пределами одного контейнера. Попытка сделать это приведет к ошибке.
environment:	Переменные среды:
LOGGER_LEVEL	Возможные значения: 0 - trace, 1 - debug, 2 - info (значение по умолчанию), 3 - warning, 4 - error, 5 - fatal.
SYSTEM_CACHE_DB	Номер базы данных для доступа scr_manager.
SYSTEM_CACHE_EXPIRE_TIME	TTL записей в кеше.
SYSTEM_CACHE_HOST	Ip адрес ноды, через которую будет осуществляться доступ к key_db
SYSTEM_CACHE_PASSWORD	Порт ноды, через который будет осуществляться доступ к key_db
SYSTEM_CACHE_TIMEOUT	Таймаут.
SYSTEM_DOWNLOADER_ATTEMPT_PERIOD	Время между попытками загрузки.
SYSTEM_DOWNLOADER_ATTEMPTS	Кол-во попыток загрузки.
SYSTEM_DOWNLOADER_FILE_TIMEOUT	Таймаут при скачивании видеофайла.
SYSTEM_DOWNLOADER_IMAGE	Адрес для загрузки образа downloader'a.
SYSTEM_DOWNLOADER_MAX_PARALLEL	Кол-во одновременного загружаемых чанков.
SYSTEM_DOWNLOADER_PLAYLIST_TIMEOUT	Таймаут при скачивании плейлиста.
SYSTEM_DOWNLOADER_SEGMENT_DURATION	Время сегментов загрузки.

SYSTEM_HTTP_ADDRESS	"Адрес сервер в формате ""0.0.0.0:". По умолчанию используется порт 8080."
SYSTEM_HTTP_READ_TIMEOUT	Таймаут на чтение запросов scr_manager.
SYSTEM_HTTP_WRITE_TIMEOUT	Таймаут на запись запросов scr_manager.
SYSTEM_INSTANCES_SETTINGS_CREATE_PIPES	Создание пайпов/файлов во время работы инстансов.
SYSTEM_INSTANCES_SETTINGS_DOCKER_DIR	Стандартная папка для хранения файлов, результатов работы docker контейнеров.
SYSTEM_INSTANCES_SETTINGS_FAIL_AFTER	"Время для перевод инстанса в статус ""Failed"" при отсутствии на нем обработки данных."
SYSTEM_INSTANCES_SETTINGS_FFPROBE_TIMEOUT	Таймаут при работе с ffprobe.
SYSTEM_INSTANCES_SETTINGS_MOUNTED_DIR	Префикс приманченной директории для составления итогового url мастер-плейлиста.
SYSTEM_INSTANCES_SETTINGS_OUTPUT_DIR	Папка для публикации.
SYSTEM_INSTANCES_SETTINGS_PROXY_OUTPUT_TYPE	Тип прокси элемента.
SYSTEM_INSTANCES_SETTINGS_PROXY_SETTINGS_HOST	Хост прокси элемента.
SYSTEM_INSTANCES_SETTINGS_PROXY_SETTINGS_TIMEOUT	Таймаут прокси.
SYSTEM_INSTANCES_SETTINGS_SKIP_INSTANCE_DELETION	Включение удаления инстансов с сервера после завершения задачи.
SYSTEM_INSTANCES_SETTINGS_STATUS_CHECK	Период опроса статусов работы инстансов.
SYSTEM_INSTANCES_SETTINGS_TIMEOUT	Таймаут.
SYSTEM_OPERATOR	Используемый оператор.
SYSTEM_PACKAGER_CHUNK_LIST_SIZE	Предельное кол-во сохраняемых чанков для rtsp потока.
SYSTEM_PACKAGER_ENCRYPTION_SCHEME	Схема шифрования.
SYSTEM_PACKAGER_FFMPEG_JOIN_A_V	Флаг, отвечающий за сбор видео-, аудио- потоков в 1 файл. Актуален для профилей шифрования gs_aes и plain_hls.
SYSTEM_PACKAGER_IMAGE	Адрес для загрузки образа packager'a.
SYSTEM_PACKAGER_KEY_ROTATION_PERIOD	Период ротации ключей.
SYSTEM_PACKAGER_KMS_TIMEOUT	Таймаут kms сервера.
SYSTEM_PACKAGER_KMS_URL	Адрес kms сервера.

SYSTEM_PACKAGER_SEGMENT_DURATION	Время на которое делятся сегменты видео.
SYSTEM_HLS_PACKAGER_IMAGE	Адрес для загрузки образа hls_3_packager_process.
SYSTEM_HLS_ENCRYPTOR_IMAGE	Адрес для загрузки образа hls_3_playlist_encryptor_process.
SYSTEM_PIPELINE_ATTEMPTS_NUMBER	Количество попыток проверки.
SYSTEM_PIPELINE_PING_INTERVAL	Интервал проверки состояния.
SYSTEM_PROCESS_MONITOR_UPDATE_DURATION	Время обновлений.
SYSTEM_PROMETHEUS_HTTP_ADDRESS	Адрес прометеус сервера с метриками.
SYSTEM_PROMETHEUS_HTTP_READ_TIMEOUT	Таймаут на чтение запросов прометеус.
SYSTEM_PROMETHEUS_HTTP_WRITE_TIMEOUT	Таймаут на запись запросов прометеус.
SYSTEM_PUBLISHER_IMAGE	Адрес для загрузки образа publisher'a.
SYSTEM_PUBLISHER_STATUS_TIMEOUT	Таймаут работы публшера.
SYSTEM_PUBLISHER_UPDATE_PERIOD	Период обновления данных публшера.
SYSTEM_PUBLISHER_MAX_WORKERS_VOD	Максимальное число процессов для публикации VOD контента.
SYSTEM_PUBLISHER_MAX_WORKERS_LIVE	Максимальное число процессов для публикации LIVE контента.
SYSTEM_PUBLISHER_DELETE_AFTER_COPY	Флаг, отвечающий за удаление уже скопированных чанков в исходных папках.
SYSTEM_RTSP_PROCESS_CHUNK_LIST_SIZE	Предельное кол-во сохраняемых чанков для rtsp потока.
SYSTEM_RTSP_PROCESS_IMAGE	Адрес для загрузки образа rtsp_process'a.
SYSTEM_RTSP_PROCESS_SEGMENT_DURATION	Время на которое делятся сегменты видео.
SYSTEM_RTSP_PROCESS_WAIT_TIMEOUT	Таймаут для перехода rtsp задачи в статус In progress.
SYSTEM_SUPPORT_SERVER_FFPROBE_TIMEOUT	Таймаут ffprobe
SYSTEM_SCRAMBLER_DB_ASYNC_REPLICA_DB_NAME	Наименование базы данных Scrambler.
SYSTEM_SCRAMBLER_DB_ASYNC_REPLICA_HOST	Адрес БД.
SYSTEM_SCRAMBLER_DB_ASYNC_REPLICA_MAX_CONNS	Максимальное количество соединений к БД.
SYSTEM_SCRAMBLER_DB_ASYNC_REPLICA_PASSWORD	Пароль для доступа к БД Scrambler.

SYSTEM_SCRAMBLER_DB_ASYNC_REPLICA_PORT	Порт для доступа к БД Scrambler.
SYSTEM_SCRAMBLER_DB_ASYNC_REPLICA_TIMEOUT	Таймаут для доступа к БД Scrambler.
SYSTEM_SCRAMBLER_DB_ASYNC_REPLICA_USER	Пользователь для доступа к БД Scrambler.
SYSTEM_SCRAMBLER_DB_MASTER_DB_NAME	Наименование базы данных Scrambler.
SYSTEM_SCRAMBLER_DB_MASTER_HOST	Адрес БД.
SYSTEM_SCRAMBLER_DB_MASTER_MAX_CONNS	Максимальное количество соединений к БД.
SYSTEM_SCRAMBLER_DB_MASTER_PASSWORD	Пароль для доступа к БД Scrambler.
SYSTEM_SCRAMBLER_DB_MASTER_PORT	Порт для доступа к БД Scrambler.
SYSTEM_SCRAMBLER_DB_MASTER_TIMEOUT	Таймаут для доступа к БД Scrambler.
SYSTEM_SCRAMBLER_DB_MASTER_USER	Пользователь для доступа к БД Scrambler.
SYSTEM_SERIALIZATION_EMIT_DEFAULTS	Серилизация значений по умолчанию.
SYSTEM_TRANSCODER_IMAGE	Адрес для загрузки образа transcoder'a.
SYSTEM_TRANSCODER_PROTOCOL	Протокол взаимодействия для транскодера.
SYSTEM_TRANSCODER_SYNC EVERY	Период синхронизации транскодера.
SYSTEM_TRANSCODER_TRANSCODING_GPU_NUMBER	Номер устройства для транскодирования.

4.1.4. Настройка и развертывание Scrambler

4.1.4.1. CD для артефактов БД

При развертывании Scrambler происходит установка SCH и API для БД через механизм Kubernetes Jobs. В процессе установки сохраняется лог в контейнере.

```
scr_db_sch:
  enabled: true

scr_db_api:
  enabled: true

# You can optionally override database address and port here:
db:
  db_mng: scr
  address: 11.111.11.111
  master_port: 5432
```

Этот режим поддерживают scr_db_*, error_mapper_db_*.

4.1.4.2. CD для KeyDB

Для хранения состояний задач Scrambler использует keydb-ha-cluster, доступ до которого осуществляется через Nodeport. В качестве стандартного порта используется "31379".

```
key_db:
  enabled: true
  system:
    password: password
    timeout: 10 # KeyDB.
    NodeIp: 22.222.22.222 # KeyDB master-node.
    NodePort: 31379
```

4.1.4.3. CD для бновления support_server

При разворачивании Scrambler происходит обновление support_server на всех ранее созданных серверах для запуска Scrambler Task через механизм Kubernetes Jobs. В процессе установки сохраняется лог в контейнере.

```
support_server_init:
  enabled: true
```

4.1.4.4. Описание параметров scr_manager

Ниже представлена таблица, которая описывает параметры, используемые scr_manager сервером при деплое и работе.

Параметр	Описание
scr_manager.enabled	Добавление сервиса в деплой.
scr_manager.replicas	Кол-во реплик пода.
scr_manager.hpa.enabled	Включение hpa.
scr_manager.hpa.minReplicas	Минимальное число hpa реплик.
scr_manager.hpa.maxReplicas	Максимальное число hpa реплик.
scr_manager.hpa.targetCPUUtilizationPercentage	Процент целевого использования CPU для hpa.
scr_manager.resources.limits.cpu	Ограничение сри для пода
scr_manager.resources.limits.memory	Ограничение памяти для пода
scr_manager.resources.requests.cpu	Запрос сри для пода
scr_manager.resources.requests.memory	Запрос памяти для пода
scr_manager.ingress.hostname	Адресс ingress'a сервиса.

scr_manager.ingress.enabled	Добавление ingress в деплой.
scr_manager.ingress.annotations	Аннотации для ингресса
scr_manager.udpNodePort.enabled	Использовать NodePort для сервиса.
scr_manager.udpNodePort.portNumber	Номер NodePort'a.
scr_manager.configEnabled	Использование конфига.
scr_manager.config.logger.level	Степень логирования событий. Возможные значения: 0 - trace, 1 - debug, 2 - info (значение по умолчанию), 3 - warning, 4 - error, 5 - fatal.
scr_manager.config.serialization.emit_defaults	Серилизация значений по умолчанию.
scr_manager.config.cache.db	Номер базы данных для доступа scr_manager.
scr_manager.config.cache.expire_time	TTL записей в кеше.
scr_manager.config.instances_settings.create_pipes	Создание пайпов/файлов во время работы инстансов.
scr_manager.config.instances_settings.status_check	Период опроса статусов работы инстансов.
scr_manager.config.instances_settings.fail_after	"Время для перевод инстанса в статус ""Failed"" при отсутствии на нем обработки данных."
scr_manager.config.instances_settings.proxy_settings.host	Хост прокси элемента.
scr_manager.config.instances_settings.proxy_settings.timeout	Таймаут прокси.
scr_manager.config.instances_settings.proxy.output_type	Тип прокси элемента.
scr_manager.config.instances_settings.output_dir	Папка для публикации.
scr_manager.config.instances_settings.docker_dir	Стандартная папка для хранения файлов, результатов работы docker контейнеров.
scr_manager.config.instances_settings.timeout	Таймаут.

scr_manager.config.instances_settings.ffprobe_timeout	Таймаут при работе с ffprobe.
scr_manager.config.instances_settings.skip_instance_deletion	Включение удаления инстансов с сервера после завершения задачи.
scr_manager.config.transcoder.sync_every	Период синхронизации транскодера.
scr_manager.config.packager.encryption_scheme	Схема шифрования.
scr_manager.config.packager.segment_duration	Время на которое делятся сегменты видео.
scr_manager.config.packager.chunk_list_size	Предельное кол-во сохраняемых чанков для live потока.
scr_manager.config.packager.key_rotation_period	Период ротации ключей.
scr_manager.config.packager.kms_timeout	Таймаут kms сервера.
scr_manager.config.packager.kms_url	Адрес kms сервера.
scr_manager.config.packager.ffmpeg_join_a_v	Флаг, отвечающий за сбор видео-, аудио-потоков в 1 файл. Актуален для профилей шифрования gs_aes и plain_hls.
scr_manager.config.publisher.update_period	Период обновления данных публишера.
scr_manager.config.publisher.status_timeout	Таймаут работы публишера.
scr_manager.config.publisher.max_workers_vod	Максимальное число процессов для публикации VOD контента.
scr_manager.config.publisher.max_workers_live	Максимальное число процессов для публикации LIVE контента.
scr_manager.config.publisher.delete_after_copy	Флаг, отвечающий за удаление уже скопированных чанков в исходных папках.
scr_manager.config.rtsp_process.segment_duration	Время на которое делятся сегменты видео.
scr_manager.config.rtsp_process.chunk_list_size	Предельное кол-во сохраняемых чанков для rtsp потока.

scr_manager.config.rtsp_process.wait_timeout	Таймаут для перехода rtsp задачи в статус In progress.
scr_manager.config.downloader.attempts	Кол-во попыток загрузки.
scr_manager.config.downloader.attempt_period	Время между попытками загрузки.
scr_manager.config.downloader.max_parallel	Кол-во одновременного загружаемых чанков.
scr_manager.config.downloader.playlist_timeout	Таймаут при скачивании плейлиста.
scr_manager.config.downloader.file_timeout	Таймаут при скачивании видеофайла.
scr_manager.config.downloader.segment_duration	Время сегментов загрузки.
scr_manager.config.support_server.ffprobe_timeout	Таймаут ffprobe
scr_manager.config.operator	Используемый оператор.
scr_manager.config.process_monitor.update_duration	Время обновлений.
scr_manager.config.pipeline.ping_interval	Интервал проверки состояния.
scr_manager.config.pipeline.attempts_number	Попыток проверки.
scr_manager.config.tasks_consumer.buffer_size	Число задач, которые будут отбираться из БД для исполнения за 1 цикл.
scr_manager.config.tasks_consumer.max_routines	Количество go-рутин для исполнения задач.
scr_manager.config.tasks_consumer.sleep_time	Время в секундах между циклами отбора задач для выполнения.
scr_manager.config.playlist_url.mounted_dir	Префикс примаунченной дирректории для составления итогового url мастер-плейлиста.
scr_manager.config.playlist_url.use_mounted_dir	Использования адреса примаунченной дирректории в ссылке на итоговый мастер-плейлист.

4.1.4.5. Описание параметров key_db

Ниже представлена таблица, которая описывает параметры, используемые key_db сервером при деплое и работе.

Параметр	Описание
key_db.enabled	Добавление сервиса в деплой.
key_db.ingress.enabled	Добавление ingress в деплой.
key_db.ingress.host.admin	Адрес ingress администратора.
key_db.system.password	Пароль, используемый для доступа.
key_db.system.timeout	Таймаут.
key_db.system.NodeIp	Ip адрес ноды, через которую будет осуществляться доступ.
key_db.system.NodePort	Порт ноды, через который будет осуществляться доступ.

4.1.4.6. Описание параметров ags

Ниже представлена таблица, которая описывает параметры, используемые ags сервером при деплое и работе.

Параметр	Значение по умолчанию	Описание
ags.enabled	true	Добавление сервиса в деплой
ags.replicas	2	Кол-во реплик пода
ags.hpa.enabled	true	Включение hpa
ags.hpa.minReplicas	2	Минимальное число hpa реплик
ags.hpa.maxReplicas	4	Максимальное число hpa реплик
ags.hpa.targetCPUUtilizationPercentage	25	Процент целевого использования CPU для hpa
ags.ingress.hostname	ags.testdrp.tz.cas	Адресс ingress'a сервиса
ags.ingress.enabled	true	Добавление ingress в деплой
ags.ingress.useHttps	true	Включение https ингреса
ags.ingress.annotations		Аннотации для ингресса
ags.resources.default_limits.cpu	128m	Ограничение сри для пода
ags.resources.default_limits.memory	128Mi	Ограничение памяти для пода
ags.resources.default_requests.cpu	96m	Запрос сри для пода
ags.resources.default_requests.memory	64Mi	Запрос памяти для пода
ags.system.mds.mr_address	am.testmds.cas.local	Адресс account manager'a
ags.system.mds.mr_protocol	http	Протокол связи с account manager'a
ags.system.mds.service_name	configManagerService	Название сервиса для работы с account manager'ом
ags.system.manager_host	scr-manager-svc	"Адресс manager сервиса. Если используется manager в качестве docker-контейнера, то требуется указать его адрес в офрмате "" <ip-сервера>:<port из SYSTEM_HTTP_ADDRESS> ""
ags.env.GOGC	'400'	Определяет отношение между процессором GC и памятью в GO

ags.env.GOMAXPROCS	'1'	Максимальное количество потоков GO
ags.config.Enabled	true	Использование конфига
ags.config.logger.level	debug	Степень логирования событий. Возможные значения: trace, debug, info, warning, error, fatal
ags.config.errors. api_not_allowed.code	4009	Код ошибки api_not_allowed
ags.config.errors. api_not_allowed.title	"	Заголовок ошибки api_not_allowed
ags.config.errors. api_not_allowed.detail	'API Not Found'	Детали об ошибке api_not_allowed
ags.config.errors. method_not_allowed.code	4010	Код ошибки method_not_allowed
ags.config.errors. method_not_allowed.title	Method Not Allowed	Заголовок ошибки method_not_allowed
ags.config.errors. method_not_allowed.detail	"	Детали об ошибке method_not_allowed
ags.config.errors.internal.code	4008	Код ошибки internal
ags.config.errors.internal.title	Внутренняя ошибка	Заголовок ошибки internal
ags.config.errors.internal.detail	"	Детали об ошибке internal

4.1.4.7. Описание параметров ags_web

Описание используемых параметров ags_web аналогично описанию, приведенному в разделе *Описание параметров ags* (вместо префикса ags используется префикс ags_web).

4.1.4.8. Описание параметров ems

Ниже представлена таблица, которая описывает параметры, используемые ems сервером при деплое и работе.

Параметр	Значение по умолчанию	Описание
ems.enabled	false	Добавление сервиса в деплой
ems.replicas	1	Кол-во реплик пода
ems.hpa.enabled	true	Включение hpa
ems.hpa.minReplicas	1	Минимальное число hpa реплик
ems.hpa.maxReplicas	3	Максимальное число hpa реплик
ems.hpa.targetCPUUtilizationPercentage	25	Процент целевого использования CPU для hpa
ems.ingress.hostname	ems.testshield.tz.cas	Адресс ingress'a сервиса
ems.ingress.enabled	false	Добавление ingress в деплой
ems.ingress.annotations		Аннотации для ингресса
ems.resources.limits.cpu	24m	Ограничение сри для пода
ems.resources.limits.memory	54Mi	Ограничение памяти для пода
ems.resources.requests.cpu	15m	Запрос сри для пода
ems.resources.requests.memory	48Mi	Запрос памяти для пода
ems.env.GOGC	'400'	Определяет отношение между процессором GC и памятью в GO
ems.env.GOMAXPROCS	'1'	Максимальное количество потоков GO
ems.configEnabled	true	Использование конфига
ems.config.logger.level	debug	Степень логирования событий. Возможные значения: trace, debug, info, warning, error, fatal
ems.config.errors.api_not_allowed.code	4004	Код ошибки api_not_allowed
ems.config.errors.api_not_allowed.title	"	Заголовок ошибки api_not_allowed

ems.config.errors. api_not_allowed.detail	'API Not Found'	Детали об ошибке api_not_allowed
ems.config.errors. method_not_allowed.code	4005	Код ошибки method_not_allowed
ems.config.errors. method_not_allowed.title	"	Заголовок ошибки method_not_allowed
ems.config.errors. method_not_allowed.detail	"	Детали об ошибке method_not_allowed
ems.config.errors.internal.code	4102	Код ошибки internal
ems.config.errors.internal.title	Нет связи с сервером. Попробуйте позже (код 20)	Заголовок ошибки internal
ems.config.errors.internal.detail	'Method Not Allowed'	Детали об ошибке internal

4.1.4.9. Описание параметров cron

Ниже представлена таблица, которая описывает параметры, используемые cron сервером при деплое и работе.


Параметр	Значение по умолчанию	Описание
cron.enabled	true	Добавление сервиса в деплой
cron.replicas	1	Кол-во реплик пода
cron.resources.limits. cpu	312m	Ограничение сри для пода
cron.resources.limits. memory	215Mi	Ограничение памяти для пода
cron.resources. requests.cpu	250m	Запрос сри для пода
cron.resources. requests.memory	196Mi	Запрос памяти для пода
cron.ingress. hostname	cron.testdrp.tz.cas	Адресс ingress'a сервиса
cron.ingress.enabled	true	Добавление ingress в деплой
cron.ingress. annotations	kubernetes.io/ingress. class: nginx	Аннотации для ингресса

cron.env. HTTP_ADDRESS	0.0.0.0:8080	Локальный адрес http сервера cron, который будет поднят в Docker контейнере
cron.configEnabled	true	Использование конфига
cron.config.logger. level	debug	Степень логирования событий. Возможные значения: trace, debug, info, warning, error, fatal

4.1.4.10. Описание параметров scrambler_ui

Ниже представлена таблица, которая описывает параметры, используемые scrambler_ui сервером при деплое и работе.

Параметр	Значение по умолчанию	Описание
scrambler_ui.enabled	true	Добавление сервиса в деплой
scrambler_ui.replicas	1	Кол-во реплик пода
scrambler_ui.ingress.host	web.testscrambler.tz.cas	Адрес ingress
scrambler_ui.ingress.annotations		Аннотации для ингресса
scrambler_ui.resources.limits.cpu	400m	Ограничение спу для пода
scrambler_ui.resources.limits. memory	256Mi	Ограничение памяти для пода
scrambler_ui.resources.requests.cpu	128m	Запрос спу для пода
scrambler_ui.resources.requests. memory	192Mi	Запрос памяти для пода
scrambler_ui.env.VITE_API_URL	http://ags-web.scrambler.dev.int	Адрес сервиса точки входа для web запросов
scrambler_ui.env. VITE_HAS_WATERMARK	false	Использование версии scrambler_ui для dremark

 **Внимание!** При установке системы версии 2.1.0.0 необходимо, чтобы значение параметра VITE_HAS_WATERMARK было установлено в "false".

4.1.4.11. Описание параметров секции db

В секции db представлены параметры, необходимые для установки kob с БД.

Параметр	Значение по умолчанию	Описание
db.db_mng	scr	Наименование базы данных Scrambler
db.db_mng	scr-errmap	Наименование базы данных Scrambler ошибок


db.address	11.111.11.111	Адрес БД
db.master_port	5432	Порт доступа до БД
db.timeout	50	Таймаут соединения к БД
db.max_conns	10	Максимальное количество соединений к БД

4.1.4.12. Настройка переменных окружения

Настройка переменных осуществляется в gitlab.

В боковом меню выбрать **Settings** (на панели слева) -> **CI/CD** -> **Environment variables**. Отредактировать переменные. Для корректной работы и развертывания системы должны быть заданы следующие переменные:

1. ERRMAPDB_LOGIN - логин для подключения к ErrMap DB.
2. ERRMAPDB_PASSWORD - пароль для подключения к ErrMap DB.
3. SCRDB_LOGIN - логин для подключения к SCR DB.
4. SCRDB_PASSWORD - пароль для подключения к SCR DB.
5. POSTGRES_LOGIN - имя администратора PostgreSQL БД.
6. POSTGRES_PASSWORD - пароль администратора PostgreSQL БД.
7. SCRAMBLER_CERT - сертификат для использования в scrambler_ui.
8. SCRAMBLER_KEY - ключ для использования scrambler_ui.

 Параметры `_LOGIN` и `_PASSWORD` задаются пользователем и используются при подключении к соответствующим базам данных.

4.2. Подготовка локального сервера для запуска Scrambler Task

4.2.1. Установка пакетов Docker

Способ установки docker-се определяется установщиком.

4.2.2. Добавление пользователя в группу docker

В базовом варианте установки Docker, команды для управления docker-cli выполняются от имени суперпользователя. Чтобы этого избежать необходимо добавить текущего пользователя (отличного от **root**) в группу docker:

```
sudo usermod -a -G docker <current_user>
```

Чтобы изменения вступили в силу требуется выполнить перезагрузку.

4.2.3. Установка nvidia-container-toolkit

Полный User guide представлен [здесь](#). Ниже будет представлена краткая последовательность действий:

```
distribution=$(. /etc/os-release;echo $ID$VERSION_ID) \  
&& curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo apt-key add - \  
&& curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-docker.list | sudo tee /etc/apt  
/sources.list.d/nvidia-docker.list \  
&& sudo apt-get update \  
&& sudo apt-get install -y nvidia-docker2 \  
&& sudo systemctl restart docker
```

❗ При появлении ошибки вида:

```
# Unsupported distribution!  
# Check https://nvidia.github.io/nvidia-docker
```

Необходимо посетить страницу с полным [User guide](#).

Перейти к пункту [Linux Distributions](#). И найти необходимый для установленной ОС **Identifier**.

После чего выполнить представленный выше команды, заменив **\$distribution** на **Identifier** установленной OS.

4.2.4. Обновление драйверов

Для корректной работы транскодирования на GPU на сервере для запуска Scrambler Task необходим драйвер видеокарты nvidia не ниже версии 520.56.06.

Чтобы уточнить версию уже установленного драйвера используется команда:

```
nvidia-smi
```

Пункт Driver Version. Если на сервере установлен драйвер младшей версии, то его необходимо обновить.

❗ Способ выполнения этой операции остается **на усмотрение заказчика/установщика**.

Ниже представлен пример обновления драйвера nvidia из PPA-репозитория.

Для начала необходимо добавить PPA-репозиторий в исходники системы:

```
sudo add-apt-repository ppa:graphics-drivers/ppa
```

Далее необходимо обновить его кеш используя команду apt:

```
sudo apt update
```

Перейти к поиску последней доступной версии драйвера nvidia:

```
apt search nvidia-driver
```

Затем необходимо установить последнее стабильное обновление драйверов для графики nvidia:

```
sudo apt install nvidia-driver-525
```

4.2.5. Запуск docker daemon

Docker-daemon используется для работы с docker через REST API. Данный сервис автоматически устанавливается вместе с другими пакетами Docker.

Для запуска docker-daemon на 2375 порту необходимо выполнить следующую команду:

```
sudo systemctl stop docker  
sudo dockerd -H unix:// -H 0.0.0.0:2375 &
```

Далее необходимо проверить, что docker-daemon на нужной машине запущен, путем выполнения curl запроса к host-машине для получения списка запущенных контейнеров:


```
curl --location 'http://<host-ip>:2375/containers/json'
```

В ответ должен вернуться пустой список "[]".


4.2.6. Mount папки output_dir

В процессе работы Scrambler сохраняет результаты работы в папке, наименование которой настраивается при помощи параметра **scr_manager.system.instances_settings.output_dir**, по умолчанию используется значение "publ".

Данная папка стандартно находится по пути "/var/lib/docker/volumes/publ/". В дальнейшем рекомендуется "маунтить" папку сразу со стандартной директорией "_data/".

 Данная директория на момент создания сервера может не существовать.

Чтобы артефакты работы Scrambler сохранялись в **CDN directory**. Необходимо произвести mount папки "publ" на host машине с папкой на удаленном CDN.

 Способ выполнения этой операции остается **на усмотрение заказчика/установщика**.

Ниже приведен пример команды для mount'a сетевой папки без использования аутентификации.

```
sudo mount -t cifs //<DOMAIN>/<MOUNT_DIR_PATH> /var/lib/docker/volumes/publ/_data/ -o users,sec=none
```

© ООО "Цифра", 2020-2024

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