

DRE Advertising Management Server

Руководство по установке

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

Содержание

1. Аннотация	3
2. Установка системы	4
2.1. Требования к квалификации установщика	4
2.2. Системные требования	4
2.2.1. Программное обеспечение	4
2.3. Предварительные действия	4
2.3.1. Установка и настройка Redis (open source)	5
2.4. Установка и настройка системы	5
2.4.1. Настройка переменных окружения CI/CD	5
2.4.2. Настройка S3 Minio	5
2.4.3. Настройка env переменных в .gitlab-ci.yml	6
2.4.4. Загрузка прав в Account Manager	6
2.4.4.1. Формирование ролей для работы с разделами WEB UI	7
2.4.5. Настройка production.yaml	7
2.4.6. Развертывание системы	7
2.4.6.1. Развертывание компонентов AGS	7
2.4.6.2. Развертывание AMS Streamer	8
3. Администрирование системы	10
3.1. Параметры конфигурации	10

1. Аннотация

Документ предназначен для технических специалистов, занимающихся установкой, настройкой и поддержкой системы DRE Advertising Management Server (далее по тексту - AMS). Документ рассчитан на инженеров, обладающих специальными навыками и знаниями в области программного обеспечения. Данный документ содержит руководство по установке и администрированию системы.

 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. Установка системы

2.1. 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.

2.2. Системные требования

2.2.1. 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.
 - а. Если пользователь выполнял развертывание Kubernetes самостоятельно, то он сам должен создать config file (см. документацию Kubernetes).
 - б. Если Kubernetes был развернут сторонними людьми, то необходимо получить config file у администратора кластера.
Внимание! Установкой и администрированием системы Kubernetes занимаются сотрудники Отдела поддержки инфраструктуры.
2. Установить kubectl (<https://kubernetes.io/docs/tasks/tools/install-kubectl/>).
3. Установить helm (<https://helm.sh/ru/docs/intro/install/>).
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.

Для корректной работы системы AMS необходим доступ к следующим ресурсам:

1. chartmuseum (ссылка и права доступа предоставляются по запросу заказчика).
2. gitlab (ссылка и права доступа предоставляются по запросу заказчика). Также нужен доступ к репозиторию, содержащему helmfile для развертывания AMS. 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).

2.3. Предварительные действия

2.3.1. Установка и настройка Redis (open source)

В соответствии с Системными требованиями, для корректной работы системы AMS требуется поднять базу данных Redis (open source). Все Redis (open source) должны быть доступны из кластера. Параметры работы с Redis (open source) настраиваются в конфигурационных файлах сервисов AMS. Конфигурацию самих Redis (open source) менять не требуется.

2.4. Установка и настройка системы

Установка описана в документе, ссылка и права доступа на который предоставляются по запросу заказчика.

2.4.1. Настройка переменных окружения CI/CD

Перед развертыванием стенда AMS, в репозитории стенда должны быть правильно настроены переменные окружения во вкладке "Settings -> CI/CD -> Variables".

Список переменных окружения предоставляются по запросу заказчика.

Важно учесть:

1. Если не указать собственные значения для переменных AMSDB_LOGIN и AMSDB_PASSWORD, то их значение по умолчанию будет сформировано автоматически по следующему правилу - `{{SIGN}}_ams`. Тут `{{SIGN}}` -- это переменная, задаваемая в файле `.gitlab-ci.yml`.
2. Переменная `DOMAIN_ZONE` отвечает за формирование ингрессов для сервисов продукта, в случае если они не определены в `production.yaml`.

В общем случае, автоматически сформированный ingress для сервиса будет выглядеть следующим образом -- `{{SERVICE_NAME}}-{{SIGN}}.{{DOMAIN_ZONE}}`, где:

- `{{SERVICE_NAME}}` -- название сервиса, статичное и не изменяемо.
- `{{SIGN}}` -- переменная, задаваемая в файле `.gitlab-ci.yml`.
- `{{DOMAIN_ZONE}}` -- переменная окружения CI/CD

Переменная `DOMAIN_ZONE` должна быть обязательно установлена в соответствии с настройками доступа к кластеру (по поводу значения этой переменной следует обратиться за консультацией к devops-специалистам).

2.4.2. Настройка S3 Minio

При настройке S3 Minio важно убедиться, что максимальный размер загружаемых видео-файлов -- до 100Мб.

Также в S3 Minio потребуется сгенерировать access-токен доступа, который нужно будет указать в переменных окружения GitLab, перед стартом работы сервисов.

Создавать бакеты (buckets) для сервисов не требуется, они будут созданы автоматически после начала работы системы. Необходимо учесть, что ingress-nginx контроллеры кластера должны быть настроены на трафик до 100Мб для каждого файла. Это нужно для возможности передать файлы в S3 Minio через кластер AMS.

Настройка для входящего трафика:

```
ingress_int:
  hostNetwork: true
  hostPort: true
  controller:
    config:
      proxy-body-size: 100m
```

Так же потребуется расширить ограничения трафика для ингресса S3 Minio в его проекте. Сделать это можно через соответствующую аннотацию ингресса:

```
annotations:
  nginx.ingress.kubernetes.io/proxy-body-size: 100m
```

2.4.3. Настройка env переменных в .gitlab-ci.yml

Перед разворачиванием стенда AMS, в репозитории стенда в файле .gitlab-ci.yml нужно определить некоторые переменные, по следующей структуре:

```
...
variables:
  SUBMODULE_DIRNAME: ams          #
  SUBMODULE_BRANCH:
    value: develop                #
  NAMESPACE: test-${CI_PROJECT_NAME} #
  DEPLOY_TYPE: single            #      easy, common  single
  HELMFILE_STRATAGY: sync
  CONCURRENCY: 20
  SIGN: dev-stand                #      .          .
...

```

Тут наиболее важны:

- NAMESPACE - название неймспейса, в который будет задеплоен стенд
- SIGN - уникальная подпись стенда, на основе которой будут сформированы ингрессы, названия БД и admin-пользователи БД.
- DEPLOY_TYPE - тип деплоя стенда. Допустимые значения [easy, common, single]

Более подробно о настройках деплоя можно почитать по ссылке, предоставляемой заказчику по запросу.

2.4.4. Загрузка прав в Account Manager

Для загрузки прав на ACM используется скрипт ams_load_permissions, добавленный в деплой AMS.

В общем случае требуется:

1. Перед запуском деплоя добавить в ACM сервис "amsaccess".
2. В production.yaml включить выполнение загрузки прав в секции:

```
ams_load_permissions:
  enabled: true
```

В секции `account_manager` должно быть указано актуальное значение в поле `address`:

```
account_manager:
  address: "am.testmds.cas.local"
  protocol: "http"
  service_name: "amsaccess"
```

3. Выполнить деплой и проверить логи выполнения `job` со скриптом `ams-load-permissions`. Загрузка прав всегда должна быть включена в `production.yaml`.

2.4.4.1. Формирование ролей для работы с разделами WEB UI

Для корректной работы со вкладками UI, при формировании ролей, следует учитывать следующее:

```
" " _      :
AMS UI |    () AMS UI |    ()
AMS UI |    ()
AMS UI |    ()
"";
  AMS UI |    () AMS UI |    ()
"";
AMS UI |    () AMS UI |    ()
"";
AMS UI |    () AMS UI |    ()
AMS UI |    ()
AMS UI |    ()
AMS UI |    ()
AMS UI |    ()
"" (  xlsx ):
AMS UI |
AMS UI |    ()
"";
AMS UI |    () AMS UI |    ()
"";
AMS UI |    () AMS UI |    ()
AMS UI |    ()
```

2.4.5. Настройка `production.yaml`

По умолчанию, в `default.yaml` вынесены все параметры конфигурации которые можно и/или нужно менять в своем `production.yaml`. Все остальное менять не рекомендуется.

Ссылка и права доступа к файлу `default.yaml` предоставляется заказчику по запросу.

2.4.6. Развертывание системы

В деплое AMS для переопределения любого параметра конфига любого сервиса нужно придерживаться следующей структуры:

```
{SERVICE_NAME}:
server:
  config:
    {ANY_CONFIG_STRUCTURE}
```

`{SERVICE_NAME}` – это название сервиса (например, "was") `{ANY_CONFIG_STRUCTURE}` – тут в соответствии со структурой конфигурации сервиса в JSON, с сохранением вложенности, можно переопределить любой параметр.

2.4.6.1. Развертывание компонентов AGS

В AMS используется компонент продукта AGS - WAS (интеграционная версия AGS - TBD). WAS используется для проксирования запросов с UI на AMS Server и ведения аудита действий пользователей.

WAS требуется включить в production.yaml для работы UI. Секция с was, выглядит следующим образом:

```
was:
  enabled: true
  server:
    config:
      storage:
        host: "storage.test.cas.local"
      system:
        audit:
          host: "audit.test.cas.local"
          enabled: true
```

Предполагается, что настраиваться будут только перечисленные поля, остальные выставлены по умолчанию.

Для отключения аудита достаточно задать `was.server.system.audit.enabled: false`.

Параметры `was.server.system.audit.host` и `was.server.storage.host` используются только при включенном аудите и должны быть заданы в соответствии с адресами компонентов AGS - Storage и Audit. Эти компоненты для AMS являются внешними.

Минимальная поддерживаемая версия AGS для интеграции с AMS - TBD

При необходимости можно изменить любой из параметров конфигурации WAS. Описание конфигурации для WAS можно найти в документации к продукту AGS ([ссылка](#) и [права доступа предоставляются по запросу заказчика](#)).

2.4.6.2. Развертывание AMS Streamer

В этом разделе описаны версии и действия для установки сервиса или нескольких сервисов `ams_streamer` с помощью Docker Compose. Данная инструкция проверена на машине с ОС:

```
No LSB modules are available.
Distributor ID: Debian
Description:    Debian GNU/Linux 11 (bullseye)
Release:       11
Codename:      bullseye
```

Версия Docker Engine Docker version 27.3.1, build ce12230. Для запуска `ams_streamer` нужно выполнить следующие действия:

1. Установить Docker Engine на Debian по гайду с официального сайта или выполнить команды ниже.
2. Выполнить команды:

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.
  docker.com/linux/debian \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update

# Install the Docker packages:
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

3. Скопировать в каталог файлы (контейнеры стримера и файл env) для запуска `ams_streamer`.
Контейнеры стримера (ссылки и права доступа предоставляются по запросу заказчика):
- Старый парк Сибирь
 - Старый парк Центр
 - Новый парк Сибирь
 - новый парк Центр
 - Файл env

 Контейнеры стримера нужно запускать поосле того как на S3 появятся бакеты `sb-sections-region-1` и `sb-sections-region-2`.
Данные бакеты создаются `ams_section_builder` после того как на WEB UI было добавлено расписание и для него начали появляться секции для вещания.
Если стример запустить до того как появятся бакеты `sb-sections-region-1` и `sb-sections-region-2`, AMS Streamer их не возьмет в работу и потребуется перезапуск контейнеров.

4. Перейти в каталог `cd`.
5. В файле `.yaml` нужно указать корректные данные для подключения, манифест и версию образа.
6. Выполнить в каталоге команду `sudo docker compose up -d` Для разветывания нескольких `ams_streamer` нужно создать под каждый стример отдельный `.yaml`-файл со своими конфигурациями и указать все названия `.yaml` через ":" в `.env`-файле. Если это сделать, то при выполнении команды `sudo docker compose up -d` будет поднято столько контейнеров стримеров, сколько было задано `.yaml`-файлов.

3. Администрирование системы

3.1. Параметры конфигурации

Параметры конфигурации сервисов AMS Server, AMS Media, AMS Task Tracker, AMS Sections Builder, AMS Scheduler, AMS Streamer, AMS UI предоставляются заказчику по запросу.

© ООО "Цифра", 2024-2025

Документация "DRE Advertising Management Server. Руководство по установке" является объектом авторского права. Воспроизведение всего произведения или любой его части воспрещается без письменного разрешения правообладателя