

ПО «Платформа «Хроника»»

Руководство по эксплуатации

Оглавление

Функциональные характеристики	3
1. Состав платформы.....	4
2. Установка системы.....	5
2.1 Системные требования	5
2.2 Используемые компоненты и языки.....	5
2.3 Установка	5
2.3.1 Скачивание конфигурации видеоядра.....	5
2.3.2 Установка Docker	6
2.3.3 Установка и настройка PostgreSQL	6
2.3.4 Установка и запуск Redis.....	7
2.3.5 Запуск streamer	7
2.3.6 Запуск CDN.....	7
2.3.7 Запуск cluster-manager	7
2.4 Конфигурация системы	8
2.4.1 Cluster-manager	8
2.4.2 CDN	9

Функциональные характеристики

- 1) Запись видеоархива с видеокамер;
- 2) Передача в live режиме и из архива видеофайлов
- 3) Управление кластером и его работой
- 4) Авторизация запросов на доступ С

1. Состав платформы

Система видеонаблюдения состоит из нескольких компонентов:

- streamer – компонент, записывающий архив с видеокамер;
- CDN – прокси, эффективно раздает live и архив с видеокамер;
- cluster-manager – управляет работой кластера;
- access-authorizer – вспомогательный компонент, отвечающий за авторизацию запросов на доступ к CDN. Предполагается, что клиент реализует свой сервис авторизации, соответствующий контракту <https://access-authorizer.cams.is74.ru/docs>

2. Установка системы

2.1 Системные требования

- Современная операционная система Linux, имеющая интерпретатор командной строки, рекомендуется «Debian 10».
- Docker 20.10.12+
- Docker-compose 2.2.2+
- Golang 1.17.5

Аппаратные требования:

1. Оперативная память: 8 ГБ
2. Свободное дисковое пространство: не менее 100 ГБ (зависит от объема данных)
3. Количество логических ядер процессора: 4
4. Частота процессора: 2.10 GHz

2.2 Используемые компоненты и языки

Название	Лицензия	Назначение
Docker	Apache 2.0	Программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений.
Golang	BSD	Для написания серверных модулей, которые получают, обрабатывают и отправляют информацию.
Redis	BSD	Используется как быстрое хранилище данных типа «ключ-значение».
PostgreSQL	BSD	База данных для хранения долгоживущей информации.
Javascript	LGPL v2.1	Используется как встраиваемый язык для программного доступа к объектам приложений.
Css	MIT	Формальный язык описания внешнего вида документа.

2.3 Установка

2.3.1 Скачивание конфигурации видеоядра

Для установки программного обеспечения на собственного сервера требуется произвести следующие действия:

- загрузить архив и сохранить на ПК (сервере), где планируется развернуть систему;
- распаковать его с помощью архиватора (7-zip, unzip, winrar и т.д), в желаемую папку на ПК или сервер, где планируется разворачивать и тестировать систему.

Необходимо создать на сервере следующие необходимые директории, в которых будут

располагаться конфигурации

Директории для хранения конфигурации приложений видеоядра и для хранения redis данных:

```
mkdir /var/server  
mkdir /var/server/redis-data
```

Директории для хранения видеофрагментов:

```
mkdir /var/spool/storage  
mkdir /var/spool/storage-live  
mkdir /var/spool/storage-ssd
```

Из скаченного архива, из папки videcore/var/server все содержимое надо переместить в системную папку /var/server на сервере:

```
mv videcore/server/* /var/server/
```

А также переместить файл с кронами в системную папку сервера:

```
mv videcore/etc/cron.d/cluster_manager /etc/cron.d/cluster_manager
```

2.3.2 Установка Docker

Инструкция по установке на сайте системы <https://docs.docker.com/engine/install>. Если разворачивание и тестирование системы будет происходить не от пользователя root, следует добавить локального пользователя в группу docker, для корректной работы:

```
sudo usermod -aG docker `whoami`
```

Также для того, чтобы скачать все необходимые docker образы приложений, требуется авторизоваться:

```
docker login registry.is74.ru
```

2.3.3 Установка и настройка PostgreSQL

Инструкция по установке PostgreSQL на сайте системы <https://www.postgresql.org/download/linux/debian/>

Небольшая настройка конфига:

- Делаем postgres доступным. В /etc/postgresql/17/main/postgresql.conf изменяем переменную listen_addresses на listen_addresses = '*'
- Разрешаем подключение к postgres из docker. Добавляем в /etc/postgresql/17/main/pg_hba.conf: host all all 0.0.0.0/0 md5.

Создание пользователей для приложений (пароли указать свои):

```
CREATE USER user_streamer WITH PASSWORD 'password';  
CREATE USER user_cluster_manager WITH PASSWORD 'password';
```

Создание БД для приложений:

```
CREATE DATABASE streamer WITH OWNER user_streamer;  
CREATE DATABASE "cluster-manager" WITH OWNER user_cluster_manager;
```

И еще запросы для настройки необходимых расширений:

```
-- Добавим расширение для генерации юидов --  
CREATE EXTENSION IF NOT EXISTS "uuid-oss" schema pg_catalog version  
"1.1";  
-- сбор статистики  
CREATE EXTENSION pg_stat_statements
```

2.3.4 Установка и запуск Redis

Заходим в папку /var/server на сервере, в ней находится docker compose файл с конфигурацией для запуска redis в docker. Перед запуском надо установить пароль для redis, для этого надо в файле docker-compose-redis.yml вместо <password> указать свой пароль

Выполняем команду для скачивания образа и запуска redis в docker:

```
docker compose --compatibility -f docker-compose-redis.yml pull
docker compose --compatibility -f docker-compose-redis.yml up -d
redis
```

2.3.5 Запуск streamer

Заходим в папку /var/server на сервере, в ней находится docker compose файл с конфигурацией для запуска streamer в docker.

Перед запуском надо указать пароль для бд данного приложения в postgres, с которым создавали пользователя, для этого надо в файле docker-compose-streamer.yml и в configs/streamer.yml вместо <password> указать пароль для пользователя.

Выполняем команду для скачивания образа и запуска streamer в docker:

```
docker compose --compatibility -f docker-compose-streamer.yml pull
streamer
docker compose --compatibility -f docker-compose-streamer.yml up -d
streamer
```

При запуске сервиса автоматически накатываются необходимые миграции в БД

2.3.6 Запуск CDN

Заходим в папку /var/server на сервере, в ней находится docker compose файл с конфигурацией для запуска CDN в docker.

Перед запуском надо указать пароль для redis который указали при запуске redis, а так же указать хост для access-authorizer, для этого надо в файле configs/cdn.yml в REDIS.PASSWORD указать пароль от redis и вместо в ACCESS_AUTHORIZER.HOST указать хост до access-authorizer.

Выполняем команду для скачивания образа и запуска cdn в docker:

```
docker compose --compatibility -f docker-compose-cdn.yml pull cdn
docker compose --compatibility -f docker-compose-cdn.yml up -d cdn
```

2.3.7 Запуск cluster-manager

Предполагается, что cluster-manager разворачивается в kubernetes. Но в качестве примера развертывания предлагается запускать просто в docker.

Заходим в папку /var/server на сервере, в ней находится docker compose файл с конфигурацией для запуска cluster-manager в docker.

Перед запуском надо указать пароль для бд данного приложения в postgres, с которым создавали пользователя, для этого надо в файле docker-compose-cluster-manager.yml и в configs/cluster-manager.yml вместо <password> указать пароль для пользователя.

Выполняем команду для скачивания образа и запуска cluster-manager http сервера и необходимых воркеров и крон в docker:

```
docker compose --compatibility -f docker-compose-cluster-manager.yml
pull
docker compose --compatibility -f docker-compose-cluster-manager.yml
up -d cluster-manager-http
docker compose --compatibility -f docker-compose-cluster-manager.yml
up -d cluster-manager-camera-set-server
docker compose --compatibility -f docker-compose-cluster-manager.yml
up -d cluster-manager-cron
```

При запуске сервиса автоматически накатываются необходимые миграции в БД

2.4 Конфигурация системы

2.4.1 Cluster-manager

Добавляем роль сервера:

```
curl --location 'http://<host_server>:8002/server-role' \
--header 'Content-Type: application/json' \
--data '{
  "id": "default",
  "name": "default servers"
}'
```

Добавляем сервер:

```
curl --location 'http://<host_server>:8002/server' \
--header 'Content-Type: application/json' \
--data '{
  "name": "Видеосервер 1",
  "domain": "dev-video-streamer1-chel6.is74.ru",
  "role": [
    "default"
  ],
  "cameraCountCap": 100,
  "isActive": true,
  "id": 1
}'
```

Добавляем словарь шаблона времени хранения

```
curl --location 'http://<host_server>:8002/storage-template/' \
--header 'Content-Type: application/json' \
--data '{
  "name": "3 дня",
  "archiveMin": 1,
  "archiveMax": 3,
  "deleted": false,
  "serverRole": "default"
}'
```

Добавляем словарь кодек:

```
curl --location 'http://<host_server>:8002/codec' \
--header 'Content-Type: application/json' \
--data '{
  "name": "Default",
  "audioParams": "-c:a aac -ac 1 -b:a 64k -ar 8000",
  "preParams": "-threads 1",
  "videoParams": "-c:v copy"
}'
```

Идентификатор созданной записи смотрим в базе. Добавляем шаблон хранения:

```
curl --location 'http://<host_server>:8002/source-template' \
--header 'Content-Type: application/json' \
--data-raw '{
  "name": "Beward",
  "audioEnabled": true,
  "isOnvif": false,
  "mainCodecID": "ac3d2623-b74b-4b81-9a89-e3e114182bbd",
  "mainEnabled": true,
  "mainURLPattern": "rtsp://#LOGIN#:#PASSWORD#@#IP#:#PORT#/av0_0",
  "subCodecID": "ac3d2623-b74b-4b81-9a89-e3e114182bbd",
  "subEnabled": true,
  "subURLPattern": "rtsp://#LOGIN#:#PASSWORD#@#IP#:#PORT#/av0_1"
}'
```

Идентификатор созданной записи смотрим в базе. Добавляем камеру:

```
curl --location 'http://<host_server>:8002/camera' \
--header 'Content-Type: application/json' \
--data '{
  "host": "18688230104D.cctv.is74.ru",
  "login": "admin",
  "password": "admin",
  "mediaPort": 554,
  "onvifPort": 80,
  "sourceTemplateID": "d99fc1ef-92ea-41de-9c2d-c140b4a1f855",
  "storageTemplateID": "05169233-2712-46ed-b7e6-558d9e69a094",
  "serverID": 1,
  "serverRole": "default",
  "audioEnabled": true,
  "activeUntil": "9999-12-30 09:00:00.000000",
  "deleted": false
}'
```

2.4.2 CDN

CDN - прокси, эффективно раздает live и архив с видеокамер.

Генерация токена доступа и воспроизведение через CDN. Обращаемся к access-authorizer:

```
curl --location 'http://<host_access_authorizer>/access/create' \
--header 'Content-Type: application/json' \
--data '{
  "items": [
```

```
{
  "cameraUUID": "20c896c7-4048-4e2c-8886-98bee17302ad",
  "permissions": {
    "targets": [
      "hls-live",
      "snapshot-live"
    ]
  }
},
"ttl": "2023-05-31T00:15:00+05:00",
"userIdentity": "user-id=123"
}'
```

Затем с вернувшимся токеном можно обращаться к CDN:

- `http://<host_server>:8004 /hls/playlists/ts.m3u8?uuid=20c896c7-4048-4e2c-8886-98bee17302ad&quality=main&token=bearer-jE6py4Oezr8psTDyiQ6lZFxt6Ke7Y9OM`
- [http:// <host_server>:8004 /snapshot?uuid=20c896c7-4048-4e2c-8886-98bee17302ad&token=bearer-jE6py4Oezr8psTDyiQ6lZFxt6Ke7Y9OM](http://<host_server>:8004 /snapshot?uuid=20c896c7-4048-4e2c-8886-98bee17302ad&token=bearer-jE6py4Oezr8psTDyiQ6lZFxt6Ke7Y9OM)