

Программный комплекс «Единая Медиаплатформа»

Инструкция по эксплуатации

На 56 листах

Москва, 2024

Аннотация

Настоящий документ описывает процессы, обеспечивающие эксплуатацию программного комплекса «Единая Медиаплатформа» (далее – Медиаплатформа).

Оглавление

1	Используемые термины и определения.....	5
2	Назначение Медиаплатформы.....	11
3	Порядок взаимодействия пользователя с системой и сервисами Медиаплатформы	12
3.1	Внутренняя система управления задачами	12
3.1.1	Регистрация клиента.....	12
3.1.2	Получение JWT-токена для клиента.....	12
3.1.3	Создание пресета	13
3.1.4	Создание задачи на обработку файла	18
3.1.5	Проверка токена.....	25
3.2	Система загрузки аудиовизуального контента в сети Интернет.....	26
3.2.1	Загрузка одним или несколькими файлами	26
3.2.2	Загрузка чанками файлов	27
3.2.3	Загрузка путем скачивания файла с FTP-сервера.....	27
3.2.4	Загрузка путем скачивания файла с HTTP-сервера.....	28
3.2.5	Многопоточная загрузка	28
3.2.6	Удаление исходных файлов после загрузки	28
3.3	Микросервис диагностики файлов аудиовизуального контента	29
3.3.1	Диагностика загружаемых файлов.....	29
3.4	Система транскодирования аудиовизуального контента	32
3.4.1	Настройка профилей транскодирования	35
3.4.2	Обрезка черных полос (каше).....	36
3.4.3	Выравнивание громкости.....	36
3.4.4	Нарезка и склейка контента.....	36
3.4.5	Прикрепление заставки к исходному контенту	37
3.4.6	Поддержка субтитров в формате Teletext и DVB subtitles	38
3.4.7	Сохранение результатов транскодирования	39
3.5	Модуль трансляции аудиовизуального контента	43
3.5.1	Деинтерлейсинг видеовещания	44
3.5.2	Наложение логотипа на видеовещание	46
4	Служебные функции Медиаплатформы.....	48
4.1	Логирование действий и событий.....	48
4.2	Мониторинг	49
5	Действия при сбоях в функционировании Медиаплатформы	51
5.1	Профилактические действия для предупреждения сбоев в работе Медиаплатформы ...	51

5.2 Основные положения для исследования проблем, связанных с работой Медиаплатформы.....	51
5.3 Известные проблемы в работе Медиаплатформы и методы их решения	52
5.4 Действия при неустановленной проблеме	55
5.4.1 Остановка и старт Медиаплатформы	55
5.4.2 Остановка Медиаплатформы.....	55
5.4.3 Повторный запуск Медиаплатформы.....	55
5.4.4 Откат установки.....	55

1 Используемые термины и определения

Термин	Определение
API	Программный интерфейс, описание способов взаимодействия одной компьютерной программы с другими
Base64	Стандарт кодирования двоичных данных при помощи только 64 символов ASCII. Алфавит кодирования содержит латинские символы A-Z, a-z и цифры 0-9
Bucket	Логическая сущность, которая помогает организовать хранение медиафайлов в хранилище, например, в S3
CPU	Центральное процессорное устройство (или центральное обрабатывающее устройство) — электронный блок либо интегральная схема, исполняющая машинные инструкции (код программ), главная часть аппаратного обеспечения компьютера или программируемого логического контроллера
CRF	Метод сжатия видео, который обеспечивает максимальное качество при минимальном размере файла. Сжатие CRF основано на контроле качества вместо контроля битрейта, что делает его более эффективным в сравнении с другими кодеками. Он позволяет управлять размером файла при сохранении качества
CUDA	Программно-аппаратная архитектура параллельных вычислений, которая позволяет существенно увеличить вычислительную производительность благодаря использованию графических процессоров фирмы Nvidia
DAR	Отношение ширины к высоте изображения, соотношение сторон, для телевидения DAR традиционно составлял 4: 3 (он же полноэкранный режим), стандартом для HDTV является 16: 9 (он же широкоэкранный режим)
FFmpeg	Набор свободных библиотек (сборников подпрограмм или объектов, используемых для разработки программного обеспечения) с открытым исходным программным кодом, которые позволяют записывать, конвертировать и передавать цифровые аудио, видеозаписи в различных форматах

FTP	Протокол передачи файлов по сети
FileHear	Объектное хранилище, облачный сервис, позволяющий хранить файлы любого типа и объема. Хранилище используется для сохранения исходных, загруженных пользователем и обработанных Медиаплатформой материалов
FPS	Количество кадров в секунду в видеофайле
GPU	Графический процессор; отдельное устройство персонального компьютера, ноутбука, выполняющее графический рендеринг
Grafana	Аналитическая платформа с открытым исходным кодом, которая позволяет опрашивать и визуализировать данные, отправлять предупреждения и просматривать метрики программного обеспечения независимо от того, где они хранятся
HLS	Коммуникационный протокол для потоковой передачи медиа на основе HTTP
HTTP	Сетевой протокол прикладного уровня передачи данных
HTTPS	Расширение протокола HTTP для поддержки шифрования в целях повышения безопасности
IP-адрес	Уникальный числовой идентификатор устройства в компьютерной сети, работающей по протоколу IP
JWT-токен	Открытый стандарт для создания токенов доступа, основанный на формате JSON. Токены создаются сервером, подписываются секретным ключом и передаются клиенту, который в дальнейшем использует данный токен для подтверждения подлинности аккаунта
Kafka	Распределённый программный брокер сообщений с открытым исходным кодом, разрабатываемый в рамках фонда Apache на языках Java и Scala
Kubernetes	Открытое программное обеспечение для оркестровки контейнеризированных приложений: автоматизации их развёртывания, масштабирования и координации в условиях кластера

MD5	Алгоритм хеширования (преобразования по определённому алгоритму массива данных), предназначенный для создания контрольных сумм или «отпечатков» сообщения произвольной длины и последующей проверки их подлинности
MIME	Спецификация для передачи по сети файлов различного типа: изображений, музыки, текстов, видео, архивов и др. Указание MIME-типа используется в HTML обычно при передаче данных форм и вставки на страницу различных объектов
Nvidia	Разработчик графических процессоров и систем на чипах
REF	Параметр, указывающий количество предыдущих кадров, на которых может базироваться текущий кадр. Большое количество ссылочных кадров может улучшить качество выходного видео, но также может увеличить размер файла и требования к производительности при декодировании и кодировании видео
RTMP	Проприетарный протокол потоковой передачи данных, в основном используемый для передачи потокового видео и аудиопотоков с веб-камер через интернет
S3	Объектное хранилище для хранения больших объемов данных в исходном формате без иерархии и разбивки на отдельные каталоги без ограничений по масштабированию
SAR	Отношение количества пикселей по горизонтали к количеству пикселей по вертикали (эти цифры указаны в разрешении видеофайла)
SRT	Транспортный протокол, предназначенный для потоковой передачи видео, который гарантирует минимальную задержку и высокую степень надежности
Stateless	Архитектурный подход, при котором данные о прошлых взаимодействиях с клиентом не сохраняются на сервере. Подразумевает, что в каждом запросе должна содержаться полная информация для его корректной обработки
Swagger UI	Набор ресурсов в виде HTML, JavaScript и CSS, которые динамически генерируют документацию из API

Trickmode	Функция показа соответствующего эскиза (thumbnail'a) во время движения мышки на полосе прокрутки видеофайла
Upscale	Масштабирование видео до более высокого разрешения
URI	Унифицированный указатель ресурса; система унифицированных адресов электронных ресурсов, или единообразный определитель местонахождения ресурса (файла)
URL	Адрес, который выдан уникальному ресурсу в сети Интернет
VLC	Свободный кроссплатформенный медиапроигрыватель, функционирующий на большинстве современных операционных систем и мобильных платформ
БД	База данных
Битрейт	Количество бит, используемых для передачи, обработки данных в единицу времени, используется при измерении эффективной скорости передачи потока данных по каналу — минимального размера канала, который сможет пропустить этот поток без задержек
Валидация	Доказательство удовлетворения требованиям конкретного пользователя, продукта, услуги или системы
Деинтерлейсинг	Процесс совмещения чётных и нечётных строк чересстрочного формата для создания одного кадра из двух чересстрочных и дальнейшего вывода на экран с прогрессивной развёрткой, такой как компьютерный монитор
Декодирование	Процесс, при котором задача декодирования видеофайла (например, кодированного в формате H.264 или H.265) перекладывается на специализированные аппаратные компоненты в видеокарте или процессоре
Демультимплексирование	Процесс разделения и направления отдельных потоков данных, объединенных для передачи по общему каналу связи или среде
Деплой	Развертывание и запуск системы или сайта в его рабочей среде

Дискретизация	Процесс преобразования аналогового сигнала в цифровую форму для обработки данных
Каше	Технология согласования соотношения сторон экрана, не требующая обрезки исходного изображения
Кодек	Программа, выполняющая преобразование данных аудио и/или видеосигнала
Кодирование	Преобразование файлов определенного типа из одного формата в другой, более подходящий для конкретных целей, или/и изменение некоторых спецификаций цифрового потока
Контейнер	Формат файла, определяющей распределение аудио, видео, а в некоторых случаях и текстовой информации внутри него
Медиафайл	Компьютерный файл, содержащий аудио- и видеoinформацию
Метаданные	Информация о другой информации, или данные, относящиеся к дополнительной информации о содержимом или объекте
Мультиплексирование	Уплотнение канала связи; передача нескольких потоков данных с меньшей скоростью по одному каналу связи путем создания в исходном канале связи нескольких подканалов связи с меньшей пропускной способностью
Под	В Kubernetes, запрос на запуск одного или более контейнеров на одном узле
Стрим	Прямой эфир на интернет-сервисе
Топик	В Kafka, способ группировки потоков сообщений по категориям. Производители публикуют сообщения определенной категории в топике, а потребители подписываются на этот топик и читают из него сообщения
Транскодирование	Обработка медиафайлов, прямое цифровое преобразование одной кодировки в другую
Формат пикселя	Способ, которым графические карты обрабатывают информацию о цвете и яркости каждого пикселя на экране

Цветовое пространство	Модель кодировки цвета; палитра разных оттенков, где каждый цвет имеет определённые координаты по принципу "точки на карте"
Чанк	Часть медиафайла для удобства его загрузки и обработки. Обычно файл делится на чанки одинакового размера, кроме последнего чанка

2 Назначение Медиaplatformы

Программный комплекс «Единая Медиaplatforma» является интегрированной системой обработки медиаконтента.

Медиaplatforma позволяет:

- Анализировать загруженный медиаконтент и оценивать его качество;
- Обрабатывать медиаконтент, сохранять его в различном качестве и форматах;
- Транслировать медиаконтент в различных качествах;
- Создавать серии эскизов для ознакомительного просмотра;
- Осуществлять прочие функции по форматированию аудио- и видеофайлов.

3 Порядок взаимодействия пользователя с системой и сервисами Медиаплатформы

3.1 Внутренняя система управления задачами

Взаимодействие пользователя с внутренней системой управления задачами Медиаплатформы предусматривает последовательное выполнение ряда действий, описанных ниже и включающих в себя:

1. Регистрацию клиента;
2. Получение JWT-токена для клиента;
3. Создание пресета;
4. Создание задачи на обработку файла;
5. Проверку токена.

3.1.1 Регистрация клиента

Для регистрации клиента необходимо выполнить следующие действия:

1. Пробросить порт с сервиса авторизации Медиаплатформы для доступа к API сервиса;
2. Создать ключ для аутентификации администратора;
3. Пробросить порт к БД;
4. Отправить запрос на регистрацию клиента в следующем формате:

```
curl --location 'http://localhost:<insert_port>/api/v1/admin/account' \  
--header 'accept: application/json' \  
--header 'x-generator-key: <insert_admin_key>' \  
--header 'Content-Type: application/json' \  
--data '{  
  "client_id": "1",  
  "client_name": "EVP"  
'  
--client_id Идентификатор клиента/партнёра  
--client_name Наименование клиента/партнёра
```

5. Получить ключ API в следующем формате:

```
{  
  "key": "c59fe1bb-702a-43de-9d73-b43696e4ed64"  
}
```

6. Выполнить SQL-запрос в БД `mp_auth` и проверить, что в таблице создана запись для клиента 1 EVP;

7. Выполнить SQL-запрос в БД `mp_auth` и проверить, что для клиента был создан ключ API. Ключ должен соответствовать полученному в ответе из шага 4.

3.1.2 Получение JWT-токена для клиента

Для получения JWT-токена необходимо выполнить следующие действия:

1. Пробросить порт с сервиса авторизации Медиаплатформы для доступа к API сервиса;
2. Создать клиент и получить ключ API;
3. Пробросить порт к БД;
4. Отправить запрос на генерацию JWT-токена в следующем формате:

```
curl --location --request POST 'http://localhost:58498/api/v1/auth/token' \  
--header 'accept: application/json' \  
--header 'x-api-key: <insert_api_key>' \  
--header 'x-api-key' Ключ API
```

5. При успешном запросе будет получен JWT-токен в следующем формате:

```
{  
  "t": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
eyJjbGllbnRfaWQiOiJxIiwiaXNjaXhwIjozNzE4LCJpYXQiOiE3MTAyNjYzMTh9.  
veqM2dXq_Rp5Z7DQds8jwkdPDUv3IK2JIVKZJFqrOrM"  
}
```

6. Выполнить SQL-запрос в БД mp_auth и проверить, что в таблице создана запись с токеном для клиента 1 EVP. Токен должен соответствовать полученному в ответе из шага 4.

3.1.3 Создание пресета

Для создания пресета необходимо выполнить следующие действия:

1. Пробросить порт с сервиса авторизации Медиаплатформы для доступа к API сервиса;
2. Пробросить порт с сервиса управления задачами для доступа к API сервиса;
3. Зарегистрировать клиент;
4. Получить ключ API;
5. Получить JWT-токен. Убедиться, что токен валидный, в том числе по сроку действия;
6. Пробросить порт к БД;
7. Отправить запрос на создание пресета в следующем формате:

```
curl --location 'http://localhost:56422/v1/settings' \  
--header 'accept: application/json' \  
--header 'Content-Type: application/json' \  
--header 'Authorization: Bearer <insert_jwt_token>' \  
--data '{  
  "preset": {  
    "message_type": "TASKMANAGER_SETTINGS",  
    "ffmpeg_decoding_options": {  
      "threads": "1"  
    }  
  }  
}'
```

```
},
"master_playlist": true,
"hw_device": "cuda",
"profiles": [
  {
    "name": "high",
    "container_type": "hls",
    "filename_extension": "m3u8",
    "pix_bit_depth": 8,
    "video": {
      "stream_selection": "best",
      "codec": "libx264",
      "filter_options": {
        "width": 1920,
        "height": 1080,
        "overlay": false,
        "padding": false,
        "rotate": true,
        "allow_swap_dimensions": true,
        "pix_fmt": "yuv420p"
      },
      "ffmpeg_encoding_options": {
        "g": "60",
        "threads": "4",
        "profile": "main",
        "preset": "superfast",
        "crf": "23",
        "maxrate": "8M",
        "bufsize": "4M"
      }
    },
    "audio": {
      "stream_selection": "best",
      "codec": "aac",
      "filter_options": {
```

```
    "channels": 2,
    "samplerate": 48000
  },
  "ffmpeg_encoding_options": {
    "b": "256k"
  }
},
"cpu": {
  "name": "high",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 8,
  "video": {
    "stream_selection": "best",
    "codec": "libx264",
    "filter_options": {
      "width": 1920,
      "height": 1080,
      "overlay": false,
      "padding": false,
      "rotate": true,
      "allow_swap_dimensions": true,
      "pix_fmt": "yuv420p"
    },
    "ffmpeg_encoding_options": {
      "g": "60",
      "threads": "4",
      "profile": "main",
      "preset": "superfast",
      "crf": "23",
      "maxrate": "8M",
      "bufsize": "4M"
    }
  }
},
"audio": {
```

```
"stream_selection": "best",
"codec": "aac",
"filter_options": {
  "channels": 2,
  "samplerate": 48000
},
"ffmpeg_encoding_options": {
  "b": "256k"
}
},
"gpu": {
  "name": "high",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 8,
  "video": {
    "stream_selection": "best",
    "codec": "h264_nvenc",
    "filter_options": {
      "width": 1920,
      "height": 1080,
      "overlay": false,
      "padding": false,
      "rotate": true,
      "allow_swap_dimensions": true
    },
    "ffmpeg_encoding_options": {
      "g": "60",
      "threads": "1",
      "profile": "main",
      "preset": "p1",
      "b": "8388608",
      "bufsize": "8388608",
      "maxrate": "8388608",
```

```
        "tune": "ull",
        "refs": "1",
        "no-scenecut": "1",
        "coder": "cabac"
    }
},
"audio": {
    "stream_selection": "best",
    "codec": "aac",
    "filter_options": {
        "channels": 2,
        "samplerate": 48000
    },
    "ffmpeg_encoding_options": {
        "b": "256k"
    }
}
}
],
"thumbnail": {
    "number": 1,
    "offset": 10,
    "step": 0,
    "width": 1280,
    "height": 720,
    "quality": 2,
    "padding": false,
    "rotate": true,
    "allow_swap_dimensions": true
},
"trickmode": {
    "frame_width": 128,
    "frame_height": 72,
    "columns": 10,
```

```
"rows": 10,
"frames": 100,
"quality": 8
},
"probe": {
  "check_demux": true,
  "check_decode": false
}
},
"preset_id": "auth_test"
}' Получен ответ 200 ОК.
Body:
"ОК"
```

8. Выполнить SQL-запрос в БД mp_taskmanager и проверить, что в таблице создана запись с новым пресетом для клиента EVP. Пресет должен быть привязан к клиенту EVP, id 1.

3.1.4 Создание задачи на обработку файла

Для создания задачи на обработку файла необходимо выполнить следующие действия:

1. Получить JWT-токен. Убедиться, что токен валидный, в том числе по сроку действия;
2. Создать пресет для клиента;
3. Отправить запрос на создание задачи на обработку в следующем формате:

```
curl --location 'https://uploader.dev.media.cyrm.ru/mp-stage/upload/v1/task' \
--header 'accept: application/json' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer <insert_jwt_token>' \
--data '{
  "request_id": "insert_unique_uuid",
  "source": {
    "file_uri": "http://ftphhttp/test/original.mp4"
  },
  "preset_id": "auth_test"
}' Получен ответ 200 ОК.
```

Пример body:

```
{
  "TaskID": "d2a3e5f8-5119-4f98-a3b8-16be937b979e",
```

```
"Status": "Created"
}
```

4. Проверить наличие сообщения о создании задачи на скачивание в топике Kafka taskmanager в следующем формате:

```
{
  "message_type": "DOWNLOAD_TASK_CREATE",
  "task_id": "d2a3e5f8-5119-4f98-a3b8-16be937b979e",
  "priority": 0,
  "client_id": "1",
  "preset_id": "auth_test",
  "file": {
    "original_filename": "http://ftphhttp/test/original.mp4",
    "size": 0,
    "local_path": ""
  },
  "metadata": {},
  "device": {
    "client_ip": "",
    "user_agent": ""
  },
  "errors": []
}
```

5. Проверить статус загрузки файла в топике Kafka taskmanager. Исходный файл должен быть загружен в хранилище S3.

6. В топике Kafka taskmanager должен быть зафиксирован статус об успешной загрузке в следующем формате:

```
{
  "message_type": "TRANSFER_TASK_FINISHED",
  "error": "",
  "status": "OK",
  "task_id": "d2a3e5f8-5119-4f98-a3b8-16be937b979e",
  "source": {
    "provider": "s3",
    "bucket": "media-platform-dev-issues",
    "s3_region": "ru-central1",
  }
}
```

```

"host_uri": "https://storage.yandexcloud.net",
"file_uri": "d2a3e5f8-5119-4f98-a3b8-16be937b979e/original.mp4"
},
"destination": {
  "provider": "s3",
  "bucket": "media-platform-dev-issues",
  "s3_region": "ru-central1",
  "host_uri": "https://storage.yandexcloud.net",
  "file_uri": "d2a3e5f8-5119-4f98-a3b8-16be937b979e/original.mp4"
},
"telemetry_trace_context": {
  "trace_id": "",
  "parent_span_id": ""
},
"errors": []
}

```

7. Убедиться, что файл прошел обработку на разрешенный формат. В топике Kafka taskmanager должен быть зафиксирован статус об успешной обработке сервисом Медиаплатформы в следующем формате:

```

{
  "message_type": "PROBE_TASK_FINISHED",
  "task_id": "d2a3e5f8-5119-4f98-a3b8-16be937b979e",
  "status": "OK",
  "errors": [],
  "location": {
    "provider": "s3",
    "bucket": "media-platform-dev-issues",
    "s3_region": "ru-central1",
    "host_uri": "https://storage.yandexcloud.net",
    "file_uri": "d2a3e5f8-5119-4f98-a3b8-16be937b979e/original.mp4"
  },
  "file_size": 40188057,
  "container_short_names": "mov,mp4,m4a,3gp,3g2,mj2",
  "streams": [
    {

```

```
"stream_type": "VIDEO",
"stream_idx": 0,
"start": 0.0,
"duration": 31.64,
"codec": "hevc",
"bitrate": 10012978,
"width": 1920,
"height": 1080,
"pix_fmt": "yuv420p10le",
"sample_aspect_ratio": {
  "num": 1,
  "den": 1
},
"display_aspect_ratio": {
  "num": 16,
  "den": 9
},
"frame_rate": {
  "num": 25,
  "den": 1
},
"color_range": "tv",
"color_primaries": "bt709",
"color_trc": "bt709",
"color_space": "bt709",
"chroma_location": "left"
},
{
"stream_type": "AUDIO",
"stream_idx": 1,
"start": 0.0,
"duration": 31.68,
"codec": "aac",
"bitrate": 128175,
"samplerate": 48000,
```

```
"channels": 2
}
],
"best_video_stream": 0,
"best_audio_stream": 1,
"demux_ok": true,
"start_timestamp_ms": 1710267523103,
"duration_us": 1564653
}
```

8. Убедиться, что файл успешно прошел транскодирование. В топике Kafka taskmanager должен быть зафиксирован статус об успешном выполнении обработки файла в следующем формате:

```
{
"message_type": "TRANSCODER_TASK_FINISHED",
"task_id": "d2a3e5f8-5119-4f98-a3b8-16be937b979e_pri0",
"status": "OK",
"errors": [],
"files": [
{
"profile_name": "high",
"location": {
"provider": "s3",
"bucket": "media-platform-dev-issues",
"s3_region": "ru-central1",
"host_uri": "https://storage.yandexcloud.net",
"file_uri": "d2a3e5f8-5119-4f98-a3b8-16be937b979e_pri0/high.m3u8"
},
"metadata": {
"mime_type": "application/vnd.apple.mpegurl",
"file_size": 1154,
"duration": 31.68000030517578,
"frame_size": {
"width": 1920,
"height": 1080,
"display_aspect_ratio": {
```

```
    "num": 16,  
    "den": 9  
  }  
}  
,  
"number_of_video_streams": 1,  
"number_of_audio_streams": 1,  
"container": "hls",  
"streams": [  
  {  
    "stream_type": "video",  
    "bitrate": 8443744,  
    "codec": "h264_nvenc",  
    "frame_size": {  
      "width": 1920,  
      "height": 1080,  
      "display_aspect_ratio": {  
        "num": 16,  
        "den": 9  
      }  
    },  
    "frame_rate": {  
      "num": 25,  
      "den": 1  
    }  
  },  
  {  
    "stream_type": "audio",  
    "bitrate": 256704,  
    "codec": "aac",  
    "samplerate": 48000,  
    "channels": 2  
  }  
],  
"watermark": false
```

```

    }
  ],
  "master_playlist": {
    "location": {
      "provider": "s3",
      "bucket": "media-platform-dev-issues",
      "s3_region": "ru-central1",
      "host_uri": "https://storage.yandexcloud.net",
      "file_uri": "d2a3e5f8-5119-4f98-a3b8-16be937b979e_pri0/master.m3u8"
    }
  },
  "trickmode": {
    "mime_type": "image/jpeg",
    "canvas_width": 1280,
    "canvas_height": 720,
    "rows": 10,
    "columns": 10,
    "frames": 100,
    "step": 0.3163999915122986,
    "files": [
      {
        "provider": "s3",
        "bucket": "media-platform-dev-issues",
        "s3_region": "ru-central1",
        "host_uri": "https://storage.yandexcloud.net",
        "file_uri": "d2a3e5f8-5119-4f98-a3b8-16be937b979e_pri0/T0.jpeg"
      }
    ]
  },
  "start_timestamp_ms": 1710267524735,
  "duration_us": 5970993
}

```

9. Скачать выходной файл из хранилища S3;

10. Проверить характеристики видео файла на соответствие параметрам для выходного файла выполнив команду `ffprobe -i high.m4s`. Характеристики файла должны соответствовать параметрам выходного файла в следующем формате:

```
ffmpeg
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'high.m4s':
Metadata:
  major_brand   : iso5
  minor_version : 512
  compatible_brands: iso5iso6mp41
  encoder      : Lavf59.37.100
Duration: 00:00:31.66, start: 0.000000, bitrate: 8709 kb/s
Stream #0:0[0x1](und): Video: h264 (Main) (avc1 / 0x31637661), yuv420p(tv, bt709, progressive), 1920x1080 [SAR 1:1 DAR 16:9], 7951 kb/s, 25 fps, 25 tbr, 12800 tbn (default)
Metadata:
  handler_name   : Video Media Handler
  vendor_id     : [0][0][0][0]
Stream #0:1[0x2](und): Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, stereo, fltp, 256 kb/s (default)
Metadata:
  handler_name   : Sound Media Handler
  vendor_id     : [0][0][0][0]
```

11. Открыть файл на просмотр в плеере (например, VLC). Файл должен корректно воспроизводиться.

3.1.5 Проверка токена

Для проверки токена необходимо выполнить следующие действия:

1. Пробросить порт с сервиса авторизации Медиаплатформы для доступа к API сервиса;
2. Пробросить порт с сервиса управления задачами для доступа к API сервиса;
3. Зарегистрировать клиент;
4. Получить ключ API;
5. Получить JWT-токен. Убедиться, что токен валидный, в том числе по сроку действия;
6. Отправить запрос на проверку JWT-токена в следующем формате:

```
curl --location 'http://localhost:58498/api/v1/auth/token' \
--header 'accept: text/plain' \
--header 'Authorization: Bearer <insert_jwt_token>'
```

7. Указать токен, выданный для клиента EVP. В результате должен быть получен ответ 200 OK. В случае невалидного токена будет получен ответ 401 Unauthorized или Invalid or expired JWT.

3.2 Система загрузки аудиовизуального контента в сети Интернет

Система загрузки аудиовизуального контента Медиаплатформы в сети Интернет поддерживает различные варианты загрузки пользовательских файлов, в том числе следующими способами:

- Одним или несколькими файлами;
- Чанками файла;
- Путем скачивания файла с FTP-сервера;
- Путем скачивания файла с HTTP-сервера;
- Многопоточная загрузка.

Указанные способы загрузки будут рассмотрены подробнее ниже.

3.2.1 Загрузка одним или несколькими файлами

В целях загрузки файла в Медиаплатформу данным способом необходимо выполнить следующие действия:

1. Подобрать файл визуального контента допустимого формата (например, в формате .mp4) размером не более 24 Гб и общей продолжительностью не более 12 часов;
2. Загрузить файл в хранилище S3, используя конфигурационный файл с одним профилем (например, 1_profile.json);
3. Убедиться, что файл прошел обработку на разрешенный формат. В результате в топике Kafka с наименованием taskmanager будет зафиксирован статус об успешной загрузке. При этом будет получено сообщение следующего формата:

```
{
  "message_type": "TRANSFER_TASK_FINISHED",
  "error": "",
  "status": "OK",
  "task_id": "01HAY42KPJH98CR0ZKWPAJ2X04",
  "source": {
    "provider": "s3",
    "bucket": "mp-support-storage",
    "s3_region": "ru-central1",
    "host_uri": "https://storage.yandexcloud.net",
    "file_uri": "01HAY42KPJH98CR0ZKWPAJ2X04/12hours+.mp4"
  },
  "destination": {
    "provider": "s3",
    "bucket": "mp-support-storage",
    "s3_region": "ru-central1",
```

```

"host_uri": "https://storage.yandexcloud.net",
"file_uri": "01HAY42KPJH98CR0ZKWPAJ2X04/12hours+.mp4"
},
"telemetry_trace_context": {
"trace_id": "2f3a1614f5c8648d",
"parent_span_id": "2f3a1614f5c8648d"
},
"errors": []
}

```

Допускается одновременная загрузка в нескольких консолях типа sendtool. Для этого необходимо открыть по отдельному окну консоли для каждого загружаемого файла.

3.2.2 Загрузка чанками файлов

В целях загрузки файла в Медиаплатформу данным способом необходимо выполнить следующие действия:

1. Открыть Swagger UI сервиса Uploader по ссылке https://<domain_address>/namespace/_/docs/swagger/index.html. В ответ на запрос будет получен код 201;
2. Создать задачу на загрузку с помощью поля POST Create new upload task;
3. Проверить наличие сообщения о создании задачи загрузки в топике taskmanager. В топике Kafka taskmanager будет зафиксирован статус о создании задачи загрузки;
4. Отправить чанк файла с помощью поля PUT Upload bytes;
5. В поле Upload ExternalTask ID вставить task_id, полученный при создании задачи на загрузку;
6. В поле part ввести значение «1»;
7. В поле body вставить содержимое файла base64.txt. Будет получен ответ о завершеном статусе задачи;
8. Проверить статус загрузки файла в топике Kafka taskmanager (будет зафиксирован статус об успешной загрузке). Также файл будет загружен в хранилище S3.

В результате успешной загрузки будет получено сообщение соответствующего формата (см. п. 3.2.1).

3.2.3 Загрузка путем скачивания файла с FTP-сервера

В целях загрузки файла в Медиаплатформу данным способом необходимо выполнить следующие действия:

1. Загрузить на FTP-сервер исходный файл.
2. Открыть Swagger UI сервиса Медиаплатформы по ссылке https://<domain_address>/namespace/_/docs/swagger/index.html;
3. Создать задачу на загрузку с помощью поля POST Create download task. Будет получен ответ на запрос в виде кода 200 ОК;
4. Проверить наличие сообщения о создании задачи на скачивание в топике taskmanager. В топике будет зафиксирован статус о создании задачи скачивания;
5. Проверить статус загрузки файла в Kafka топике taskmanager. Исходный файл будет загружен в хранилище S3;

6. В топике Kafka taskmanager будет зафиксирован статус об успешной загрузке;
 7. Убедиться, что файл прошел обработку на разрешенный формат. В топике Kafka taskmanager будет зафиксирован статус об успешной обработке сервисом Медиаплатформы.
- В результате успешной загрузки будет получено сообщение соответствующего формата (см. п. 3.2.1).

3.2.4 Загрузка путем скачивания файла с HTTP-сервера

В целях загрузки файла в Медиаплатформу данным способом необходимо выполнить следующие действия:

1. Загрузить на FTP-сервер исходный файл;
 2. Открыть Swagger UI сервиса Медиаплатформы по ссылке https://<domain_address>/namespace/_/docs/swagger/index.html;
 3. Создать задачу на загрузку с помощью поля POST Create download task. Будет получен ответ на запрос в виде кода 200 ОК;
 4. Проверить наличие сообщения о создании задачи на скачивание в топике Kafka taskmanager. В нем будет зафиксирован статус о создании задачи на скачивание;
 5. Проверить статус загрузки файла в топике Kafka taskmanager. Исходный файл будет загружен в хранилище S3;
 6. В топике Kafka taskmanager будет зафиксирован статус об успешной загрузке;
 7. Убедиться, что файл прошел обработку на разрешенный формат. В топике Kafka taskmanager будет зафиксирован статус об успешной обработке сервисом Медиаплатформы.
- В результате успешной загрузки будет получено сообщение соответствующего формата (см. п. 3.2.1).

3.2.5 Многопоточная загрузка

В целях загрузки файла в Медиаплатформу данным способом необходимо выполнить следующие действия:

1. Загрузить исходный файл, указав в команде для инструмента sendtool ключ --threads со значением 4. В результате в консоли будут отображены сообщения о том, что файл был отправлен в 4 потока - "Start thread n/ End thread n";
2. Убедиться, что файл прошел обработку на разрешенный формат. В топике Kafka taskmanager будет зафиксирован статус об успешной обработке файла сервисом Медиаплатформы;
3. Проверить статус загрузки файла в топике Kafka taskmanager (будет зафиксирован статус об успешной загрузке). Также файл будет загружен в хранилище S3.

В результате успешной загрузки будет получено сообщение соответствующего формата (см. п. 3.2.1).

3.2.6 Удаление исходных файлов после загрузки

Также доступна функция удаления файлов в исходном формате после их загрузки в Медиаплатформу. Для этого необходимо включить опцию удаления файла оригинала после успешного транскодирования с наименованием DELETE_SOURCES в конфигурационном файле 1_profile.json. При этом при попытке скачать файл оригинала из хранилища S3 по ссылке формата https://<domain_address>/<namespace>/<task_id>/original.mp4 будет получена ошибка 404.

3.3 Микросервис диагностики файлов аудиовизуального контента

Микросервис диагностики файлов аудиовизуального контента обеспечивает устойчивость средств Медиаплатформы к поврежденному загружаемому контенту путем выполнения диагностики ошибок исходных файлов.

3.3.1 Диагностика загружаемых файлов

При диагностике файлов должен использоваться конфигурационный файл с одним профилем (например, 1_profile.json). В целях проверки валидности загружаемого файла необходимо выполнить следующие действия:

1. Загрузить исходный файл;
2. Убедиться, что файл прошел обработку.

В случае успешной диагностики в топике Kafka taskmanager будет зафиксирован соответствующий статус обработки микросервисом. При этом будет получено сообщение следующего формата:

```
{
  "message_type": "PROBE_TASK_FINISHED",
  "task_id": "01HAWE5MVC13X846QD5BNQ9D2Q",
  "status": "OK",
  "errors": [],
  "location": {
    "provider": "s3",
    "bucket": "mp-support-storage",
    "s3_region": "ru-central1",
    "host_uri": "https://storage.yandexcloud.net",
    "file_uri": "01HAWE5MVC13X846QD5BNQ9D2Q/avc_test.mp4"
  },
  "file_size": 40252099,
  "container_short_names": "mov,mp4,m4a,3gp,3g2,mj2",
  "streams": [
    {
      "stream_type": "VIDEO",
      "stream_idx": 0,
      "duration": 31.799999237060547,
      "codec": "h264",
      "bitrate": 10030136,
      "width": 1920,
      "height": 1080,
      "pix_fmt": "yuv420p",
```

```
"sample_aspect_ratio": {
  "num": 1,
  "den": 1
},
"display_aspect_ratio": {
  "num": 16,
  "den": 9
},
"frame_rate": {
  "num": 25,
  "den": 1
},
"color_range": "tv",
"color_primaries": "bt709",
"color_trc": "bt709",
"color_space": "bt709",
"chroma_location": "left"
},
{
  "stream_type": "AUDIO",
  "stream_idx": 1,
  "duration": 31.70133399963379,
  "codec": "aac",
  "bitrate": 128003,
  "samplerate": 48000,
  "channels": 2
}
],
"best_video_stream": 0,
"best_audio_stream": 1,
"demux_ok": true,
"start_timestamp_ms": 1695318989955,
"duration_us": 1222182
}
```

Успешно диагностированные файлы аудиовизуального контента будут доступны для дальнейшей обработки программными средствами Медиаплатформы (например, для транскодирования).

В случае возникновения ошибки диагностики файла (например, на этапе проверки на разрешенный формат) в топике Kafka taskmanager будет зафиксирован соответствующий статус обработки файла микросервисом. При этом будет получено сообщение следующего формата:

```
{
  "message_type": "PROBE_TASK_FINISHED",
  "task_id": "01GVK7RB246FZTK9CVV0ZQ5BWR",
  "status": "FAILED",
  "errors": [
    {
      "error_code": 1030204,
      "error_message": "Error decoding. Cannot detect stream 0 duration"
    }
  ],
  "location": {
    "provider": "s3",
    "bucket": "mp-rupor-beta",
    "s3_region": "ru-central1",
    "host_uri": "https://storage.yandexcloud.net",
    "file_uri": "01GVK7RB246FZTK9CVV0ZQ5BWR/deadlock.mp4"
  },
  "file_size": 357461,
  "container_short_names": "mov,mp4,m4a,3gp,3g2,mj2",
  "streams": [
    {
      "stream_type": "VIDEO",
      "stream_idx": 0,
      "duration": 0.0,
      "codec": "hevc",
      "bitrate": 18115721,
      "width": 720,
      "height": 1280,
      "display_aspect_ratio": {
```

```
"num": 9,  
"den": 16  
}  
},  
{  
"stream_type": "AUDIO",  
"stream_idx": 1,  
"duration": 8.057459831237793,  
"codec": "aac",  
"bitrate": 262020,  
"samplerate": 44100,  
"channels": 2  
}  
],  
"best_video_stream": 0,  
"best_audio_stream": 1,  
"demux_ok": true,  
"start_timestamp_ms": 1678904143365,  
"duration_us": 202747  
}
```

Дальнейшая обработка файла с диагностированной ошибкой будет прервана.

3.4 Система транскодирования аудиовизуального контента

Система транскодирования аудиовизуального контента обеспечивает транскодирование файлов после их загрузки и проверки программными средствами Медиаплатформы в предустановленный формат со следующими характеристиками:

- Контейнер - m4s
- Видео кодек - h264
- Разрешение - 1920x1080
- Соотношение сторон - 16:9
- FPS - 30
- Аудио кодек - aac
- Аудио каналы – 2
- Частота дискретизации аудио – 48 кГц

Проверить характеристики файла на соответствие выходным параметрам возможно с помощью выполнения следующей команды консоли:

```
ffprobe -i high.m4s
```

При этом будет получено сообщение следующего формата:

ffmpeg

Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'high.m4s':

Metadata:

major_brand : iso5

minor_version : 512

compatible_brands: iso5iso6mp41

encoder : Lavf59.37.100

Duration: 00:00:31.67, start: 1.458000, bitrate: 8384 kb/s

Stream #0:0[0x1](und): Video: h264 (Main) (avc1 / 0x31637661), yuv420p(tv, bt709, progressive), 1920x1080 [SAR 1:1 DAR 16:9], 7646 kb/s, 30 fps, 30 tbr, 15360 tbn (default)

Metadata:

handler_name : VideoHandler

vendor_id : [0][0][0][0]

Stream #0:1[0x2](und): Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, stereo, fltp, 256 kb/s (default)

Metadata:

handler_name : SoundHandler

vendor_id : [0][0][0][0]

Транскодированные с помощью Медиаплатформы файлы аудиовизуального контента будут доступны для воспроизведения в плеере (например, VLC) в корректном виде с поддержкой аудиоряда.

Также изменения параметров выходного файла по сравнению с исходным форматом будут отображаться в свойствах проигрываемого в плеере файла в части следующих параметров:

- Кодека;
- Разрешения;
- Битрейта;
- Пресета кодека;
- Профиля кодека.

Отображение параметров в сравнении между исходным и выходным форматом файлов приведено на Рисунках 1 и 2.

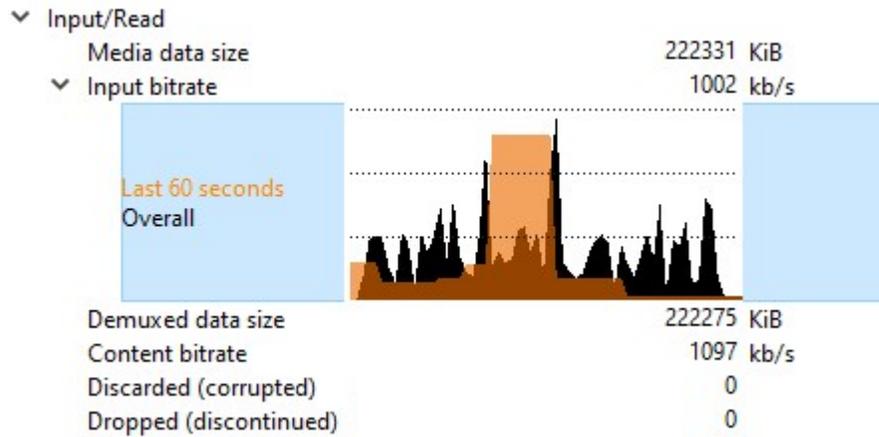


Рисунок 1 – Пример битрейта в исходном файле средствами отображения метаданных плеера VLC

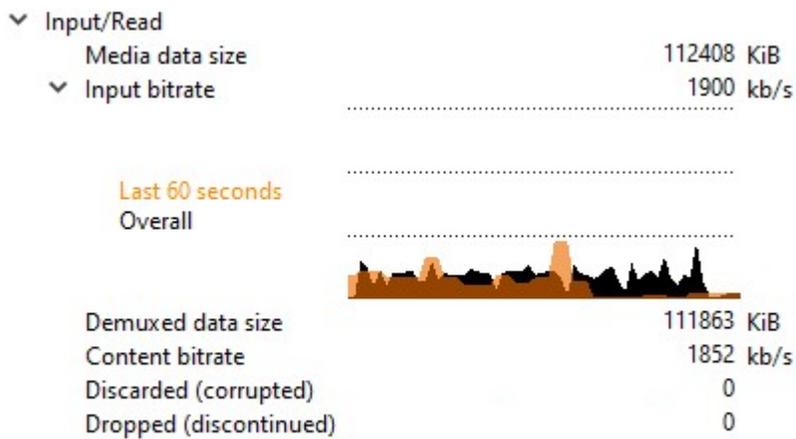


Рисунок 2 – Пример битрейта в выходном файле средствами отображения метаданных плеера VLC

При этом в процессе транскодирования производится унификация файлов и приведение их параметров к одному виду, в том числе возможны следующие виды изменений выходного файла по сравнению с оригиналом:

- Увеличение/уменьшение разрешения видеоряда;
- Изменение FPS видеоряда;
- Изменение частоты дискретизации аудиоряда;
- Изменение числа каналов аудиоряда;
- Изменение деинтерлейсинга.

Однако, будут сохранены следующие параметры исходного файла:

- Цветовое пространство;
- Наложение статических изображений (файл водяного знака должен быть замещен в доступном Медиаплатформе хранилище с изменением параметров доступа к нему в конфигурационном файле*);

- Наложение динамических изображений (файл водяного знака должен быть замещен в доступном Медиаплатформе хранилище с изменением параметров доступа к нему в конфигурационном файле*);

- Степень прозрачности водяного знака (файл водяного знака должен быть замещен в доступном Медиаплатформе хранилище с изменением параметров доступа к нему в конфигурационном файле*);

- Применение водяного знака для каждого выходного файла (файл водяного знака должен быть замещен в доступном Медиаплатформе хранилище с изменением параметров доступа к нему в конфигурационном файле*);

- Поддержка нескольких аудиодорожек исходного файла;

- Вставка черных полей видеофайла;

- Ориентация видеоряда (вертикальная или горизонтальная);

***Примечание:** изменение параметров доступа файла водного знака в конфигурационном файле производится в блоке `Overlay/Source` в части следующих параметров:

- `source.provider` - тип хранилища;

- `source.bucket` - имя контейнера (bucket) в хранилище S3;

- `source.s3_region` - название региона хранилища S3;

- `source.host_uri` - сетевой путь к службе S3 в формате `{схема}://{хост|IP}[:{порт}]`;

- `source.file_uri` - путь к файлу в хранилище;

- `source.http_url` - необязательный путь к медиафайлу для типа провайдера HTTP; полный URL-адрес, включая все опции GET-запроса.

Также при наличии ошибок синхронизации аудио- и видеоряда исходного файла в процессе транскодирования будет произведено их исправление. Файл выходного формата будет сохранен в синхронизированном виде.

Дополнительно в процессе транскодирования средствами Медиаплатформы возможно разделение аудио- и видеоряда на отдельные сущности. Для получения исключительно видеоряда без аудио в консоли необходимо выполнить следующую команду:

```
ffprobe -i high.m4s
```

Для получения исключительно аудиоряда без видео в консоли необходимо выполнить следующую команду:

```
ffprobe -i audio-only.m4s
```

Процесс использует несколько транскодеров в параллельном режиме (CPU и GPU) и поддерживает обработку очереди заданий на транскодирование по принципу использования первого освободившегося транскодера.

Проверить обработку файла тем или иным транскодером возможно путем просмотра топиков `Kafka transcoder.tasks.cpu` и `transcoder.tasks.gpu` соответственно на предмет наличия сообщений вида `"TRANSCODER_TASK"` с `task_id` исходного файла.

3.4.1 Настройка профилей транскодирования

Для настройки профилей транскодирования необходимо поместить сообщение в топик `Kafka taskmanager.settings`. При запуске сервис Медиаплатформы читает последнее сообщение из данного топика и применяет заданные настройки. В процессе работы настройки применяются по мере их добавления в топик `taskmanager.settings`.

Для корректной работы необходимо, чтобы сообщение имело заголовок `type=TASKMANAGER_SETTINGS`.

3.4.2 Обрезка черных полос (каше)

Если в исходном файле присутствует каше в виде черных полос, то средствами Медиаплатформы возможно выполнение их обрезки в процессе транскодирования. Для этого необходимо выполнить следующие действия:

3. Использовать конфигурационный файл с одним профилем (например, `1_profile_crop_letterbox_true.json`), где включен параметр `crop`;
4. Загрузить исходный файл;
5. Убедиться, что файл прошел обработку на разрешенный формат, проверив наличие секции `"crop_params"`;
6. Убедиться, что файл успешно прошел транскодирование;
7. Скачать выходной файл из хранилища S3;
8. Проверить характеристики видео файла на соответствие параметрам для выходного файла, выполнив команду консоли `ffprobe -i high.m4s`.

В результате характеристики файла будут соответствовать параметрам выходного файла. В метаданных файла будет отображено, что высота видео изменилась после обрезки черных полос. В плеере видеофайл будет воспроизводиться без каше в верхней и нижней части изображения.

3.4.3 Выравнивание громкости

Выравнивание громкости необходимо для комфортного просмотра видеоконтента. С этой целью сервисом Медиаплатформы в загружаемых файлах определяются показатели громкости аудиоряда. Если показатели слишком тихие или слишком громкие, то возможно транскодирование аудиоряда. Для этого необходимо выполнить следующие действия:

1. Использовать конфигурационный файл с одним профилем (например, `1_profile_normalize_audio_true.json`), где включен параметр `"normalize_volume"` и в профиль добавлен параметр `"target_db"` со значением целевой громкости аудиоряда.
2. Загрузить исходный файл;
3. Убедиться, что файл прошел обработку на разрешенный формат, проверив наличие секции `"volume_params"`;
4. Убедиться, что файл успешно прошел транскодирование;
5. Скачать выходной файл из хранилища S3;
6. Проверить характеристики видео файла на соответствие параметрам для выходного файла, выполнив команду консоли `ffmpeg -i high.m4s -filter:a volumedetect -f null /dev/null`

В результате характеристики файла будут соответствовать параметрам выходного файла. В метаданных файла будет отображено, что `mean_db` равно указанной в профиле `target_db`. В плеере выходной файл будет воспроизводиться с соответствующе измененной громкостью по сравнению с исходным.

3.4.4 Нарезка и склейка контента

Функционал нарезки и склейки видеоконтента необходим для автоматического удаления из исходного видео сегментов, которые не предназначены для публикации (например, кадры с технической информацией). В задаче на транскодирование передаётся список сегментов с временем их начала и конца.

На этапе транскодирования из исходного видео выделяются сегменты и склеиваются в результирующий видеоряд.

На этапе определения параметров создания эскиза учитывается длительность склеенного видео.

После склейки временные метки кадров корректируются. Последовательность обработки полностью аналогична основному потоку. Если в задаче на транскодирование присутствует секция с сегментами для нарезки, Медиаплатформа на основе длительности всего видео и сегментов для нарезки формирует метаданные для миниатюр.

При этом производятся следующие действия:

- Получение задания;
- Нарезка сегментов из списка;
- Транскодирование сегменты исходного видеофайла таким образом, чтобы фреймы были внутри выбранных сегментов;
- Корректировка временных штампов выходных кадров таким образом, чтобы они соответствовали полученному таймлайну;
- Формирование миниатюр и аудиодорожек.

Для этого необходимо выполнить следующие действия:

1. Отправить запрос на скачивание файла, добавив блок `editor_list`: В случае успеха в ответе на запрос будет получен код 200 ОК;
2. Убедиться, что файл загружен и прошел обработку на разрешенный формат;
3. Убедиться, что в задачу транскодирования были добавлены параметры редактирования файла. При этом в соответствующей задаче топике `Kafka transcoder.tasks` будет зафиксировано сообщение, в котором присутствует поле `editor` из задачи на скачивание;
4. Убедиться, что файл успешно прошел транскодирование;
5. Скачать выходной файл из хранилища S3;
6. Проверить характеристики видео файла на соответствие параметрам для выходного файла, выполнив команду `ffprobe -i high.m4s`.

В результате характеристики файла будут соответствовать параметрам выходного файла. Длина файла будет соответствовать ожидаемым после редактирования (например, ~ 10 секунд). В плеере будут воспроизводиться только сохраненные части видео - с 5 по 10 и с 15 по 20 секунды.

3.4.5 Прикрепление заставки к исходному контенту

Для прикрепления заставки к основному видео, должны быть выполнены следующие настройки:

- Ссылка на заставку должна быть указана в пресете для транскодирования;
- Внутренняя ссылка должна быть передана в задаче на транскодирование;
- Видеофайл с заставкой загружен в локальное хранилище.

При установке параметр кодирования в пресете, видео с заставкой перекачивается в локальное хранилище и запускается процесс определения параметров видео (размер кадра, формат пикселя). Метаданные сохраняются для дальнейшего использования. Для загрузки заставки используется тот же самый процесс, что и для загрузки обычного видео. При обработке исходного видео, метаданные заставки сравниваются с метаданными исходного видео и приводятся к нему в случае различий (размер кадра уменьшается или увеличивается, формат пикселя, битность и FPS приводятся к целевым значениям).

Сравнивается наличие видео- и аудиодорожек. Если количество видео- и аудиодорожек не совпадает, то заставка не прикрепляется.

Логотип (оверлей) на заставке в текущей реализации не прожигается.

Для выполнения прикрепления заставки необходимо выполнить следующие действия:

1. Загрузить файл заставки (например, preroll_countdown.webm) на сервер (например, HTTP);
2. Создать пресет с помощью запроса POST Set transcoding preset.
3. В поле request скопировать содержимое файла (например, preset_preroll.json).

Будет получен ответ на запрос в виде кода 200 ОК;

4. В БД в таблице preset_links найти task_id с префиксом PREROLL;
5. Выполнить поиск в топике taskmanager по task_id;
6. Убедиться, что файл заставки прошел проверку. Файл заставки должен быть загружен в хранилище S3.
7. В топике Kafka taskmanager должен быть зафиксирован статус об успешном пробировании файла.
8. Создать задачу на загрузку с помощью поля POST Create download task. Будет получен ответ на запрос в виде кода 200 ОК. Тестовый файл будет загружен в хранилище S3.
9. Убедиться, что файл загружен и прошел обработку на разрешенный формат;
10. Убедиться, что файл успешно прошел транскодирование;
11. Скачать выходной файл из хранилища S3;
12. Проверить характеристики видео файла на соответствие параметрам для выходного файла выполнив команду ffprobe -i high.m4s.

В результате характеристики файла будут соответствовать параметрам выходного файла. Длина файла будет соответствовать ожидаемой после добавления заставки. В плеере видео будет воспроизводиться с заставкой.

3.4.6 Поддержка субтитров в формате Teletext и DVB subtitles

Средствами Медиаплатформы реализована поддержка субтитров в формате Teletext и DVB. Для отображения субтитров в видеофайлах необходимо выполнить следующие действия:

1. Загрузить на сервер (например, HTTP) видеофайл, содержащий 2 потока субтитров в формате DVB- и Teletext;
2. Создать клиент и ключ API;
3. Получить JWT-токен;
4. Пробросить порт для сервиса Медиаплатформы;
5. Кодек mp2 должен быть добавлен в белый список;
6. Отправить запрос на создание пресета;
7. Отправить запрос на создание задачи на обработку, указав файл для скачивания имя файла (например, BBC.ts) и имя пресета (например, preset_1profile_ts);
8. Убедиться, что файл был загружен и прошел обработку на разрешенный формат
9. Убедиться, что файл успешно прошел транскодирование;
10. Скачать выходной файл из хранилища S3;
11. Проверить характеристики видео файла на соответствие параметрам для выходного файла, выполнив команду fprobe -i high.m4s. Характеристики файла должны соответствовать параметрам выходного файла;
12. Открыть результирующий файл в плеере (например, VLC);

13. Проверить отображение DVB- и Teletext-субтитров. Субтитры будут сохранены, оба варианта субтитров будут воспроизводиться при их выборе в плеере.

3.4.7 Сохранение результатов транскодирования

На стенде Медиаплатформы может быть настроен перенос результирующих файлов на ресурс FileHear Rutube. Для этого в Медиаплатформе должна быть настроена прямая загрузка файла в FileHear Rutube в части следующих параметров:

- STORAGE_PROVIDER - тип используемого хранилища для загрузки файлов. Возможные значения: {S3|FILEHEAP}. Значение по умолчанию: S3;
- FILEHEAP_USERNAME - пользователь для подключения к хранилищу FILEHEAP;
- FILEHEAP_PASSWORD - пароль для подключения к хранилищу FILEHEAP;
- FILEHEAP_ENDPOINT - ссылка на сервер для загрузки файлов в хранилище подключения к хранилищу FILEHEAP.

При наличии настроек для успешно загруженных и транскодированных файлов будет создана задача на копирование результирующих файлов в FileHear. При этом в топике Kafka fhdelivery будет размещено сообщение с задачей на копирование файлов следующего формата:

```
{
  "message_type": "DELIVERY_TASK",
  "task_id": "01HDP6JNR8RHZBAYQ0CBEBR5VW",
  "error": "",
  "status": ""
}
```

В топике taskmanager будет зафиксирован статус об успешном копировании файлов следующего формата:

```
{
  "message_type": "DELIVERY_TASK_FINISHED",
  "task_id": "01HDP6JNR8RHZBAYQ0CBEBR5VW",
  "location": [
    {
      "source": {
        "provider": "s3",
        "s3_region": "",
        "host_uri": "",
        "file_uri": "01HDP6JNR8RHZBAYQ0CBEBR5VW/T0.jpeg"
      },
    },
    "destination": {
      "provider": "fileheap",
      "bucket": "b7c6207b-7478-4a9e-97f1-0fb31c3b834c",
    }
  ]
}
```

```
"s3_region": "",
"host_uri": "",
"file_uri": "",
"http_url": "http://10.65.8.78:8888/RO/54a77730-da02-11ec-82ae-
005056afce89/83/4c/b7c6207b74784a9e97f10fb31c3b834c"
},
{
"source": {
"provider": "s3",
"s3_region": "",
"host_uri": "",
"file_uri": "01HDP6JNR8RHZBAYQ0CBEBR5VW/high.m3u8"
},
"destination": {
"provider": "fileheap",
"bucket": "3dbf075f-a508-489f-b9dd-4246946732bf",
"s3_region": "",
"host_uri": "",
"file_uri": "",
"http_url": "http://10.65.8.77:8888/RO/1f0b66c2-da02-11ec-8845-
005056af1a6b/32/bf/3dbf075fa508489fb9dd4246946732bf"
},
{
"source": {
"provider": "s3",
"s3_region": "",
"host_uri": "",
"file_uri": "01HDP6JNR8RHZBAYQ0CBEBR5VW/high.m4s"
},
"destination": {
"provider": "fileheap",
"bucket": "7bdc5884-46fb-456e-a752-0ebd086c879d",
"s3_region": "",
```

```
"host_uri": "",
"file_uri": "",
"http_url": "http://10.65.8.77:8888/RO/1f0b66c2-da02-11ec-8845-
005056af1a6b/87/9d/7bdc588446fb456ea7520ebd086c879d"
}
},
{
"source": {
"provider": "s3",
"s3_region": "",
"host_uri": "",
"file_uri": "01HDP6JNR8RHZBAYQ0CBEBR5VW/high_master.m3u8"
},
"destination": {
"provider": "fileheap",
"bucket": "ee25f881-6548-4418-8c91-ea4fae67cc51",
"s3_region": "",
"host_uri": "",
"file_uri": "",
"http_url": "http://10.65.8.78:8888/RO/54a77730-da02-11ec-82ae-
005056afce89/cc/51/ee25f881654844188c91ea4fae67cc51"
}
},
{
"source": {
"provider": "s3",
"s3_region": "",
"host_uri": "",
"file_uri": "01HDP6JNR8RHZBAYQ0CBEBR5VW/master.m3u8"
},
"destination": {
"provider": "fileheap",
"bucket": "ecbd4baa-5dcd-4d24-80d9-e949c6fe4a0c",
"s3_region": "",
"host_uri": "",
```

```
"file_uri": "",
"http_url": "http://10.65.8.77:8888/RO/326dc8fe-da02-11ec-b269-
005056af1a6b/4a/0c/ecbd4baa5dcd4d2480d9e949c6fe4a0c"
}
},
{
"source": {
"provider": "s3",
"s3_region": "",
"host_uri": "",
"file_uri": "01HDP6JNR8RHZBAYQ0CBEBR5VW/thumbnail.jpg"
},
"destination": {
"provider": "fileheap",
"bucket": "41aa376e-9268-4319-a0be-398714ef0d7f",
"s3_region": "",
"host_uri": "",
"file_uri": "",
"http_url": "http://10.65.8.78:8888/RO/503a7eb8-da02-11ec-8e03-
005056afce89/0d/7f/41aa376e92684319a0be398714ef0d7f"
}
},
{
"source": {
"provider": "s3",
"s3_region": "",
"host_uri": "",
"file_uri": "01HDP6JNR8RHZBAYQ0CBEBR5VW/trickmode.json"
},
"destination": {
"provider": "fileheap",
"bucket": "e69027dc-2ce7-4093-98b1-960d24d72839",
"s3_region": "",
"host_uri": "",
"file_uri": "",
```


видеовещания осуществляется для следующих исходных форматов аудиовизуального контента:

- RTMP
- SRT

Транскодирование видеовещания в указанных форматах средствами Медиаплатформы включает в себя следующие опции:

- Деинтерлейсинг
- Наложение логотипа

3.5.1 Деинтерлейсинг видеовещания

Для использования опции деинтерлейсинга при видеовещании необходимо выполнить следующие действия:

1. Пробросить порт для сервиса управления задачами Медиаплатформы;
2. Отправить запрос на создание вещания, пример:

```
curl --location 'http://localhost:59833/v1/streaming' \  
--header 'accept: application/json' \  
--header 'Content-Type: application/json' \  
--header 'Authorization: Bearer <insert_jwt_token>' \  
--data '{  
"input": {  
"address": "srt://10.133.75.119:1935"  
},  
"ffmpeg_video_decoding_options": {  
"threads": "4"  
},  
"ffmpeg_audio_decoding_options": {  
"threads": "1"  
},  
"transcoders": {  
"video": [  
{  
"name": "stream",  
"codec": "libx264",  
"filter_options": {  
"width": 1280,  
"height": 720,  
"overlay": false,  
"keep_interlace": false
```

```
},  
"ffmpeg_encoding_options": {  
  "threads": "3",  
  "g": "60",  
  "maxrate": "3M"  
}  
},  
],  
"audio": [  
  {  
    "name": "48kHz",  
    "codec": "aac",  
    "filter_options": {  
      "channels": 2,  
      "samplerate": 48000  
    },  
    "ffmpeg_encoding_options": {  
      "b": "256k"  
    }  
  }  
],  
},  
"channels": {  
  "rtmp": [  
    {  
      "name": "stream",  
      "video": "stream",  
      "audio": "48kHz",  
      "address": "rtmp://10.133.73.39:1935/live/stream"  
    }  
  ]  
}  
}' Получен ответ 200 ОК.
```

Пример body:

01HRZJRN0GYS785BX668Y2P4HS

3. Убедиться, что вещание успешно запущено. В топике Kafka taskmanager должен быть зафиксирован статус старта вещания:

```
{
"message_type": "STREAMER_START_RESPONSE",
"task_id": "01HRZJJ3BM7BGDBRMNM45JG204",
"status": "OK"
}
```

4. Открыть вещание в ffplay и удостовериться, что деинтерлейсинг был выполнен. Развертка должна быть вида progressive:

```
Input #0, flv, from 'rtmp://localhost:1936/live/stream': 0B f=0/0
Metadata:
|RtmpSampleAccess: true
Server      : NGINX RTMP (github.com/arut/nginx-rtmp-module)
displayWidth  : 960
displayHeight : 720
fps          : 0
profile      :
level       :
Duration: 00:00:00.00, start: 32.917000, bitrate: N/A
Stream #0:0: Video: h264 (High), yuv420p(top first), 960x720 [SAR 1:1 DAR 4:3], 25 fps, 25 tbr,
1k tbn
Stream #0:1: Audio: aac (LC), 48000 Hz, stereo, fltp, 256 kb/s
47.77 A-V: -0.033 fd= 37 aq= 40KB vq= 188KB sq= 0B f=0/0
```

5. Аудио- и видеоряд при этом должны успешно проигрываться. Видеоряд не должен тормозить и/или содержать артефакты.

3.5.2 Наложение логотипа на видеовещание

Для использования опции наложения логотипа на видеовещание необходимо выполнить следующие действия:

1. Пробросить порт для сервиса управления задачами Медиаплатформы.
2. Загрузить файлы водяных знаков в хранилище (например, like.gif, premier_logo.png).
3. Отправить запрос на создание вещания (см. пример в п. 3.5.1);
4. Убедиться, что вещание успешно запущено. В топике Kafka taskmanager должен быть зафиксирован статус старта вещания (см. пример в п. 3.5.1);
5. Открыть вещание в плеере;
6. Проверить наличие водяных знаков в стриме. В нижнем левом углу должна отображаться анимация (например, like.gif);
7. В нижнем правом углу должна отображаться статичная картинка (например, premier_logo.png);

8. Аудио- и видеоряд при этом должны успешно проигрываться. Видеоряд не должен тормозить и/или содержать артефакты.

4 Службные функции Медиаплатформы

4.1 Логирование действий и событий

Логирование действий и событий в стандартных потоках вывода происходит независимо друг от друга для каждого сервиса Медиаплатформы.

В случае деплоя Kubernetes доступ к логам можно получить через стандартный механизм чтения логов пода.

Также поддерживается возможность вывода логов в формате json для дальнейшей интеграции с сервером логов.

Для доступа к логам действий и событий необходимо выполнить следующие действия:

1. Должен быть доступен поток MPEG TS HTTP на вход;
2. Должен быть открыт Swagger UI TaskManager;
3. Используя запрос POST Starts a new stream, отправить сообщение для старта стрима.
4. В поле request добавить содержимое файла stream_start.json.
5. Для смены источника изменить адрес в поле address. В ответ на запрос будет получен код 200 и task_id. В топике Kafka streamer.tasks.cpu будет зафиксирован статус об успешном старте стрима;
6. Открыть логи сервиса Медиаплатформы;
7. Удостовериться, что началась генерация live-стрима. В логах сервиса должны быть зафиксированы сообщения следующего типа:

```
2023-12-21 20:33:39.687 [T] [ReceivingWorkerThread.cpp @ operator():147] New message
received: offset=1, timestamp=2023-12-21 20:33:39.680,
key=01HJ71AWTQ9C02JTCQWBENKTV, type=STREAMER_START_REQUEST, body:
{"message_type":"STREAMER_START_REQUEST","task_id":"01HJ71AWTQ9C02JTCQWBEN
NKTV","input":{"address":"http://astra.zxz.su:8008/1705"},"ffmpeg_video_decoding_options":{"th
reads":"4"},"ffmpeg_audio_decoding_options":{"threads":"1"},"transcoders":{"video":[{"name":"h
igh","stream_selection":"","codec":"libx264","filter_options":{"width":1280,"height":720,"fps":{"n
um":30,"den":1},"overlay":false,"padding":false,"rotate":false,"allow_swap_dimensions":false},"ff
mpeg_encoding_options":{"bufsize":"10M","g":"60","maxrate":"5M","threads":"2"}},{"name":"lo
w","stream_selection":"","codec":"libx264","filter_options":{"width":854,"height":480,"fps":{"num
":30,"den":1},"overlay":false,"padding":false,"rotate":false,"allow_swap_dimensions":false},"ffmpe
g_encoding_options":{"bufsize":"5M","g":"60","maxrate":"2.5M","threads":"3"}],"audio":[{"nam
e":"48kHz","stream_selection":"","codec":"aac","filter_options":{"channels":2,"samplerate":48000}
,"ffmpeg_encoding_options":{"b":"256k"}]}],"channels":{"rtmp":[{"name":"high","video":"high",
"audio":"48kHz","address":"rtmp://10.42.10.196:1935/live/high"}, {"name":"low","video":"low", "au
dio":"48kHz","address":"rtmp://10.42.10.196:1935/live/low"}],"hls":[{"name":"HLS
channel","renditions":[{"name":"main","video":"high","audio":"48kHz"}, {"name":"secondary", "vid
eo":"low","audio":"48kHz"}],"destination":{"provider":"s3","bucket":"mp-cre-ft-01-
storage","s3_region":"ru-central1","host_uri":"https://minio.cyrm-
dev.cyrm.zxz.su:9000","file_uri":"streamer"}]}}
```

```
2023-12-21 20:33:39.687 [D] [Thread.cpp @ start:25] start
2023-12-21 20:33:39.687 [I] [StreamerApplication.cpp @ onStartResponse:131] Start response OK,
message=
```

```
2023-12-21 20:33:39.688 [T] [KafkaSender.cpp @ postMessage:147] Outgoing message:
topic=taskmanager, type=STREAMER_START_RESPONSE, body:
```

```
{
  "message_type": "STREAMER_START_RESPONSE",
  "task_id": "01HJ71AWTQ9C02JTCQWBENNKTV",
  "status": "OK"
}
```

```
2023-12-21 20:33:39.688 [I] [Thread.cpp @ operator():31] Thread id=0x7f11837fe000
```

```
2023-12-21 20:33:39.688 [I] [PipelineThread.cpp @ runSafe:51] Streaming
task='01HJ71AWTQ9C02JTCQWBENNKTV' was started
```

```
2023-12-21 20:33:39.700 [D] [AVUtils.cpp @ libavLogString:121] [V] [tcp @ 0x7f1174004100]
Starting connection attempt to 10.9.200.57 port 8008
```

```
2023-12-21 20:33:39.700 [D] [AVUtils.cpp @ libavLogString:121] [V] [tcp @ 0x7f1174004100]
Successfully connected to 10.9.200.57 port 8008
```

8. Пробросить порт с сервиса streamer-scr;
9. Открыть ссылку в плеере. Пример ссылки: `rtpm://localhost:1935/live/high`. В плеере должен воспроизводиться стрим;
10. Используя запрос DELETE Stops a stream, отправить сообщение для остановки стрима;
11. В поле `stream_id` добавить `task_id` стрима. В ответе на запрос будет получен код 200 OK;
12. В топике Kafka `streamer.tasks.scr` будет зафиксирован статус об успешном получении задачи на остановку стрима;
13. Открыть логи сервиса Медиаплатформы;
14. Удостовериться, что генерация стрима остановилась. В логах сервиса должны быть зафиксированы сообщения следующего типа:

```
2023-12-21 20:36:23.013 [T] [KafkaSender.cpp @ postMessage:147] Outgoing message:
topic=taskmanager, type=STREAMER_STOP_RESPONSE, body:
```

```
{
  "message_type": "STREAMER_STOP_RESPONSE",
  "task_id": "01HJ71AWTQ9C02JTCQWBENNKTV",
  "status": "OK",
  "stream_duration_sec": 163.323508059
}
```

```
2023-12-21 20:36:23.013 [D] [Thread.cpp @ requestStop:79] requestStop
```

4.2 Мониторинг

Для доступа к мониторингу функционирования сервисов Медиаплатформы необходимо выполнить следующие действия:

1. Должен быть доступен поток MPEG TS HTTP на вход;

2. Должен быть открыт Swagger UI TaskManager;
3. Используя запрос POST Starts a new stream, отправить сообщение для старта стрима.
4. В поле request добавить содержимое файла stream_start.json.
5. Для смены источника изменить адрес в поле address. В ответ на запрос будет получен код 200 и task_id. В топике Kafka streamer.tasks.cpu будет зафиксирован статус об успешном старте стрима;
6. Пробросить порт с сервиса streamer-scr;
7. Открыть ссылку в плеере. Пример ссылки: `rtmp://localhost:1935/live/high`. В плеере должен проигрываться стрим;
8. Открыть дашборды Streamer в Grafana. На графиках "Запись HLS потоков в S3/Кол-во демультимплексированных байт из входящих потоков (бт/с)/Кол-во мультимплексированных байт в исходящих потоках (байт/с)" будут отображены всплески активности на момент старта стрима;
9. Используя запрос DELETE Stops a stream, отправить сообщение для остановки стрима;
10. В поле stream_id добавить task_id стрима. В ответе на запрос будет получен код 200 OK. В топике Kafka streamer.tasks.cpu будет зафиксирован статус об успешном получении задачи на остановку стрима;
11. Открыть дашборды Streamer в Grafana. На графиках "Кол-во демультимплексированных байт из входящих потоков (бт/с)/Кол-во мультимплексированных байт в исходящих потоках (байт/с)" будет отображена убыль всплесков активности на момент остановки стрима;
12. График "Запись HLS" потоков в хранилище S3 не будет расти.

5 Действия при сбоях в функционировании Медиаплатформы

Медиаплатформа является сложным программным комплексом, в работе которого возможны сбои. В этом случае администратору Медиаплатформы необходимо принять меры к обнаружению и устранению причины сбоя. При невозможности обнаружения и/или устранения причины сбоя необходимо зарегистрировать инцидент. Для уменьшения количества сбоев рекомендуется выполнять действия, приведенные в разделе "Профилактические действия для предупреждения сбоев в работе Медиаплатформы" (см. п. 5.1).

5.1 Профилактические действия для предупреждения сбоев в работе Медиаплатформы

Профилактическое обслуживание Медиаплатформы заключается в следующих процессах:

- Установка обновлений безопасности для используемых сторонних программных систем и обновлений, на которых проведена валидация Медиаплатформы;
- Мониторинг окончания действия сертификатов на URL для API-запросов Медиаплатформы;
- Мониторинг заполненности хранилища и настройка по удалению ненужного контента;
- Регулярная проверка работоспособности Медиаплатформы, включая загрузку медиаконтента в систему, обработку медиаконтента и получение обработанного контента.

5.2 Основные положения для исследования проблем, связанных с работой Медиаплатформы

В общем случае, чтобы найти причину сбоя в работе Медиаплатформы, выполняются следующие шаги:

1. Проводится общий осмотр сервисов Медиаплатформы средствами Kubernetes, Rancher или Lens;
2. Проводится осмотр работоспособности стенда средствами мониторинга по результатам изучения соответствующих графиков в сервисе сбора метрик (например, Grafana);
3. Изучается лог сервиса управления задачами;
4. В случае обнаружения проблем связи с Kafka проверяется лог Kafka;
5. Изучаются сообщения в топике Kafka по адресу topics/taskmanager;
6. При наличии проблем с обработкой медиаконтента изучаются отчёты сервиса транскодирования;
7. Для выявления проблем с сетевой связностью, хранилищем и/или настройкой балансировщика изучаются отчёты сервиса загрузки контента.
8. При невозможности скачать имеющийся медиаконтент из хранилища через сервис доставки изучаются отчеты сервиса доставки.

При обнаружении проблем в сторонних системах необходимо следовать руководствам по решению проблем в этих системах.

При обнаружении проблем в подсистемах Медиаплатформы:

- Если подобный случай приведен в разделе «Известные проблемы в работе Медиаплатформы и методы их решения» (см. п. 5.3), необходимо следовать приведенным в нем указаниям;
- Если описание проблемы отсутствует в разделе «Известные проблемы в работе Медиаплатформы и методы их решения», необходимо зарегистрировать инцидент, по возможности, предоставив описание проблемы и логи, собранные во время выполнения шагов, приведенных в данном разделе.

5.3 Известные проблемы в работе Медиаплатформы и методы их решения

N	Проблема	Симптомы	Методы решения	Заметки
1.	Отказ драйверов Nvidia и CUDA на узлах	<ul style="list-style-type: none"> - Экземпляры GPU-Transcoder перестают обрабатывать видео. - #nvidia-smi на GPU-узлах возвращает ошибку 	<ol style="list-style-type: none"> 1. Перезагрузка GPU узлов. 2. Перезагрузка экземпляров GPU-Transcoder на дефектном GPU узле. 3. Проверка драйверов Nvidia и CUDA. 4. Проверка настройки nvidia-runtime 	Особенности настройки приведены в документации Nvidia docs.nvidia
2.	Отказ драйверов Nvidia и CUDA на подах	<ul style="list-style-type: none"> - Экземпляр GPU-Transcoder перестаёт обрабатывать видео. - #nvidia-smi на экземплярах GPU-Transcoder возвращает ошибку. - #nvidia-smi на GPU узлах работает корректно 	<ol style="list-style-type: none"> 1. Перезагрузка дефектных экземпляров GPU-Transcoder. 2. Проверка настройки nvidia-runtime 	Особенности настройки приведены в документации Nvidia docs.nvidia
3.	Закончилось действие сертификата на балансере	<ul style="list-style-type: none"> - Не загружается / не выгружается видео. - При пробросе порта для сервиса загрузки видео загружается. - При пробросе порта для сервиса доставки контент можно выгрузить 	<ol style="list-style-type: none"> 1. Проверить валидность сертификатов балансировщика. 2. Проверить логи и конфигурацию балансировщика 	
4.	Хранилище переполнено или недоступно	<ul style="list-style-type: none"> - Не загружается или не выгружается видео. - При пробросе порта для сервиса загрузки 	<ol style="list-style-type: none"> 1. Проверить доступность хранилища. 2. Проверить наличие нужных директорий или бакетов 	

		<p>видео не загружается.</p> <ul style="list-style-type: none"> - Хранилище недоступно. - Ошибки в логах сервисов загрузки и/или управления задачами при работе с хранилищем 	<p>(конфигурация в сервисе загрузки поможет понять необходимое).</p> <p>3. Проверить наличие ключей доступа и их соответствие ключам в конфигурации Медиаплатформы (конфигурация сервиса загрузки)</p>	
5.	Не работает Kafka	<ul style="list-style-type: none"> · Медиаплатформа не обрабатывает медиаконтент. · Сервисы Медиаплатформы в логах содержат сведения о потерянной связи с Kafka 	<ol style="list-style-type: none"> 1. Перезагрузка Kafka. 2. Перезагрузка сервисов Медиаплатформы 	Вероятны проблемы с узлом, на котором размещена Kafka
6.	Неподдерживаемые кодеки на устаревших версиях драйверов NVIDIA	<p>Экземпляры GPU-Transcoder постоянно перезагружаются и не могут обработать медиаконтент с ошибкой.</p> <p>Пример для кодека av1_nvenc - gpu-transcoder log:</p> <p>2023-07-26 08:23:38.646 [I] [VideoFilter.cpp @ open:61] Using GPU filter for source video stream #0</p> <p>2023-07-26 08:23:38.647 [D] [AVUtils.cpp @ libavLogString:111] [V] [av1_nvenc @ 0x7f61f6e26d40]</p>	<ol style="list-style-type: none"> 1. Проверка и обновление версии драйверов NVIDIA до 525 мажорной версии. 2. Выяснение TASK_ID с проблемными файлами, не подлежащими обработке. 3. Удаление директорий с проблемными файлами. 4. Перезагрузка экземпляров GPU-Transcoder. 5. Повторное разворачивание стенда 	Проводятся испытания и обновление версий драйверов для поддержки новых кодеков, зависящих от драйверов NVIDIA

		<p>Loaded Nvenc version 12.0 2023-07-26 08:23:38.647 [D] [AVUtils.cpp @ libavLogString:111] [V] [av1_nvenc @ 0x7f61f6e26d40] Nvenc initialized successfully 2023-07-26 08:23:38.655 [W] [AVUtils.cpp @ libavLogString:111] [W] [av1_nvenc @ 0x7f61f6e26d40] Codec not supported 2023-07-26 08:23:38.655 [C] [AVUtils.cpp @ libavLogString:111] [F] [av1_nvenc @ 0x7f61f6e26d40] Provided device doesn't support required NVENC features</p>		
7.	<p>Неподдерживаемые кодеки согласно спецификации Медиаплатформы</p>	<p>Медиаплатформа не обрабатывает медиаконтент – в отчётах сервиса управления задачами упоминается Unsupport codec</p>	<p>Проверить соответствие кодеков и формата загружаемого медиаконтента требованиям к загружаемым медиафайлам</p>	
8.	<p>Произведено масштабирование сервисов транскодирования или анализа данных средствами Kubernetes</p>	<p>Спорадические ошибки при обработке медиаконтента</p>	<p>Проверить соответствие partitions для сервиса анализа данных, CPU-Transcoder, GPU-Transcoder в Kafka. При необходимости привести в соответствие</p>	

			количество сервисов Медиаплатформы или заново развернуть стенд с корректной конфигурацией	
9.	Экземпляры GPU-Transcoder попали на узел без поддержки GPU	<ul style="list-style-type: none"> - Не обрабатывается медиаконтент на экземплярах GPU-Transcoder. - Выводятся ошибки на экземплярах GPU-Transcoder вида «could not find hwdevice» 	Переместить экземпляры GPU-Transcoder на узлы с необходимой оснасткой вручную или заново развернуть стенд	Вероятны проблемы с ребалансировкой подов и некорректной настройкой taints или другой системой распределения подов по специальным узлам

5.4 Действия при неустановленной проблеме

5.4.1 Остановка и старт Медиаплатформы

При неустановленной проблеме имеется возможность выполнить перезапуск Медиаплатформы.

5.4.2 Остановка Медиаплатформы

Для остановки сервисов Медиаплатформы необходимо выполнить следующие действия:

- Запомнить текущее количество реплик каждого сервиса (это необходимо для последующего возвращения нужного количества реплик);
- Средствами Kubernetes уменьшить количество реплик сервисов Медиаплатформы до нуля (при этом инфраструктурные сервисы останавливать нельзя).

5.4.3 Повторный запуск Медиаплатформы

Средствами Kubernetes восстановить нужное количество реплик для каждого из сервисов Медиаплатформы.

5.4.4 Откат установки

Действия при неуспешном перезапуске:

1. Удалить уже установленную версию Медиаплатформы, согласно документу "Инструкции по развертыванию Единой Медиаплатформы".
2. Вновь установить предыдущую версию Медиаплатформы, согласно документу "Инструкции по развертыванию Единой Медиаплатформы".

Для этого необходимо выбрать патч-версию Медиаплатформы меньше на 1 от устанавливаемой.

Если патч-версия равна 0, то необходимо выбрать версию с минорным значением, меньшим на 1 и максимальным номером патч-версии.

Так как Медиаплатформа состоит из набора stateless микросервисов, все обрабатываемые данные сохраняются во внешних сервисах.

Отдельных процедур по возврату данных в прежнее состояние не требуется.

