

**Инструкция по эксплуатации
программного комплекса
«Единая Медиаплатформа»**

Москва

2023

Содержание

Термины и определения.....	5
1. Введение.....	10
2. Архитектура Медиаплатформы.....	11
3. Описание компонентов Медиаплатформы.....	13
3.1. Uploader.....	13
3.1.1. Назначение сервиса Uploader.....	13
3.1.2. Переменные окружения Uploader.....	13
3.1.3. Сбор метрик Uploader.....	23
3.2. Prober.....	64
3.2.1. Назначение.....	64
3.2.2. Переменные окружения Prober.....	64
3.2.3. Сбор метрик.....	72
3.3. Transcoder.....	94
3.3.1. Назначение Transcoder.....	94
3.3.2. Обработка нескольких профилей.....	94
3.3.3. Переменные среды Transcoder.....	94
3.3.4. Сбор метрик.....	102
3.3.5. Алгоритм выравнивания аудио и видео дорожек.....	128
3.3.6. Алгоритм масштабирования видео.....	130
3.3.7. Допустимые аудио и видео временные метки в исходных файлах.....	132
3.4. Опции энкодеров.....	133
3.4.1. Опции, поддерживаемые h264_nvenc.....	133
3.4.2. Опции, поддерживаемые libx264.....	136
3.5. Task Manager.....	138
3.5.1. Назначение сервиса Task Manager.....	138
3.5.2. Переменные окружения Task Manager.....	138
3.5.3. Сбор метрик.....	143
3.6. Настройка профилей транскодирования.....	167
3.6.1. Настройка профилей транскодирования.....	167
3.6.2. Транскодирование видеофайлов в зависимости от типа глубины цвета с использованием CPU и GPU.....	167
3.6.3. Формат сообщения.....	168
3.7. Apache Kafka.....	211
3.8. Мониторинг.....	223

3.8.1. Функциональная схема	223
3.8.2. Трейсы	223
3.8.3. Метрики в Grafana	224
4. Жизненный цикл обработки медиаданных	237
4.1. Особенности работы Медиаплатформы.....	245
4.1.1. Информация о цветовом пространстве и диапазоне.....	245
5. REST API.....	247
5.1. Создание задачи на загрузку и обработку медиафайлов	247
5.2. Загрузка медиафайла одним файлом	249
5.3. Загрузка медиафайлов блоками	250
5.4. Запрос состояния загрузки.....	252
6. Взаимодействие между компонентами Медиаплатформы через Apache Kafka	255
6.1. Настройки профилей обработки исходного медиафайла	255
6.1.1. Настройка профилей транскодирования	255
6.1.2. Транскодирование видеофайлов в зависимости от типа глубины цвета с использованием CPU и GPU.....	255
6.1.3. Формат сообщения	256
6.2. Задание на загрузку медиафайла.....	299
6.3. Загрузка медиафайла	302
6.4. Задание на анализ исходного медиафайла	307
6.5. Результат анализа исходного медиафайла	311
6.6. Задание на обработку исходного медиафайла.....	317
6.7. Результат создания эскиза (thumbnail) медиафайла.....	353
6.8. Результат обработки исходного медиафайла.....	357
6.9. Сообщения о статусах обработки	369
6.10. Сообщения об ошибках	382
6.10.1. Составной ключ ошибки.....	382
6.10.2. Пример (ошибка 1010101)	383
6.10.3. Ошибки Медиаплатформы	383
6.11. Ограничения на загружаемое видео	392
7. Процедуры проверки работоспособности и действия при сбоях	394
7.1. Процедуры проверки работоспособности.....	394
7.1.1. Проверка работоспособности сервисов Медиаплатформы после развертывания .	394
7.1.2. Проверка работоспособности Медиаплатформы в процессе эксплуатации	395
7.2. Действия при сбоях в функционировании Медиаплатформы	397

7.2.1. Профилактические действия для предупреждения сбоев в работе Медиаплатформы	397
7.2.2. Основные положения для исследования проблем, связанных с работой Медиаплатформы	397
7.2.3. Известные проблемы в работе Медиаплатформы и методы их решения	399
7.2.4. Действия при неустановленной проблеме	401

Термины и определения

Термин	Определение
Apache Kafka (Kafka)	Распределённый программный брокер сообщений с открытым исходным кодом, разрабатываемый в рамках фонда Apache на языках Java и Scala.
API (Application Programming Interface)	Программный интерфейс, описание способов взаимодействия одной компьютерной программы с другими.
Bucket	Бакет, логическая сущность, которая помогает организовать хранение медиафайлов.
Frontend	Внешний по отношению к Медиаплатформе сервис, взаимодействующий с Медиаплатформой для получения, обработки, хранения и выдачи по запросу медиаконтента или других данных Медиаплатформы.
FPS (Frames per second)	Количество кадров в секунду в видеофайле.
GOP (group of pictures, группа изображений)	В кодировании видео группа изображений, или структура GOP, определяет порядок расположения внутрикадровых и межкадровых кадров. GOP - упорядоченная цепочка следующих друг за другом изображений в кодированном видеопотоке. Кадры объединяются в группы для целей межкадровой компрессии, без которой передача видеопотока по сети расходовала бы огромное время и трафик. Обнаружение нового GOP в сжатом видеопотоке означает, что декодеру не нужны никакие предыдущие кадры для декодирования следующих, и позволяет осуществлять быстрый поиск по видео.
I-кадр (Intra-coded frame)	Кадр с внутренним кодированием, или ключевой кадр видеофайла.

IP-адрес	Уникальный числовой идентификатор устройства в компьютерной сети, работающей по протоколу IP.
Kubernetes	Открытое программное обеспечение для оркестровки контейнеризированных приложений: автоматизации их развёртывания, масштабирования и координации в условиях кластера.
MD5	Алгоритм хеширования, 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.
NTP (Network Time Protocol)	Сетевой протокол для синхронизации внутренних часов компьютера с использованием сетей с переменной латентностью.
Offset	См. смещение.
P-кадр (Predicted frame)	Прогнозируемый кадр, который содержит информацию о том, что изменилось по сравнению с предыдущими кадрами видеофайла.
PAR (pixel aspect ratio)	Соотношение сторон пикселя.
Pod	Абстрактный объект Kubernetes, представляющий собой «обертку» для одного или группы контейнеров. Контейнеры в поде запускаются и работают вместе, имеют общие сетевые ресурсы и хранилище. Модули Медиаплатформы запускаются как поды Kubernetes.

REST API	Архитектурный подход, который устанавливает ограничения для API: как они должны быть устроены и какие функции поддерживать. Это позволяет стандартизировать работу программных интерфейсов, сделать их более удобными и производительными.
S3	Объектное хранилище. Его уникальность заключается в хранении огромного объема данных в исходном формате без иерархии и разбивки на отдельные каталоги. У хранилища S3 нет ограничений по масштабированию.
SaaS (программное обеспечение как услуга)	Модель обслуживания, при которой подписчикам предоставляется готовое прикладное программное обеспечение, полностью обслуживаемое провайдером. Поставщик в этой модели самостоятельно управляет приложением, предоставляя заказчикам доступ к функциям с клиентских устройств, как правило через мобильное приложение или веб-браузер.
SAR (sample aspect ratio)	Отношение количества пикселей по горизонтали к количеству пикселей по вертикали (эти цифры указаны в разрешении видеофайла).
Thumbnail	Эскиз видео, уменьшенная до небольших размеров копия изображения для компактного представления пользователю.
Trickmode	См. трикмод.
UNIX-time	Количество секунд, прошедших с полуночи (00:00:00 UTC) 1 января 1970 года.
URI (Uniform Resource Identifier)	Унифицированный указатель ресурса — система унифицированных адресов электронных ресурсов, или единообразный определитель местонахождения ресурса (файла).
URL (Uniform Resource Locator)	Адрес, который выдан уникальному ресурсу в интернете.

БД	База данных.
Base64	Стандарт кодирования двоичных данных при помощи только 64 символов ASCII. Алфавит кодирования содержит латинские символы A-Z, a-z, цифры 0-9.
Контейнер	Формат файла, определяющей распределение аудио, видео, а в некоторых случаях и текстовой информации внутри него.
Медиаплатформа	Интегрированная система обработки медиаконтента. Эта система позволяет: <ul style="list-style-type: none"> • проанализировать загруженный медиаконтент и оценить его качество; • обработать медиафайл и сохранить его в различном качестве и в нескольких форматах; • создать серию эскизов для ознакомительного просмотра; • извлечь аудиодорожку из загруженного медиафайла.
Медиафайл	Компьютерный файл, содержащий аудио- и видеoinформацию.
Под	См. pod.
Смещение	Метка в файле, с которой начинается загрузка чанка.
Сторонний сервис	См. Front.
Стрим	Прямой эфир на интернет-сервисе.
Транскодирование	Обработка медиафайлов, прямое цифровое преобразование одной кодировки в другую.
Трикмод	Функция показа соответствующего Thumbnail-а во время движения мышки на полосе прокрутки видеофайла.
Холст	Растровый холст, задающий размер и разрешение для цифровых изображений.

Чанк	Часть медиафайла для удобства его загрузки и обработки. Обычно файл делится на чанки одинакового размера, кроме последнего чанка.
------	---

1. Введение

Медиаплатформа — интегрированная система обработки медиаконтента.

Система позволяет:

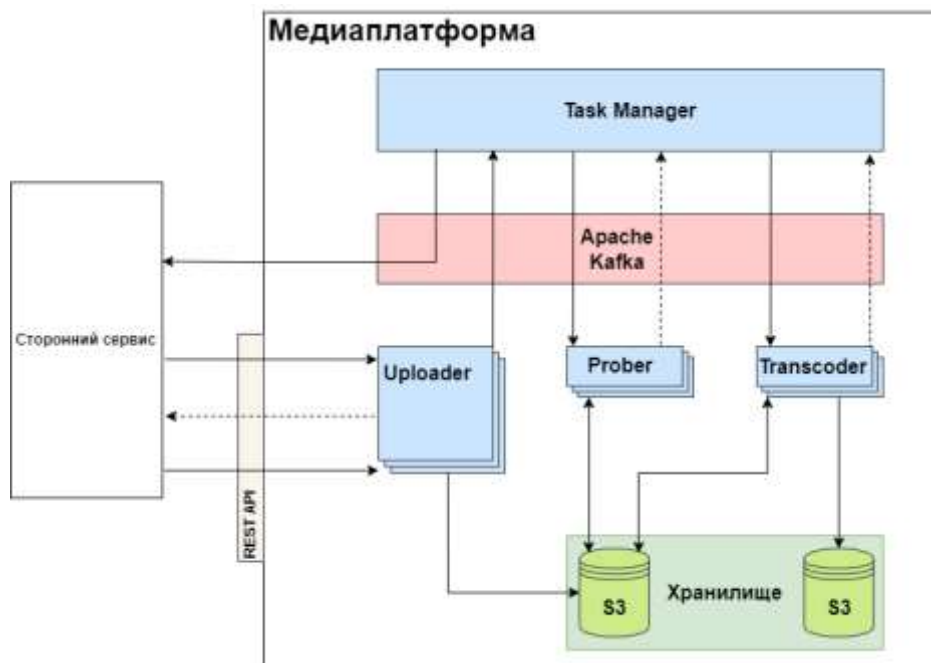
- проанализировать загруженный медиаконтент и оценить его качество;
- обработать медиафайл и сохранить его в различном качестве и в нескольких форматах;
- создать серию эскизов для ознакомительного просмотра;
- извлечь аудиодорожку из загруженного медиафайла.

Медиаплатформа имеет простой программный интерфейс (API), позволяющий подключить к ней существующие и развивающиеся сервисы работы с медиаконтентом. Компоненты системы можно развернуть как на стороне заказчика, так и получить по модели SaaS (программное обеспечение как услуга).

Для оркестровки системы приложений используется Kubernetes.

Настоящий документ содержит описание Медиаплатформы - её интерфейсов, компонентов, а также процедур, связанных с обработкой медиаконтента.

2. Архитектура Медиаплатформы



Микросервисы типа - stateless, данные в приложениях не сохраняются.

Краткое описание модулей платформы и их взаимодействий приведено в таблице

ниже.

Модуль	Описание
Task Manager	<p>Task Manager является центром реализации бизнес-логики Медиаплатформы и предназначен для оркестрации работы модулей системы. Через брокер сообщений Apache Kafka Task Manager взаимодействует со следующими модулями:</p> <ul style="list-style-type: none"> • Uploader – от модуля Uploader модуль Task Manager получает информацию о загрузке исходных медиафайлов; • Prober – после получения информации о загрузке медиафайлов Task Manager ставит для Prober задачу анализа загруженных медиафайлов; • Transcoder – после анализа загруженного Task Manager ставит модулю Transcoder задачу на обработку медиафайлов. <p>Также Task Manager транслирует в Kafka статусы обработки медиафайлов, которые доступны для считывания извне Медиаплатформы. Подробнее см. подраздел Task Manager.</p>
Uploader	<p>Uploader предназначен для приёма загружаемых пользовательских медиафайлов и сохранения их во внутреннем хранилище.</p> <p>Uploader взаимодействует со следующими модулями Медиаплатформы:</p> <ul style="list-style-type: none"> • хранилище S3 – после загрузки медиафайлов пользователем Uploader помещает полученные медиафайлы в хранилище; • Task Manager – Uploader посредством брокера сообщений Kafka уведомляет Task Manager о состоянии загрузки. <p>Подробнее см. подраздел Uploader.</p>

Prober	<p>Prober анализирует исходные медиафайлы и собирает необходимую информацию для формирования задания на дальнейшую обработку.</p> <p>Prober взаимодействует со следующими модулями системы:</p> <ul style="list-style-type: none"> • Task Manager – Prober посредством брокера задач Kafka получает от Task Manager задачи на анализ загруженных пользователями медиафайлы и возвращает результаты такого анализа; • хранилище – исходные медиафайлы для анализа Prober берёт из хранилища, в которое эти медиафайлы поместил модуль Uploader. <p>Подробнее см. подраздел Prober.</p>
Transcoder	<p>Transcoder выполняет обработку (транскодирование) медиафайлов после анализа, проведённого модулем Prober, на основе профилей обработки, настроенных в Task Manager.</p> <p>Transcoder взаимодействует со следующими модулями системы:</p> <ul style="list-style-type: none"> • Task Manager – Transcoder посредством брокера сообщений Kafka получает от Task Manager задачи на обработку исходного медиафайла и возвращает сведения о результатах такой обработки; • хранилище S3 – <ul style="list-style-type: none"> ○ исходный медиафайл для обработки Transcoder берёт из хранилища, в которое этот медиафайл поместил модуль Uploader; ○ обработанный медиафайл в зависимости от настроек Transcoder помещает в то же хранилище или использует другое. <p>Подробнее см. подраздел Transcoder.</p>
S3	<p>Объектное хранилище S3 (Object storage S3) — это облачный сервис, позволяющий хранить файлы любого типа и объема.</p> <p>Хранилище используется для:</p> <ul style="list-style-type: none"> • хранения исходных медиафайлов, загруженных пользователями; • размещения обработанных медиаданных для последующего просмотра пользователями. <p>В зависимости от настроек модулей системы можно использовать одно или два хранилища для загрузки исходного медиафайла и размещение обработанных медиаданных.</p> <p>С хранилищем взаимодействуют следующие модули системы:</p> <ul style="list-style-type: none"> • Uploader – модуль Uploader помещает в хранилище исходные медиафайлы, загруженные пользователем; • Prober – модуль Prober анализирует медиафайлы, сохранённые модулем Uploader; • Transcoder – модуль Transcoder считывает исходные медиафайлы и записывает обработанные файлы. <p>В зависимости от настроек это может быть одно и то же или два разных хранилища. Альтернативным вариантом по использованию S3 является файловая система (fs). Только для внутреннего использования.</p>
Apache Kafka	<p>Брокер сообщений, обеспечивающий взаимодействие модулей Медиаплатформы.</p>

3. Описание компонентов Медиаплатформы

3.1. Uploader

3.1.1. Назначение сервиса Uploader

Сервис Uploader выполняет загрузку пользовательских файлов в облачное S3 хранилище для последующей обработки различными сервисами.

3.1.2. Переменные окружения Uploader

В таблице ниже представлены переменные окружения, которые используются для настройки компонента Uploader.

Переменная	Тип данных	Формат	Диапазон возможных значений	Обязательно	Описание
Блок базовых настроек					
POD_NAME	String		Название пода всегда имеет значение uploader	Нет	Идентификатор пода, уникальное строковое значение. Используется при выборке заданий из БД для сервиса Uploader. Значение по умолчанию: отсутствует.
REST_API_LISTEN_ADDRESS	String	host:port		Нет	Адрес, который будет использован для REST API. Пример: 0.0.0.0:1337. Значение по умолчанию: localhost:1337.
LOG_LEVEL	Integer or String		<ol style="list-style-type: none"> 1. 0 или DEBUG 2. 1 или INFO 3. 2 или WARN 4. 3 или ERROR 5. 4 или FATAL 	Нет	Уровень логирования. Допустимые значения: <ul style="list-style-type: none"> • 0 или DEBUG - уровень отладки; записи журнала, необходимые для отладки приложения; • 1 или INFO - уровень информирования; важные действия в приложении; • 2 или WARN - уровень предупреждения;

			6. 7 или DISA BLED		<ul style="list-style-type: none"> 3 или ERROR - уровень ошибок; приложение продолжает работать; 4 или FATAL - критическая ошибка, после которой приложение не будет работать; 7 или DISABLED - записи в журнал не ведутся. <p>Значение по умолчанию: INFO.</p>
CFG_TYPE	String		<ol style="list-style-type: none"> yaml json ini env 	Нет	Тип конфигурационного файла.
HTTP_COMPRESS	Boolean		<ol style="list-style-type: none"> true false 	Нет	Параметр, который указывает на необходимость сжатия содержимого ответа в формат gzip.
HTTP_LOGS	Boolean		<ol style="list-style-type: none"> true false 	Нет	Параметр, который указывает на необходимость логирования входящих запросов и ответов.
CFG_FILE_PATH	String	/file_path		Нет	Путь до файла конфигурации.
LOST_TASK_SEARCH_TIMEOUT	Integer			Нет	Периодичность запросов на поиск потерянных задач и для перепосылки сообщений в Kafka. Первая перепосылка производится по истечении времени, заданного в переменной RESEND_NOTIFY_TIMEOUT.
LIMIT_TASK_SEARCH_RECORDS	Integer			Нет	Количество записей, возвращаемых за один запрос при поиске просроченных задач.

LIMIT_RESEND_NOTIFY_TASK_RECORDS	Integer			Нет	Количество записей, возвращаемых за один запрос при перепосылке сообщений, если Kafka была не доступна.
UPLOAD_TIMEOUT	Integer			Нет	Время в секундах, после которого задача будет считаться потерянной.
RESEND_NOTIFY_TIMEOUT	Integer			Нет	Время в секундах, после которого будет производиться попытка перепосылки сообщения в Kafka.
MAX_UPLOAD_FILE_SIZE	Integer			Нет	Максимальный размер загружаемого файла в байтах.
Блок метрик					
ZIPKIN_ENDPOINT_URL	Строка	http://path		Нет	Адрес подключения к Zipkin.
ZIPKIN_SAMPLE_FACTOR	Число с плавающей точкой		0.00 - 1.00	Нет	<p>Параметр, отвечающий за частоту трассировки.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> • 0.0 - никогда не выполнять трассировку; • 1.0 - всегда выполнять трассировку; • 0.01 - 1.00 выполнять трассировку в x*100% случаев.
Блок настроек, относящийся к настройкам подключения к базе данных					
DATABASE_TYPE	String		postgres	Да	<p>Тип используемой базы данных для хранения статусов заданий.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> • postgres - база данных PostgreSQL. <p>Настройки подключения к базе данных необходимо указать в зависимости от</p>

					выбранного типа базы данных. Значение по умолчанию: отсутствует.
DATABASE_SERVER_HOST	String	host		Да, если DATABASE_TYPE=postgres	IP-адрес или доменное имя сервера базы данных.
DATABASE_SERVER_PORT	Integer	port		Да, если DATABASE_TYPE=postgres	Порт сервера базы данных.
DATABASE_NAME	String			Да, если DATABASE_TYPE=postgres	Имя базы данных. База данных должна быть создана заранее. Структура таблиц в случае отсутствия какой-либо таблицы будет создана автоматически.
DATABASE_USERNAME	String			Да, если DATABASE_TYPE=postgres	Имя пользователя для доступа к базе данных.
DATABASE_PASSWORD	String			Да, если DATABASE_TYPE=postgres	Пароль для доступа к базе данных.
DATABASE_SCHEMA	String				Название схемы БД, в которой будут созданы таблицы для работы Uploader.
DATABASE_DSN_OPTIONS	String				Строка дополнительных параметров подключения к БД.
Блок настроек подключения к диспетчеру задач (очереди задач)					
KAFKA_BROKERS	String	host или host:port, несколько значений разделяются запятыми		Да	Список брокеров Kafka в виде значений, разделённых запятыми. Значение по умолчанию: отсутствует.

<p>KAFKA_TOPIC_TASK_MANAGER</p>	<p>String</p>		<p>Да</p>	<p>Имя топика Kafka, в который будут посылаться сообщения об этапах загрузки медиафайлов в Медиаплатформу:</p> <ul style="list-style-type: none"> • начало загрузки исходного медиафайла во временное хранилище компонента Uploader; • завершение загрузки медиафайла во временное хранилище компонента Uploader; • сбой загрузки медиафайла во временное хранилище компонента Uploader; • начало перемещения исходного медиафайла в постоянное хранилище Медиаплатформы для анализа; • завершение перемещения исходного медиафайла в постоянное хранилище Медиаплатформы для анализа. <p>Эти сообщения предназначены для считывания компонентом Task Manager.</p> <p>Значение по умолчанию: KAFKA_TOPIC_TASK_MANAGER.</p>
--	---------------	--	-----------	--

KAFKA_SEND_TIMEOUT	Integer		Значение должно быть меньше значения переменной PROXY_SEND_TIMEOUT	Нет	Время, в течение которого Uploader пытается послать сообщение в Kafka, в миллисекундах. PROXY_SEND_TIMEOUT задается в файле <code>nginx.conf</code> и по умолчанию составляет 60 сек.
Блок настроек подключения к хранилищу медиаданных					
S3_ENDPOINT_URL	String	host:port		Да	Адрес S3 хранилища в формате <code>http://<хост>:<порт></code> . Значение по умолчанию: отсутствует.
S3_REGION	String			Да	Название региона хранилища S3 (настройка S3). Значение, которое должно совпадать с аналогичным значением на стороне хранилища S3. Допустимо вводить любые значения, однако подразумевается, что здесь будет указан код страны (например, <code>ru-central</code>). Значение по умолчанию: отсутствует.
S3_BUCKET	String			Да	Имя контейнера (bucket) в хранилище S3. Значение по умолчанию: отсутствует.
S3_KEY_ID	String			Да	Логин для подключения к хранилищу S3. Значение по умолчанию: отсутствует.
S3_SECRET_KEY	String			Да	Пароль для подключения к хранилищу S3. Значение по умолчанию: Отсутствует.
S3_FORCE_PATH_STYLE	Boolean		1. true 2. false	Да	Позволяет использовать имя контейнера (bucket) в качестве доменного имени последнего уровня в строке подключения. Например, если

					<ul style="list-style-type: none"> адрес ресурса: http://test.ru/; имя контейнера (bucket): videos; настоящая настройка включена (true); <p>то полная строка подключения будет выглядеть следующим образом: https://videos.test.ru. Как видно из примера, название контейнера используется в качестве домена последнего уровня. Доступны следующие значения настройки:</p> <ul style="list-style-type: none"> true - настройка включена, имя контейнера используется в адресе подключения к хранилищу в качестве доменного имени последнего уровня; false - настройка выключена. <p>Значение по умолчанию: отсутствует.</p>
DISABLE_WRITE_S3	Boolean		<ol style="list-style-type: none"> true false 	Нет	<p>Параметр для отладки, запрещает или разрешает запись файла в хранилище S3.</p> <p>Значение по умолчанию: true, т.е. запись в S3 разрешена.</p>
S3_WAITER_MAX_DURATION	Integer			Нет	<p>Время ожидания доступности файла после загрузки в S3, в секундах.</p>
S3_WAITER_MIN_DELAY	Integer			Нет	<p>Минимальное время между повторными попытками (проверкой) доступности</p>

					файла после загрузки в S3, в секундах.
--	--	--	--	--	--

Переменные среды указываются в файле или файлах YAML. Это может быть как единый файл для всех компонентов Медиаплатформы, так и несколько файлов.

В зависимости от вариантов развёртывания переменные можно указать вручную или включить в настройки из других источников.

Ниже представлен пример настроек компонента Uploader.

env:

Блок базовых настроек

- name: POD_NAME

value: uploader

- name: LOG_LEVEL

value: debug

- name: CFG_TYPE

value: env

- name: HTTP_COMPRESS

value: false

- name: HTTP_LOGS

value: false

- name: CFG_FILE_PATH

value: /etc/mediaplatform/config.json

- name: LOST_TASK_SEARCH_TIMEOUT

value: 10000

- name: LIMIT_TASK_SEARCH_RECORDS

value: 1

- name: LIMIT_RESEND_NOTIFY_TASK_RECORDS

value: 50

- name: UPLOAD_TIMEOUT

value: 86400

- name: RESEND_NOTIFY_TIMEOUT

value: 120

- name: MAX_UPLOAD_FILE_SIZE

value: 10737418240

Точка подключения к API

- name: REST_API_LISTEN_ADDRESS
value: "0.0.0.0:1337"

Метрики

- name: ZIPKIN_ENDPOINT_URL
value: http://test.ru/api/v2/spans
- name: ZIPKIN_SAMPLE_FACTOR
value: 0.00

Настройки брокера сообщений

- name: NOTIFIERS
value: kafka
- name: KAFKA_BROKERS
value: 127.0.0.1:9093
- name: KAFKA_TOPIC
value: KAFKA_TOPIC_TASK_MANAGER
- name: KAFKA_SEND_TIMEOUT
value: 5000

Настройки базы данных

- name: DATABASE_TYPE
value: postgres
- name: DATABASE_SERVER_HOST
value: localhost
- name: DATABASE_SERVER_PORT
value: "5432"
- name: DATABASE_NAME
value: postgresql
- name: DATABASE_USERNAME

value: db_user
- name: DATABASE_PASSWORD
value: db_password
- name: DATABASE_SCHEMA
value: public
- name: DATABASE_DSN_OPTIONS
value: statement_cache_mode=describe
- name: DefaultS3WaiterMaxDuration
value: 20
- name: DefaultS3WaiterMinDelay
value: 5

Хранилище S3

- name: STORAGE_PROVIDER
value: s3
- name: S3_ENDPOINT_URL
value: https://test.ru:9000
- name: S3_REGION
value: ru-central
- name: S3_ID_KEY
value: s3_id_key
- name: S3_SECRET_KEY
value: s3_secret_key
- name: S3_BUCKET
value: test
- name: S3_FORCE_PATH_STYLE
value: true
- name: DISABLE_WRITE_S3
value: true

3.1.3. Сбор метрик Uploader

Отправителем метрик является модуль Uploader, входящий в состав Медиаплатформы.

Метрики собираются с помощью системы мониторинга и оповещения Prometheus и отображаются в виде графиков, счетчиков и в других представлениях с помощью инструмента для визуализации и анализа данных Grafana.

Отображаемый отрезок времени на всех графиках по умолчанию: 6 часов.

Все метрики выводятся для списка подов или конкретного пода.

3.1.3.1. Описание функциональных клавиш Grafana

- **namespace** – выбор окружения;
- **prober_instance** – поды Prober;
- **transcoder_instance** – поды Transcoder;
- **uploader_instance** – поды Uploader;
- **taskmanager_instance** – поды Task Manager.

Для всех функциональных клавиш, кроме **namespace**, значение по умолчанию – all.

3.1.3.2. Описание метрик Grafana

При выборе большого временного интервала или частом перезапуске подов на графиках и счётчиках относительно него могут отображаться все запущенные поды и их состояние. Для мониторинга последних следует выбирать более короткий интервал. Данное поведение позволяет отслеживать состояние конкретного пода в конкретный момент времени.

Разрывы на графиках и отсутствие информации возможны в случае, если после старта сервиса метрика ни разу не отсылалась или сервис не был запущен или не работает в следствии системной ошибки.

Метрики, которые регистрируют единичные всплески, могут быть не видны на графиках, где для его построения используется функции `rate` или `increase`, которые возвращают скорость изменения метрики на заданном интервале, при этом первая делит полученное значение на этот интервал. Для регистрации и отслеживания подобных изменений следует использовать счётчики.

	Наименование дашборда	Название метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
1	Overall 1	Uploader - общее количество ошибок по текстовым кодам после старта	Счетчик / Таблица	Общее количество ошибок по внутренним кодам ошибок с маппингом кодовых на текстовые лейблы с момента последней перезагрузки.	<ul style="list-style-type: none"> code 	<p>Метрика: mp_uploader_errors</p> <ul style="list-style-type: none"> {{code_text}}: sum(mp_uploader_errors{code!="0", namespace=~"\$namespace", pod=~"\$uploader_instance"}) by (code, code_text)
2	Overall 1	Uploader - количество ошибок по внутренним кодам (кол-во/с)	График	Общее количество ошибок по внутренним кодам ошибок.	<ul style="list-style-type: none"> code text_code 	<p>Метрика: mp_uploader_errors</p> <ul style="list-style-type: none"> Total: sum(sum(rate(mp_uploader_errors{code!="0", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval])) by (code)) {{code}}: sum(rate(mp_uploader_errors{code!="0", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval])) by (code)

3	Overall	Uploader - статус подключения к сервисам	График	<p>Статус связи с сервисами:</p> <ul style="list-style-type: none"> • 1 - состояние инициализации процесса загрузки; 0 - состояние инициализации отгрузки 	<ul style="list-style-type: none"> • exported_service: <ul style="list-style-type: none"> ○ kafka ○ postgres ○ s3 	<p>Метрика: mp_uploader_connection_status</p> <ul style="list-style-type: none"> • {{exported_service}} - {{pod}}: mp_uploader_connection_status{exported_service="kafka", namespace=~"\$namespace", pod=~"\$uploader_instance"} • {{exported_service}} - {{pod}}: mp_uploader_connection_status{exported_service="postgres", namespace=~"\$namespace", pod=~"\$uploader_instance"} • {{exported_service}} - {{pod}}: mp_uploader_connection_status{exported_service="s3", namespace=~"\$namespace", pod=~"\$uploader_instance"}
---	---------	--	--------	--	--	---

	Наименование дашборда	Название метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
				<p>е т .</p> <p>Все графики суммируются, каждый следующий будет выглядеть, как своё значение плюс предыдущее.</p>		
4	Overall	Uploader - статус подключения	Измеритель	Показывает текущее состояние подключения к соответствующему сервису.	<ul style="list-style-type: none"> • exported_service: <ul style="list-style-type: none"> ○ kafka ○ postgres ○ s3 	<p>Метрика: mp_uploader_connection_status</p> <ul style="list-style-type: none"> • kafka: mp_uploader_connection_status{exported_service="kafka", namespace=~"\$namespace", pod=~"\$uploader_instance"} • postgres: mp_uploader_connection_status{exported_service="postgres", namespace=~"\$namespace", pod=~"\$uploader_instance"} • s3: mp_uploader_connection_status{exported_service="s3", namespace=~"\$namespace", pod=~"\$uploader_instance"}

	Наименование дашборда	Название метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
5	Uploader	Количество запросов, не прошедших валидацию (кол-во/с)	График	Средняя экстраполированная скорость изменения количества запросов на загрузку, не прошедших валидацию.	type: <ul style="list-style-type: none"> unknown unexpected 	Метрика: mp_uploader_unhandled_messages <ul style="list-style-type: none"> Total: <pre>sum(rate(mp_uploader_unhandled_messages{pod=~"\$uploader_instance"}[\$__rate_interval]) >=0 or on() vector(0))</pre>
6	Uploader	Кол-во неверных запросов после старта	Счетчик	Количество запросов на загрузку, не прошедших валидацию за время работы сервиса с момента последней перезагрузки.	type: <ul style="list-style-type: none"> unknown unexpected 	Метрика: mp_uploader_unhandled_messages <ul style="list-style-type: none"> Validation Failed: <pre>sum(mp_uploader_unhandled_messages{pod=~"\$uploader_instance"}) >=0 or on() vector(0)</pre>

	Наименование дашборда	Название метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
7	Uploader	Количество загруженных файлов (кол-во/с)	График	Средняя экстраполированная скорость изменения величины общих загрузок файлов за определенный период. На графике представлено общее кол-во успешных операций, не успешных, а также общее кол-во этих операций.	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ success ○ failed 	<p>Метрика: mp_uploader_uploaded_files</p> <ul style="list-style-type: none"> • Total: <pre>sum(rate(mp_uploader_uploaded_files{namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]) >=0 or on() vector(0))</pre> • Success: <pre>sum(rate(mp_uploader_uploaded_files{status="success", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval])) >=0 or on() vector(0) + sum(rate(mp_uploader_uploaded_files{status="success", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]) >=0 or on() vector(0))</pre> • Failed: <pre>sum(rate(mp_uploader_uploaded_files{status="failed", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]) >=0 or on() vector(0) + sum(rate(mp_uploader_uploaded_files{status="failed", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]) >=0 or on() vector(0))</pre>

	Наименование дашборда	Название метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
8	Uploader	Общее количество загруженных файлов после старта	Счетчик	Количество успешных загрузок, неуспешных (в том числе незавершенные объединения чанков) и общее количество, накопленное значение за время работы сервиса с момента последней перезагрузки.	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ success ○ failed 	<p>Метрика: mp_uploader_uploaded_files</p> <ul style="list-style-type: none"> • Total: <pre>sum(mp_uploader_uploaded_files{namespace=~"\$namespace", pod=~"\$uploader_instance"}) >=0 or on() vector(0)</pre> • Success: <pre>sum(mp_uploader_uploaded_files{status="success", namespace=~"\$namespace", pod=~"\$uploader_instance"}) >=0 or on() vector(0) + sum(mp_uploader_uploaded_files{status="success", namespace=~"\$namespace", pod=~"\$uploader_instance"}) >=0 or on() vector(0)</pre> • Failed: <pre>sum(mp_uploader_uploaded_files{status="failed", namespace=~"\$namespace", pod=~"\$uploader_instance"}) >=0 or on() vector(0) + sum(mp_uploader_uploaded_files{status="failed", namespace=~"\$namespace", pod=~"\$uploader_instance"}) >=0 or on() vector(0)</pre>

	Наименование дашборда	Название метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
9	Uploader	Количество загруженных чанков (кол-во/с)	График	Средняя экстраполированная скорость изменения величины общих загрузок чанков за определенный период. На графике представлено общее количество успешных операций, не успешных, а также общее количество этих операций.	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ success ○ failed 	<p>Метрика: mp_uploader_uploaded_chunks</p> <ul style="list-style-type: none"> • Total: <pre>sum(rate(mp_uploader_uploaded_chunks{namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]) >=0 or on() vector(0))</pre> • Success: <pre>sum(rate(mp_uploader_uploaded_chunks{status="success", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]) >=0 or on() vector(0) + sum(rate(mp_uploader_uploaded_chunks{status="success", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]) >=0 or on() vector(0))</pre> • Failed: <pre>sum(rate(mp_uploader_uploaded_chunks{status="failed", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]) >=0 or on() vector(0) + sum(rate(mp_uploader_uploaded_chunks{status="failed", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]) >=0 or on() vector(0))</pre>

	Наименование дашборда	Название метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
10	Uploader	Общее количество загруженных чанков	Счетчик	Количество успешных загрузок, неуспешных и общее количество, накопленное значение за время работы сервиса с момента последней перезагрузки.	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ OK ○ FAILED 	<p>Метрика: mp_uploader_uploaded_chunks</p> <ul style="list-style-type: none"> • Total: <pre>sum(mp_uploader_uploaded_chunks{namespace=~"\$namespace", pod=~"\$uploader_instance"}) >=0 or on() vector(0)</pre> • Success: <pre>sum(mp_uploader_uploaded_chunks{status="OK", namespace=~"\$namespace", pod=~"\$uploader_instance"}) >=0 or on() vector(0) + sum(mp_uploader_uploaded_chunks{status="success", namespace=~"\$namespace", pod=~"\$uploader_instance"}) >=0 or on() vector(0)</pre> • Failed: <pre>sum(mp_uploader_uploaded_chunks{status="FAILED", namespace=~"\$namespace", pod=~"\$uploader_instance"}) >=0 or on() vector(0) + sum(mp_uploader_uploaded_chunks{status="failed", namespace=~"\$namespace", pod=~"\$uploader_instance"}) >=0 or on() vector(0)</pre>

	Наименование дашборда	Название метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
1	Uploader	Средняя пропускная способность клиента - >Uploader, 0.9 квантиль (Мгб/с)	График	Средняя пропускная способность загрузки данных от клиента до Uploader, мегабит в секунду. Имеется в виду средняя скорость загрузки видеоконтента, рассчитанная путем деления размера одного чанка на полное время загрузки чанка, усредненная для всех чанков за заданный промежуток времени.	quantile: <ul style="list-style-type: none"> 0.9 	<p>Метрика: Throughput_Mbps_HTTP_Upload, mp_uploader_throughput_mbps_http</p> <ul style="list-style-type: none"> quantile - 0.9 - {{pod}}: Throughput_Mbps_HTTP_Upload{quantile="0.9", namespace=~"\$namespace", pod=~"\$uploader_instance"} quantile - 0.9 - {{pod}}: mp_uploader_throughput_mbps_http{quantile="0.9", namespace=~"\$namespace", pod=~"\$uploader_instance"}

1 2	Uploader	Средняя пропускная способность загрузки данных из Uploader в S3 по заданному квантилю, мегабит в секунду. Имеется в виду средняя скорость загрузки видеоконтента, рассчитанная путем деления размера одного чанка на полное время загрузки чанка, усредненная для всех чанков за заданный промежуток времени.	График	Средняя пропускная способность загрузки данных из Uploader в S3 по заданному квантилю, мегабит в секунду. Имеется в виду средняя скорость загрузки видеоконтента, рассчитанная путем деления размера одного чанка на полное время загрузки чанка, усредненная для всех чанков за заданный промежуток времени.	<ul style="list-style-type: none"> quantile: <ul style="list-style-type: none"> 0.9 	<p>Метрика: Throughput_Mbps_S3_Upload, mp_uploader_throughput_mbps_s3</p> <ul style="list-style-type: none"> quantile - 0.9 - {{pod}}: Throughput_Mbps_S3_Upload{quantile="0.9", namespace=~"\$namespace", pod=~"\$uploader_instance"} quantile - 0.9 - {{pod}}: mp_uploader_throughput_mbps_s3{quantile="0.9", namespace=~"\$namespace", pod=~"\$uploader_instance"}
1 3	Uploader	Среднее время выполнения	График	Среднее время выполнения	<ul style="list-style-type: none"> code: <ul style="list-style-type: none"> success 	<p>Метрика: mp_uploader_http_time</p> <ul style="list-style-type: none"> {{code}} - Success: sum by (code) ((rate(mp_uploader_http_time_sum{code=~"2.*", namespace=~"\$namespace",

Наименование дашборда	Название метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
	лнения http-запроса (с)		HTTP-запроса. Запросы объединены в группы, к которым относятся все запросы с кодами - 2.*, к неуспешным - 4.*, 5.*	<ul style="list-style-type: none"> s: 20 0, 20 1, 21 1 o faile d: 40 0, 50 0 	<pre>pod=~"\$uploader_instance" }[\$__rate_interval])) / sum by (code) ((rate(mp_uploader_http_time_count{code=~" 2.*", namespace=~"\$namespace", pod=~"\$uploader_instance" }[\$__rate_interval])) • {{code}} - Failed: sum by (code) (rate(mp_uploader_http_time_sum{code=~"4. * 5.*", namespace=~"\$namespace", pod=~"\$uploader_instance" }[\$__rate_interval])) / sum by (code) (rate(mp_uploader_http_time_count{code=~"4 .* 5.*", namespace=~"\$namespace", pod=~"\$uploader_instance" }[\$__rate_interval]))</pre>
14	Uploader	Гистограмма	Распределение времени ответа на запрос клиента: модуль Uploader успешно создал новую задачу на загрузку медиафайла.	<ul style="list-style-type: none"> • le • code: <ul style="list-style-type: none"> o 20 1 • path_pattern: <ul style="list-style-type: none"> o /upload_create_task 	<p>Метрика: mp_uploader_http_time_bucket</p> <ul style="list-style-type: none"> • {{le}}: sum by (le) <pre>(increase(mp_uploader_http_time_bucket{METHOD="POST",code="201",path_pattern="/upload_create_task",le=~"[0-9].+",namespace=~"\$namespace",pod=~"\$uploader_instance" }[\$__rate_interval]))</pre>

	Наименование дашборда	Название метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
15	Uploader	Время выполнения http-запроса загрузки чанка (гистограмма, code="200") (колво)	Гистограмма	Распределение времени ответа на запрос клиента: модуль Uploader успешно вернул запрошенные данные: информацию о медиафайле или о том, что чанк успешно загружен.	<ul style="list-style-type: none"> • le • code: <ul style="list-style-type: none"> ○ 200 • path_pattern: <ul style="list-style-type: none"> ○ /upload_chunk 	<p>Метрика: mp_uploader_http_time_bucket</p> <ul style="list-style-type: none"> • {{le}}: sum by (le) (increase(mp_uploader_http_time_bucket{METHOD="PUT",code="200",path_pattern="/uploader/v1/upload_chunk",le=~"[0-9.]+" , namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]))
16	Uploader	Время выполнения http-запроса сливания чанков (гистограмма, code="211") (колво)	Гистограмма	Распределение времени ответа на запрос клиента: модуль Uploader успешно загрузил последний чанк, загрузка файла завершена в результате запроса клиента.	<ul style="list-style-type: none"> • le • code: <ul style="list-style-type: none"> ○ 211 • path_pattern: <ul style="list-style-type: none"> ○ /upload_chunk 	<p>Метрика: mp_uploader_http_time_bucket</p> <ul style="list-style-type: none"> • {{le}}: sum by (le) (increase(mp_uploader_http_time_bucket{METHOD="PUT",code="211",path_pattern="/uploader/v1/upload_chunk",le=~"[0-9.]+" , namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]))

	Наименование дашборда	Название метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
17	Uploader	Количество принятых и загруженных байт в S3 (бт/с)	График	Количество принятых и загруженных байт в S3.	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ read ○ write 	<p>Метрика: mp_uploader_s3_bytes</p> <ul style="list-style-type: none"> • {{status}}: sum by (status) (rate(mp_uploader_s3_bytes{namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]) >=0 or on() vector(0))
18	Uploader	Количество принятых и загруженных мегабайт в S3 (мгб)	Счетчик	Количество принятых и загруженных байт в S3.	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ read ○ write 	<p>Метрика: mp_uploader_s3_bytes</p> <ul style="list-style-type: none"> • Received: (sum(mp_uploader_s3_bytes{op_type: "read", namespace=~"\$namespace", pod=~"\$uploader_instance"}) >=0 or on() vector(0))/1e+6 • Uploaded: (sum(mp_uploader_s3_bytes{op_type: "write", namespace=~"\$namespace", pod=~"\$uploader_instance"}) >=0 or on() vector(0))/1e+6

	Наименование дашборда	Название метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
19	Uploader	Среднее время выполнения запроса к БД (с)	График	Среднее время выполнения SQL-запроса к БД.	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ success ○ failed 	<p>Метрика: mp_uploader_pg_response_time</p> <ul style="list-style-type: none"> • Success: <pre>(sum(rate(mp_uploader_pg_response_time_sum{status="success", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval])) / sum(rate(mp_uploader_pg_response_time_count{status="success", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval])) >=0 or on() vector(0))</pre> • Failed: <pre>(sum(rate(mp_uploader_pg_response_time_sum{status="failed", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval])) / sum(rate(mp_uploader_pg_response_time_count{status="failed", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval])) >=0 or on() vector(0))</pre>
20	Uploader	Время выполнения запроса к БД удаления чанков (гистограмма) (колво)	Гистограмма	Время выполнения успешного SQL-запроса к БД; удалить чанки.	<ul style="list-style-type: none"> • le • query_type: <ul style="list-style-type: none"> ○ delete • status: <ul style="list-style-type: none"> ○ success • table: <ul style="list-style-type: none"> ○ chunks 	<p>Метрика: mp_uploader_pg_response_time_bucket</p> <ul style="list-style-type: none"> • {{le}}: sum by (le) <pre>(increase(mp_uploader_pg_response_time_bucket{query_type="delete", status="success", table="chunks", le=~"[0-9].+", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]))</pre>

	Наименование дашборда	Название метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
2 1	Uploader	Время выполнения запроса к БД добавления чанков (гистограмма) (колво)	Гистограмма	Время выполнения успешного SQL-запроса к БД: добавить чанки.	<ul style="list-style-type: none"> • le • query_type: <ul style="list-style-type: none"> ○ insert • status: <ul style="list-style-type: none"> ○ success • table: <ul style="list-style-type: none"> ○ chunks 	<p>Метрика: mp_uploader_pg_response_time_bucket</p> <ul style="list-style-type: none"> • {{le}}: sum by (le) (increase(mp_uploader_pg_response_time_bucket{query_type="insert", status="success", table="chunks", le=~"[0-9.]+" , namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]))
2 2	Uploader	Время выполнения запроса к БД выбора задач (гистограмма) (колво)	Гистограмма	Время выполнения успешного SQL-запроса к БД: выбрать задачи.	<ul style="list-style-type: none"> • le • query_type: <ul style="list-style-type: none"> ○ select • status: <ul style="list-style-type: none"> ○ success • table: <ul style="list-style-type: none"> ○ tasks 	<p>Метрика: mp_uploader_pg_response_time_bucket</p> <ul style="list-style-type: none"> • {{le}}: sum by (le) (increase(mp_uploader_pg_response_time_bucket{query_type="select", status="success", table="tasks", le=~"[0-9.]+" , namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]))

	Наименование дашборда	Название метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
2 3	Uploader	Время выполнения запроса к БД обновления задачи (гистограмма) (колво)	Гистограмма	Время выполнения успешного SQL-запроса к БД: обновить информацию о задаче.	<ul style="list-style-type: none"> le query_type: <ul style="list-style-type: none"> update status: <ul style="list-style-type: none"> success table: <ul style="list-style-type: none"> tasks 	Метрика: mp_uploader_pg_response_time_bucket <ul style="list-style-type: none"> {{le}}: sum by (le) (increase(mp_uploader_pg_response_time_bucket{query_type="update", status="success", table="tasks", le=~"[0-9.]+" , namespace=~"\$namespace", pod=~"\$uploader_instance"})[\$__rate_interval])
2 4	Uploader	Сообщения, отправленные в Kafka (колво/с)	График	Количество сообщений, отправленных в Kafka.	-	Метрика: mp_uploader_kafka_writed_messages <ul style="list-style-type: none"> Total: sum(rate(mp_uploader_kafka_writed_messages{namespace=~"\$namespace", pod=~"\$uploader_instance"})[\$__rate_interval]) >=0 or on() vector(0)
2 5	Uploader	Сообщения, отправленные в Kafka	Счетчик	Общее количество сообщений, отправленных в Kafka с момента перезагрузки модуля.	-	Метрика: mp_uploader_kafka_writed_messages <ul style="list-style-type: none"> Total: sum(mp_uploader_kafka_writed_messages{namespace=~"\$namespace", pod=~"\$uploader_instance"}) >=0 or on() vector(0)

	Наименование дашборда	Название метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
26	Uploader	Среднее время выполнения задания к S3 (с)	График	Среднее время выполнения операции создания задания на загрузку чанка, загрузки чанка и слияния чанков в S3.	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ success ○ failed 	<p>Метрика: mp_uploader_s3_response_time</p> <ul style="list-style-type: none"> • Success: <pre>(sum(rate(mp_uploader_s3_response_time_sum{status="success", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval])) / sum(rate(mp_uploader_s3_response_time_count{status="success", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval])) >=0 or on() vector(0))</pre> • Failed: <pre>(sum(rate(mp_uploader_s3_response_time_sum{status="failed", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval])) / sum(rate(mp_uploader_s3_response_time_count{status="failed", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval])) >=0 or on() vector(0))</pre>
27	Uploader	Время выполнения: создание задания на загрузку чанка (гистограмма) (кол-во)	Гистограмма	Время выполнения успешной операции создания задания на загрузку чанка в S3.	<ul style="list-style-type: none"> • le • op_type: <ul style="list-style-type: none"> ○ chunk_create • status: <ul style="list-style-type: none"> ○ success 	<p>Метрика: mp_uploader_s3_response_time_bucket</p> <ul style="list-style-type: none"> • {{le}}: sum by (le) <pre>(increase(mp_uploader_s3_response_time_bucket{op_type="chunk_create",status="success",le=~"[0-9].+", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]))</pre>

	Наименование дашборда	Название метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
28	Uploader	Время выполнения: загрузка чанка в S3 (гистограмма) (кол-во)	Гистограмма	Время выполнения успешной операции загрузки чанка в S3.	<ul style="list-style-type: none"> • le • op_type: <ul style="list-style-type: none"> ○ chunk_upload • status: <ul style="list-style-type: none"> ○ success 	<p>Метрика: mp_uploader_s3_response_time_bucket</p> <ul style="list-style-type: none"> • {{le}}: sum by (le) (increase(mp_uploader_s3_response_time_bucket{op_type="chunk_upload",status="success",le=~"[0-9.]+" , namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]))
29	Uploader	Время выполнения: слияние чанков в S3 (гистограмма) (кол-во)	Гистограмма	Время выполнения успешной операции слияния чанков в S3.	<ul style="list-style-type: none"> • le • op_type: <ul style="list-style-type: none"> ○ task_merge • status: <ul style="list-style-type: none"> ○ success 	<p>Метрика: mp_uploader_s3_response_time_bucket</p> <ul style="list-style-type: none"> • {{le}}: sum by (le) (increase(mp_uploader_s3_response_time_bucket{op_type="task_merge",status="success",le=~"[0-9.]+" , namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]))

	Наименование дашборда	Название метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
30	Uploader	Статусы загрузки видео-файлов (колво/с)	График	Статусы загрузки видео-файлов: загрузка начата, загрузка закончена и ошибка загрузки.	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ started ○ finished ○ failed 	<p>Метрика: mp_uploader_video_status</p> <ul style="list-style-type: none"> • Started: <pre>sum(rate(mp_uploader_video_status{status="started", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]) >=0 or on() vector(0))</pre> • Finished: <pre>sum(rate(mp_uploader_video_status{status="finished", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]) >=0 or on() vector(0))</pre> • Failed: <pre>sum(rate(mp_uploader_video_status{status="failed", namespace=~"\$namespace", pod=~"\$uploader_instance"}[\$__rate_interval]) >=0 or on() vector(0))</pre>
31	Uploader	Статусы загрузки видео-файлов после старта (колво)	Счетчик	Общее количество заданий на загрузку в состоянии начала, завершения, ошибки слияния чанков.	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ started ○ finished ○ failed 	<p>Метрика: mp_uploader_video_status</p> <ul style="list-style-type: none"> • Started: <pre>sum(mp_uploader_video_status{status="started", namespace=~"\$namespace", pod=~"\$uploader_instance"}) >= 0 or on() vector(0)</pre> • Finished: <pre>sum(mp_uploader_video_status{status="finished", namespace=~"\$namespace", pod=~"\$uploader_instance"}) >= 0 or on() vector(0)</pre> • Failed: <pre>sum(mp_uploader_video_status{status="failed", namespace=~"\$namespace", pod=~"\$uploader_instance"}) >= 0 or on() vector(0)</pre>

Наименование дашборда	Название метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных	
3 2 1	Overall	Кол-во заданий после старта	Счетчик	Кол-во заданий после старта по статусам : started, finished, failed	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ started ○ finished ○ failed 	<p>Метрика: mp_uploader_video_status</p> <ul style="list-style-type: none"> • Started: <pre>sum(mp_uploader_video_status{status="started", namespace=~"\$namespace", pod=~"\$uploader_instance"}) >=0 or on() vector(0)</pre> • Finished: <pre>sum(mp_uploader_video_status{status="finished", namespace=~"\$namespace", pod=~"\$uploader_instance"}) >=0 or on() vector(0)</pre> • Failed: <pre>sum(mp_uploader_video_status{status="failed", namespace=~"\$namespace", pod=~"\$uploader_instance"}) >=0 or on() vector(0)</pre>

3.1.3.3. Описание метрик Prometheus

Ниже приведена таблица с описанием метрик Prometheus для модуля Uploader.

- histogram - графическое представление данных в виде графика, слева, по оси Y, количество значений, попавших в бакет, справа, легенда (значения le) и размерность. Граничные значения (le) в гистограмме - это диапазоны значений, на которые разбиваются измерения. Гистограмма с граничными значениями позволяет нам увидеть распределение значений внутри каждого диапазона и оценить, насколько часто значения попадают в каждый из диапазонов. Если один из графиков начинает стремиться к нулю, значит значения в него не попадают.
- gauge - масштабируемый диалоговый индикатор, который отображает текущее значение метрики в процентах или числах. Большим преимуществом Gauge в Grafana является то, что он может легко приспособливаться к разным значениям, чтобы обеспечить оптимальное отображение информации на дашборде: может быть использован для отображения любых числовых данных, таких как счётчики, процентные доли, абсолютные значения и т.д., т.к. gauge измеряет числовое значение в определенный момент времени, а не считает скорость изменения значений, как Counter. Gauge в Grafana можно отображать не только в виде графика, но и в виде

других типов визуализации, таких как текст, таблицы или специально созданные плагины.

- `counter` - счетчик, который увеличивается на каждое новое событие. Счетчики обычно могут только увеличиваться, но не уменьшаться, и сброс в ноль может потребоваться вручную.
- `summary` - собирает информацию о распределении значений заданной величины. Это позволяет собирать статистику о количестве значений, их среднем, медиане, процентилях и т.д. Для каждого `Summary Prometheus` собирает образцы (`samples`) для значений, которые необходимо агрегировать. Для каждого образца указывается значение и метка времени, когда оно было собрано. Затем `Prometheus` вычисляет ряд статистических значений на основе этих образцов, включая сумму всех значений, количество собранных образцов и среднее значение. В `Grafana` представление `Summary` может использоваться для создания графиков и диаграмм, показывающих распределение значений величины. Графики `Summary` могут быть полезны для анализа производительности приложения, когда необходимо определить, какое количество запросов было выполнено за определенный период времени, и как быстро были выполнены запросы.

	Наименование метрики	Тип	Метки	Описание	Событие
1	mp_uploader_throughput_mbps_http	summary	<ul style="list-style-type: none"> • quantile: <ul style="list-style-type: none"> ○ 0.9 • sum • count 	<p>Средняя пропускная способность загрузки данных от клиента до Uploader, мегабит в секунду (Mbit/s). Имеется в виду средняя скорость загрузки видеоконтента, рассчитанная путем деления размера одного чанка на полное время загрузки чанка, усредненная для всех чанков за заданный промежуток времени.</p> <ul style="list-style-type: none"> • sum - сумма всех значений; • count - количество обновлений метрик. 	Начало или окончание загрузки данных от клиента до Uploader для 90% наиболее вероятных событий загрузки этих данных.

2	mp_uploader_throughput_mbps_s3	summary	<ul style="list-style-type: none"> • quantile: <ul style="list-style-type: none"> ○ 0.9 • sum • count 	<p>Средняя пропускная способность загрузки данных из Uploader в S3 по заданному квантилю, мегабит в секунду (Mbit/s). Имеется в виду средняя скорость загрузки видеоконтента, рассчитанная путем деления размера одного чанка на полное время загрузки чанка, усредненная для всех чанков за заданный промежуток времени.</p> <ul style="list-style-type: none"> • sum - сумма всех значений; • count - количество обновлений метрик. 	Начало или окончание загрузки данных из Uploader в S3 для 90% наиболее вероятных событий загрузки этих данных.
3	mp_uploader_s3_bytes	counter	<ul style="list-style-type: none"> • status <ul style="list-style-type: none"> ○ write ○ read 	<p>Количество принятых и загруженных байт в S3.</p> <ul style="list-style-type: none"> • write - принято от клиента до Uploader; • read - отправлено от Uploader в S3. 	Зафиксирована попытка записи данных в хранилище S3.

4	mp_uploader_connection_status	gauge	<ul style="list-style-type: none"> • service: <ul style="list-style-type: none"> ○ Kafka ○ Postgres ○ S3 	<p>Статус связи с сервисами.</p> <ul style="list-style-type: none"> • 1 - соединен и присутствует; • 0 - соединен и отсутствует. 	Изменилось состояние связи с сервисами.
---	--------------------------------------	-------	---	--	---

5	mp_uploader_http_time	histogram	<ul style="list-style-type: none"> • METHOD <ul style="list-style-type: none"> ○ POST ○ PUT • code <ul style="list-style-type: none"> ○ 201 ○ 200 ○ 211 ○ 400 ○ 500 • path_pattern <ul style="list-style-type: none"> ○ /uploader/v1/upload_chunk ○ /uploader/v1/upload_createtask • le <ul style="list-style-type: none"> ○ 1 ○ 2.4839607111043738 ○ 6.170060814310146 ○ 15.326188647871062 ○ 38.06965045228558 ○ 94.56351600895424 ○ 234.89205847013176 ○ 583.4626445902387 ○ 1449.2982855592077 ○ 3599.999999999999 ○ +Inf • sum • count • bucket 	<p>Время выполнения http-запроса.</p> <ul style="list-style-type: none"> • sum - сумма всех значений; • count - количество обновлений метрик; • bucket - счетчик гистограммы (интервал), куда записывается значение; • le - граничащее значения, указывающие на то, насколько увеличится показатель гистограммы. • + inf - бесконечность; • path_pattern - endpoint, на который клиент отправляет запрос. <p>Коды ответов на http-запросы:</p> <ul style="list-style-type: none"> • Код 201 (Created) 	<p>Появился новый http-запрос от клиента, на который должен поступить ответ, возникающий в процессе загрузки медиафайла.</p>
---	-----------------------	-----------	---	--	--

				<p>означает, что модуль Uploader успешно создал новую задачу на загрузку медиафайла в результате запроса клиента.</p> <ul style="list-style-type: none"> • Код 200 (OK) означает, что модуль Uploader успешно обработал запрос клиента и вернул запрошенные данные: информацию о медиафайле или о том, что чанк успешно загружен. • Код 211 (Content Retrieval Successful) означает, что модуль Uploader успешно загрузил последний чанк, загрузка 	
--	--	--	--	--	--

				<p>файла завершена в результат е запроса клиента.</p> <ul style="list-style-type: none">• Код 400 (Bad Request) означает, что модуль Uploader не может обработат ь запрос из-за некоррект ных данных, отправлен ных клиентом (заданы неверные параметр ы).• Код 500 (Internal Server Error) означает, что модуль Uploader столкнулс я с ошибкой во время попытки выполнит ь запрос клиента. Это может быть вызвано различны ми причинам	
--	--	--	--	--	--

				и, такими как ошибки в коде сервера, проблемы с базой данных, доступностью хранилища S3 или другие технические проблемы.	
6	mp_uploader_kafka_writed_messages	counter	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ failed ○ success 	<p>Количество сообщений, отправленных в Kafka с момента работы сервиса.</p> <ul style="list-style-type: none"> • failed - неуспешно; • success - успешно. 	Зафиксирована попытка отправки сообщения в Kafka.

7	mp_uploader_pg_response_time	histogram	<ul style="list-style-type: none"> • query_type: <ul style="list-style-type: none"> ○ delete ○ insert ○ select ○ update • status: <ul style="list-style-type: none"> ○ success ○ failed • table: <ul style="list-style-type: none"> ○ chunks ○ tasks • le: <ul style="list-style-type: none"> ○ 0.01 ○ 0.05 ○ 0.1 ○ 0.5 ○ 1 ○ 5 ○ 10 ○ +Inf • sum • count 	<p>Время выполнения sql-запроса к БД.</p> <ul style="list-style-type: none"> • sum - сумма всех значений; • count - количество обновлений метрик; • failed - неуспешно; • success - успешно; • bucket - счетчик гистограммы (интервал), куда записывается значение; • le - границы значения, указывающие на то, насколько увеличится показатель гистограммы. • + inf - + бесконечность; • table - названия таблиц в БД. 	<p>Зафиксирована попытка отправки запроса в БД.</p>
---	------------------------------	-----------	---	---	---

8	mp_uploader_errors	counte r	<ul style="list-style-type: none"> • code • code_text 	<p>Счетчик ошибок по внутренним кодам ошибок с маппингом кодов на текстовые лейблы.</p> <ul style="list-style-type: none"> • code - код ошибки; • code_text - текст ошибки. 	<p>Появилась новая ошибка, обработчик которой существует в Медиаплатформе.</p>
---	---------------------------	-------------	---	---	--

9	mp_uploader_s3_response_time	histogram	<ul style="list-style-type: none"> • op_type: <ul style="list-style-type: none"> ○ chunk_create ○ chunk_upload ○ task_merge • status: <ul style="list-style-type: none"> ○ success ○ failed • le <ul style="list-style-type: none"> ○ 1 ○ 5 ○ 10 ○ 30 ○ 60 ○ 300 ○ +Inf • sum • count 	<p>Время выполнения операции создания задания на загрузку чанка, загрузки чанка и слияния чанков в S3.</p> <ul style="list-style-type: none"> • sum - сумма всех значений; • count - количество обновлений метрик; • failed - неуспешно; • success - успешно; • bucket - счетчик гистограммы (интервал), куда записывается значение; • le - граничащее значения, указывающие на то, насколько увеличится показатель гистограммы. • + inf - + бесконечность. 	<ul style="list-style-type: none"> • Создано задание на загрузку чанка. • Загружен новый чанк. • Произошло объединение чанков.
---	------------------------------	-----------	---	--	---

1 0	mp_uploader_video_status	counter	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ uploading_finished ○ uploading_started ○ uploading_failed 	<p>Счетчик увеличивается каждый раз при начале, завершении загрузки, ошибки слияния чанков.</p> <ul style="list-style-type: none"> • uploading_finished - закончено загрузок; • uploading_started - начато загрузок; • uploading_failed - загрузки, завершившиеся с ошибкой. 	<ul style="list-style-type: none"> • Создано задание на загрузку чанка. • Загружен новый чанк. • Произошло объединение чанков.
1 1	mp_uploader_uploaded_chunks	counter	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ success ○ failed 	<p>Накопленное количество загруженных чанков за время работы сервиса. По статусам:</p> <ul style="list-style-type: none"> • success - успешно; • failed - неуспешно. 	Загружен новый чанк.
1 2	mp_uploader_uploaded_files	counter	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ success ○ failed 	<p>Накопленное количество загруженных медиафайлов за время работы сервиса.</p> <ul style="list-style-type: none"> • success - успешно; • failed - неуспешно (в т.ч. незавершенные объединения чанков). 	Зафиксирована попытка объединения чанков в единый медиафайл.

1 3	mp_uploader_wrong_form_at_tasks	counte r		Количество запросов на загрузку, не прошедших валидацию.	Появился новый запрос, на который сервис не подписывался или который не удалось разобрать (получен не валидный json).
--------	--	-------------	--	--	---

3.1.3.4. Пример ответа модуля Uploader как экспортера Prometheus

```
# HELP Throughput_Mbps_HTTP_Upload
# TYPE Throughput_Mbps_HTTP_Upload summary
Throughput_Mbps_HTTP_Upload{quantile="0.9"} 41.943039999999996
Throughput_Mbps_HTTP_Upload_sum 109.23620799999999
Throughput_Mbps_HTTP_Upload_count 3
# HELP Throughput_Mbps_S3_Upload
# TYPE Throughput_Mbps_S3_Upload summary
Throughput_Mbps_S3_Upload{quantile="0.9"} 41.943039999999996
Throughput_Mbps_S3_Upload_sum 109.23620799999999
Throughput_Mbps_S3_Upload_count 3
# HELP mp_uploader_s3_bytes
# TYPE mp_uploader_s3_bytes counter
mp_uploader_s3_bytes{op_type: "read"} 1.3654526e+07
mp_uploader_s3_bytes{op_type: "write"} 1.3654526e+07
# HELP mp_uploader_connection_status metric value: 0 - disconnected, 1 - connected
# TYPE mp_uploader_connection_status gauge
mp_uploader_connection_status{service="kafka"} 1
mp_uploader_connection_status{service="postgres"} 1
mp_uploader_connection_status{service="s3"} 1
# HELP mp_uploader_http_time
# TYPE mp_uploader_http_time histogram
mp_uploader_http_time_bucket{METHOD="POST",code="201",path_pattern="/uploader/v1/upload_createtask",le="1"} 0
```


mp_uploader_http_time_bucket{METHOD="POST",code="201",path_pattern="/uploader/v1/upload_createtask",le="2.4839607111043738"} 0

mp_uploader_http_time_bucket{METHOD="POST",code="201",path_pattern="/uploader/v1/upload_createtask",le="6.170060814310146"} 0

mp_uploader_http_time_bucket{METHOD="POST",code="201",path_pattern="/uploader/v1/upload_createtask",le="15.326188647871062"} 0

mp_uploader_http_time_bucket{METHOD="POST",code="201",path_pattern="/uploader/v1/upload_createtask",le="38.06965045228558"} 0

mp_uploader_http_time_bucket{METHOD="POST",code="201",path_pattern="/uploader/v1/upload_createtask",le="94.56351600895424"} 0

mp_uploader_http_time_bucket{METHOD="POST",code="201",path_pattern="/uploader/v1/upload_createtask",le="234.89205847013176"} 0

mp_uploader_http_time_bucket{METHOD="POST",code="201",path_pattern="/uploader/v1/upload_createtask",le="583.4626445902387"} 0

mp_uploader_http_time_bucket{METHOD="POST",code="201",path_pattern="/uploader/v1/upload_createtask",le="1449.2982855592077"} 0

mp_uploader_http_time_bucket{METHOD="POST",code="201",path_pattern="/uploader/v1/upload_createtask",le="3599.999999999999"} 0

mp_uploader_http_time_bucket{METHOD="POST",code="201",path_pattern="/uploader/v1/upload_createtask",le="+Inf"} 1

mp_uploader_http_time_sum{METHOD="POST",code="201",path_pattern="/uploader/v1/upload_createtask"} 2.5606811e+07

mp_uploader_http_time_count{METHOD="POST",code="201",path_pattern="/uploader/v1/upload_createtask"} 1

mp_uploader_http_time_bucket{METHOD="PUT",code="200",path_pattern="/uploader/v1/upload_chunk",le="1"} 0

mp_uploader_http_time_bucket{METHOD="PUT",code="200",path_pattern="/uploader/v1/upload_chunk",le="2.4839607111043738"} 0

mp_uploader_http_time_bucket{METHOD="PUT",code="200",path_pattern="/uploader/v1/upload_chunk",le="6.170060814310146"} 0

mp_uploader_http_time_bucket{METHOD="PUT",code="200",path_pattern="/uploader/v1/upload_chunk",le="15.326188647871062"} 0

mp_uploader_http_time_bucket{METHOD="PUT",code="200",path_pattern="/uploader/v1/upload_chunk",le="38.06965045228558"} 0

mp_uploader_http_time_bucket{METHOD="PUT",code="200",path_pattern="/uploader/v1/upload_chunk",le="94.56351600895424"} 0

mp_uploader_http_time_bucket{METHOD="PUT",code="200",path_pattern="/uploader/v1/upload_chunk",le="234.89205847013176"} 0

mp_uploader_http_time_bucket{METHOD="PUT",code="200",path_pattern="/uploader/v1/upload_chunk",le="583.4626445902387"} 0

mp_uploader_http_time_bucket{METHOD="PUT",code="200",path_pattern="/uploader/v1/upload_chunk",le="1449.2982855592077"} 0

mp_uploader_http_time_bucket{METHOD="PUT",code="200",path_pattern="/uploader/v1/upload_chunk",le="3599.999999999999"} 0

mp_uploader_http_time_bucket{METHOD="PUT",code="200",path_pattern="/uploader/v1/upload_chunk",le="+Inf"} 2

mp_uploader_http_time_sum{METHOD="PUT",code="200",path_pattern="/uploader/v1/upload_chunk"} 3.37848099e+08

mp_uploader_http_time_count{METHOD="PUT",code="200",path_pattern="/uploader/v1/upload_chunk"} 2

mp_uploader_http_time_bucket{METHOD="PUT",code="211",path_pattern="/uploader/v1/upload_chunk",le="1"} 0

mp_uploader_http_time_bucket{METHOD="PUT",code="211",path_pattern="/uploader/v1/upload_chunk",le="2.4839607111043738"} 0

mp_uploader_http_time_bucket{METHOD="PUT",code="211",path_pattern="/uploader/v1/upload_chunk",le="6.170060814310146"} 0

mp_uploader_http_time_bucket{METHOD="PUT",code="211",path_pattern="/uploader/v1/upload_chunk",le="15.326188647871062"} 0

mp_uploader_http_time_bucket{METHOD="PUT",code="211",path_pattern="/uploader/v1/upload_chunk",le="38.06965045228558"} 0

mp_uploader_http_time_bucket{METHOD="PUT",code="211",path_pattern="/uploader/v1/upload_chunk",le="94.56351600895424"} 0

mp_uploader_http_time_bucket{METHOD="PUT",code="211",path_pattern="/uploader/v1/upload_chunk",le="234.89205847013176"} 0

```

mp_uploader_http_time_bucket{METHOD="PUT",code="211",path_pattern="/uploader/v1/upload
_chunk",le="583.4626445902387"} 0
mp_uploader_http_time_bucket{METHOD="PUT",code="211",path_pattern="/uploader/v1/upload
_chunk",le="1449.2982855592077"} 0
mp_uploader_http_time_bucket{METHOD="PUT",code="211",path_pattern="/uploader/v1/upload
_chunk",le="3599.999999999999"} 0
mp_uploader_http_time_bucket{METHOD="PUT",code="211",path_pattern="/uploader/v1/upload
_chunk",le="+Inf"} 1
mp_uploader_http_time_sum{METHOD="PUT",code="211",path_pattern="/uploader/v1/upload_c
hunk"} 1.31783234e+08
mp_uploader_http_time_count{METHOD="PUT",code="211",path_pattern="/uploader/v1/upload_
chunk"} 1
# HELP mp_uploader_kafka_messages_sent
# TYPE mp_uploader_kafka_messages_sent counter
mp_uploader_kafka_messages_sent{status="failure"} 0
mp_uploader_kafka_messages_sent{status="success"} 3
# HELP mp_uploader_pg_response_time
# TYPE mp_uploader_pg_response_time histogram
mp_uploader_pg_response_time_bucket{query_type="delete",status="success",table="chunks",le="
0.01"} 1
mp_uploader_pg_response_time_bucket{query_type="delete",status="success",table="chunks",le="
0.05"} 1
mp_uploader_pg_response_time_bucket{query_type="delete",status="success",table="chunks",le="
0.1"} 1
mp_uploader_pg_response_time_bucket{query_type="delete",status="success",table="chunks",le="
0.5"} 1
mp_uploader_pg_response_time_bucket{query_type="delete",status="success",table="chunks",le="
1"} 1
mp_uploader_pg_response_time_bucket{query_type="delete",status="success",table="chunks",le="
5"} 1
mp_uploader_pg_response_time_bucket{query_type="delete",status="success",table="chunks",le="
10"} 1

```

mp_uploader_pg_response_time_bucket{query_type="delete",status="success",table="chunks",le="+Inf"} 1
mp_uploader_pg_response_time_sum{query_type="delete",status="success",table="chunks"}
0.000302
mp_uploader_pg_response_time_count{query_type="delete",status="success",table="chunks"} 1
mp_uploader_pg_response_time_bucket{query_type="insert",status="success",table="chunks",le="0.01"} 3
mp_uploader_pg_response_time_bucket{query_type="insert",status="success",table="chunks",le="0.05"} 3
mp_uploader_pg_response_time_bucket{query_type="insert",status="success",table="chunks",le="0.1"} 3
mp_uploader_pg_response_time_bucket{query_type="insert",status="success",table="chunks",le="0.5"} 3
mp_uploader_pg_response_time_bucket{query_type="insert",status="success",table="chunks",le="1"} 3
mp_uploader_pg_response_time_bucket{query_type="insert",status="success",table="chunks",le="5"} 3
mp_uploader_pg_response_time_bucket{query_type="insert",status="success",table="chunks",le="10"} 3
mp_uploader_pg_response_time_bucket{query_type="insert",status="success",table="chunks",le="+Inf"} 3
mp_uploader_pg_response_time_sum{query_type="insert",status="success",table="chunks"}
0.001797
mp_uploader_pg_response_time_count{query_type="insert",status="success",table="chunks"} 3
mp_uploader_pg_response_time_bucket{query_type="insert",status="success",table="tasks",le="0.01"} 1
mp_uploader_pg_response_time_bucket{query_type="insert",status="success",table="tasks",le="0.05"} 1
mp_uploader_pg_response_time_bucket{query_type="insert",status="success",table="tasks",le="0.1"} 1
mp_uploader_pg_response_time_bucket{query_type="insert",status="success",table="tasks",le="0.5"} 1

```
mp_uploader_pg_response_time_bucket{query_type="insert",status="success",table="tasks",le="1"
} 1
mp_uploader_pg_response_time_bucket{query_type="insert",status="success",table="tasks",le="5"
} 1
mp_uploader_pg_response_time_bucket{query_type="insert",status="success",table="tasks",le="10
"} 1
mp_uploader_pg_response_time_bucket{query_type="insert",status="success",table="tasks",le="+I
nf"} 1
mp_uploader_pg_response_time_sum{query_type="insert",status="success",table="tasks"}
0.001597
mp_uploader_pg_response_time_count{query_type="insert",status="success",table="tasks"} 1
mp_uploader_pg_response_time_bucket{query_type="select",status="success",table="tasks",le="0.
01"} 19
mp_uploader_pg_response_time_bucket{query_type="select",status="success",table="tasks",le="0.
05"} 19
mp_uploader_pg_response_time_bucket{query_type="select",status="success",table="tasks",le="0.
1"} 19
mp_uploader_pg_response_time_bucket{query_type="select",status="success",table="tasks",le="0.
5"} 19
mp_uploader_pg_response_time_bucket{query_type="select",status="success",table="tasks",le="1"
} 19
mp_uploader_pg_response_time_bucket{query_type="select",status="success",table="tasks",le="5"
} 19
mp_uploader_pg_response_time_bucket{query_type="select",status="success",table="tasks",le="10
"} 19
mp_uploader_pg_response_time_bucket{query_type="select",status="success",table="tasks",le="+I
nf"} 19
mp_uploader_pg_response_time_sum{query_type="select",status="success",table="tasks"}
0.0070779999999999999
mp_uploader_pg_response_time_count{query_type="select",status="success",table="tasks"} 19
mp_uploader_pg_response_time_bucket{query_type="update",status="success",table="tasks",le="0
.01"} 11
```

```

mp_uploader_pg_response_time_bucket{query_type="update",status="success",table="tasks",le="0
.05"} 11
mp_uploader_pg_response_time_bucket{query_type="update",status="success",table="tasks",le="0
.1"} 11
mp_uploader_pg_response_time_bucket{query_type="update",status="success",table="tasks",le="0
.5"} 11
mp_uploader_pg_response_time_bucket{query_type="update",status="success",table="tasks",le="1
"} 11
mp_uploader_pg_response_time_bucket{query_type="update",status="success",table="tasks",le="5
"} 11
mp_uploader_pg_response_time_bucket{query_type="update",status="success",table="tasks",le="1
0"} 11
mp_uploader_pg_response_time_bucket{query_type="update",status="success",table="tasks",le="+
Inf"} 11
mp_uploader_pg_response_time_sum{query_type="update",status="success",table="tasks"}
0.005382999999999999
mp_uploader_pg_response_time_count{query_type="update",status="success",table="tasks"} 11
# HELP mp_uploader_errors # TYPE mp_uploader_errors counter
mp_uploader_errors{code="1010404",code_text="error_sending_message_to_kafka"} 1
# HELP mp_uploader_s3_response_time
# TYPE mp_uploader_s3_response_time histogram
mp_uploader_s3_response_time_bucket{op_type="chunk_create",status="success",le="1"} 1
mp_uploader_s3_response_time_bucket{op_type="chunk_create",status="success",le="5"} 1
mp_uploader_s3_response_time_bucket{op_type="chunk_create",status="success",le="10"} 1
mp_uploader_s3_response_time_bucket{op_type="chunk_create",status="success",le="30"} 1
mp_uploader_s3_response_time_bucket{op_type="chunk_create",status="success",le="60"} 1
mp_uploader_s3_response_time_bucket{op_type="chunk_create",status="success",le="300"} 1
mp_uploader_s3_response_time_bucket{op_type="chunk_create",status="success",le="+Inf"} 1
mp_uploader_s3_response_time_sum{op_type="chunk_create",status="success"} 0.004652
mp_uploader_s3_response_time_count{op_type="chunk_create",status="success"} 1
mp_uploader_s3_response_time_bucket{op_type="chunk_upload",status="success",le="1"} 3
mp_uploader_s3_response_time_bucket{op_type="chunk_upload",status="success",le="5"} 3

```

```

mp_uploader_s3_response_time_bucket{op_type="chunk_upload",status="success",le="10"} 3
mp_uploader_s3_response_time_bucket{op_type="chunk_upload",status="success",le="30"} 3
mp_uploader_s3_response_time_bucket{op_type="chunk_upload",status="success",le="60"} 3
mp_uploader_s3_response_time_bucket{op_type="chunk_upload",status="success",le="300"} 3
mp_uploader_s3_response_time_bucket{op_type="chunk_upload",status="success",le="+Inf"} 3
mp_uploader_s3_response_time_sum{op_type="chunk_upload",status="success"}
0.40619000000000005
mp_uploader_s3_response_time_count{op_type="chunk_upload",status="success"} 3
mp_uploader_s3_response_time_bucket{op_type="task_merge",status="success",le="1"} 1
mp_uploader_s3_response_time_bucket{op_type="task_merge",status="success",le="5"} 1
mp_uploader_s3_response_time_bucket{op_type="task_merge",status="success",le="10"} 1
mp_uploader_s3_response_time_bucket{op_type="task_merge",status="success",le="30"} 1
mp_uploader_s3_response_time_bucket{op_type="task_merge",status="success",le="60"} 1
mp_uploader_s3_response_time_bucket{op_type="task_merge",status="success",le="300"} 1
mp_uploader_s3_response_time_bucket{op_type="task_merge",status="success",le="+Inf"} 1
mp_uploader_s3_response_time_sum{op_type="task_merge",status="success"} 0.022899
mp_uploader_s3_response_time_count{op_type="task_merge",status="success"} 1
# HELP mp_uploader_video_status
# TYPE mp_uploader_video_status counter
mp_uploader_video_status{status="failed"} 0
mp_uploader_video_status{status="finished"} 1
mp_uploader_video_status{status="started"} 1
# HELP uploaded_chunks
# TYPE uploaded_chunks counter
uploaded_chunks{status="FAILED"} 0
uploaded_chunks{status="OK"} 3
# HELP uploaded_files
# TYPE uploaded_files counter
uploaded_files{status="FAILED"} 0
uploaded_files{status="OK"} 1
# HELP wrong_format_tasks
# TYPE wrong_format_tasks counter

```

3.2. Prober

3.2.1. Назначение

Сервис является первым звеном на пути обработки файла загруженного пользователем в хранилище. Основная задача сервиса - имея доступ к контенту, предоставить информацию о медиа данных необходимую для

- планирования (без доступа к данным) дальнейшей обработки контента
- сбора статистики

Так же, при наличии ресурсов, сервис способен проверить целостность загруженных данных.

3.2.2. Переменные окружения Prober

В таблице ниже представлены переменные окружения, которые используются для настройки компонента Prober.

Переменная	Тип	Формат	Диапазон возможных значений	Обязательно	Описание
Блок базовых настроек					
POD_NAME	String			Нет	Идентификатор пода Kubernetes, уникальное значение. Значение по умолчанию: отсутствует.
BASE_PATH	String	/file_path		Нет	Базовый путь для входных и выходных файлов. Применяется для относительных путей в локальной файловой системе во входящих и исходящих сообщениях. Значение по умолчанию: отсутствует.

CONNECTOR_TYPE	String		KAFKA	Да	Брокер-сообщение, от которого Task Manager получает задачи. Используется значение KAFKA. Значение по умолчанию: KAFKA.
LOG_LEVEL	String		<ol style="list-style-type: none"> 1. none 2. fatal 3. critical 4. error 5. warning 6. notice 7. information 8. debug 9. trace 	Нет	<p>Уровень ведения журнала событий. Применяется как значение по умолчанию для всех служб, если система ведения журнала не сконфигурирована, или через командную строку компонента Transcoder.</p> <p>Допустимые значения:</p> <ul style="list-style-type: none"> • none – ведение журнала отключено; • fatal – неустраняемая ошибка; • critical – критическая ошибка; • error – ошибка; • warning – предупреждение; • notice – уведомление; • information – информация; • debug – отладка; • trace – трассировка. <p>Значение по умолчанию: information.</p>

AWS_LOG_LEVEL	String		<ol style="list-style-type: none"> 1. none 2. fatal 3. critical 4. error 5. warning 6. notice 7. information 8. debug 9. trace 	Нет	<p>Уровень логирования взаимодействия с облачным хранилищем S3.</p> <p>Допустимые значения:</p> <ul style="list-style-type: none"> • none – ведение журнала отключено; • fatal – неустранимая ошибка; • critical – критическая ошибка; • error – ошибка; • warning – предупреждение; • notice – уведомление; • information – информация; • debug – отладка; • trace – трассировка. <p>Значение по умолчанию: error.</p>
PERSISTENT_CONNECTOR	Integer		<ol style="list-style-type: none"> 1. 1 2. 2 	Нет	<p>Параметр, который указывает на доступность соединения хотя бы с одним брокером Kafka.</p>
Блок настроек параметров проверки: применяется при активированном режиме углубленной проверки файлов через демультимплексирование и декодирование					
PROBING_CHECK_PERIOD_SEC	Integer			Нет	<p>Период эвристической оценки скорости/продолжительности обработки файла, в секундах.</p> <p>Значение по умолчанию: 10.</p>

PROBING_DURATION_LIMIT_SEC	Integer			Нет	Максимально допустимая длительность обработки файла, в секундах. Если Prober оценивает, что файл будет обрабатываться дольше этого времени - задание завершается с ошибкой. Значение по умолчанию: 3600.
PROBING_RATE_LIMIT_BYTES_PER_SEC	Integer			Нет	Минимально допустимая скорость обработки файла, в байтах в секунду. Если Prober оценивает, что файл обрабатывается с меньшей скоростью - задание завершается с ошибкой. Значение по умолчанию: 1024.
Блок переменных, необходимых, если CONNECTOR_TYPE=KAFKA					
KAFKA_BROKERS	String	host или host:port, значения разделяются запятыми		Да, если CONNECTOR_TYPE=KAFKA	Список брокеров Kafka в виде значений, разделённых запятыми, в формате <хост> или <хост>:<порт>. Значение по умолчанию: отсутствует.
KAFKA_GROUP_ID	String			Нет	Идентификатор группы Kafka (consumer group.id). Значение по умолчанию: 0.
KAFKA_TOPIC_PROBE	String			Да, если CONNECTOR_TYPE=KAFKA	Имя топика Kafka, куда Task Manager ставит задачи на анализ загруженного медиафайла для компонента Prober. Значение по умолчанию: KAFKA_TOPIC_PROBE.

KAFKA_TOPIC_TASK_MANAGER	String			Да, если CONNECTOR_TYPE=KAFKA	Имя топика Kafka, куда Prober отправляет сообщения о результатах анализа загруженного медиафайла: <ul style="list-style-type: none"> • завершение анализа медиафайла; • сбой при анализе медиафайла. Сообщения предназначены для считывания компонентом Task Manager. Значение по умолчанию: KAFKA_TOPIC_TASK_MANAGER.
KAFKA_SECURITY_PROTOCOL	String			Нет	Тип проверки подлинности и шифрования Kafka. Соответствует конфигурационному параметру Kafka: security.protocol.
KAFKA_SASL_MECHANISMS	String			Да, если задано KAFKA_SECURITY_PROTOCOL	Дополнительный параметр для проверки подлинности и шифрования Kafka. Соответствует конфигурационному параметру Kafka: sasl.mechanisms.
KAFKA_SASL_USERNAME	String			Да, если задано KAFKA_SECURITY_PROTOCOL	Дополнительный параметр для проверки подлинности и шифрования Kafka. Соответствует конфигурационному параметру Kafka: sasl.username.
KAFKA_SASL_PASSWORD	String			Да, если задано KAFKA_SECURITY_PROTOCOL	Дополнительный параметр для проверки подлинности и шифрования Kafka. Соответствует конфигурационному параметру Kafka: sasl.password.
Блок настроек телеметрии					

Необязательные переменные — если не указано, сбор телеметрии осуществляться не будет.					
TELEMETRY_TRACE S_EXPORTER_TYPE	String		<ol style="list-style-type: none"> 1. C O N S O L E 2. N U L L _ S I N K 3. F I L E 	Нет	<p>Тип экспорта телеметрии. Возможные значения:</p> <ul style="list-style-type: none"> • CONSOLE - данные выводятся в консоль; • NULL_SINK - телеметрия отключена; • FILE - данные записываются в локальный файл. <p>Значение по умолчанию: NULL_SINK.</p>
TELEMETRY_TRACE S_URI	String	<p>Возможные форматы:</p> <ul style="list-style-type: none"> • /file_path • имя файла 		Зависит от TELEMETRY_TRACE_S_EXPORTER_TYPE	<p>Адрес для экспорта телеметрии:</p> <ul style="list-style-type: none"> • для FILE - имя файла; <p>Для CONSOLE и NULL_SINK эта переменная необязательна, для FILE - обязательна.</p> <p>Значение по умолчанию: отсутствует.</p>
TELEMETRY_METRICS_ADDRESS	String	host:port		Нет	<p>Адрес локального веб-сервера, по которому Prometheus сможет собрать метрики с конкретного экземпляра транскодера (Pod), формат — <хост>:<порт>.</p> <p>Значение по умолчанию: отсутствует.</p>
Блок настроек подключения к хранилищу медиаданных					

S3_KEY_ID	String			Да, если предполагается доступ к хранилищу S3	Логин для подключения к хранилищу S3. Значение по умолчанию: отсутствует.
S3_SECRET_KEY	String			Да, если предполагается доступ к хранилищу S3	Пароль для подключения к хранилищу S3. Значение по умолчанию: Отсутствует.

Переменные среды указываются в файле или файлах YAML. Это может быть как единый файл для всех компонентов Медиаплатформы, так и несколько файлов.

В зависимости от вариантов развёртывания переменные можно указать вручную или включить в настройки из других источников.

Ниже представлен пример настроек компонента Prober.

env:

Блок базовых настроек

- name: BASE_PATH
value: "/home/user/media"
- name: POD_NAME
value: "prober"
- name: LOG_LEVEL
value: "debug"
- name: AWS_LOG_LEVEL
value: "debug"
- name: PERSISTENT_CONNECTOR
value: 1

Настройки брокера сообщений

- name: CONNECTOR_TYPE
value: "KAFKA"
- name: KAFKA_BROKERS
value: 127.0.0.1:9093
- name: KAFKA_TOPIC_PROBE
value: KAFKA_TOPIC_PROBE
- name: KAFKA_TOPIC_TASK_MANAGER

value: KAFKA_TOPIC_TASK_MANAGER
- name: KAFKA_GROUP_ID
value: 0
- name: KAFKA_SECURITY_PROTOCOL
value: kafka_security_protocol
- name: KAFKA_SASL_MECHANISMS
value: kafka_sasl_mechanisms
- name: KAFKA_SASL_USERNAME
value: kafka_sasl_username
- name: KAFKA_SASL_PASSWORD
value: kafka_sasl_password

Настройка параметров проверки

- name: PROBING_CHECK_PERIOD_SEC
value: "10"
- name: PROBING_DURATION_LIMIT_SEC
value: "3600"
- name: PROBING_RATE_LIMIT_BYTES_PER_SEC
value: "1024"

Настройки подключения к хранилищу S3

- name: S3_ID_KEY
value: s3_id_key
- name: S3_SECRET_KEY
value: s3_secret_key

Настройка телеметрии

- name: TELEMETRY_METRICS_ADDRESS
value: "0.0.0.0:8080"
- name: TELEMETRY_TRACES_EXPORTER_TYPE
value: "CONSOLE"
- name: TELEMETRY_TRACES_URI

```
value: "CONSOLE"
- name: LD_LIBRARY_PATH
value: "/platform_libs"
```

3.2.3. Сбор метрик

Отправителем метрик является сервис Prober, входящий в состав Медиаплатформы.

Метрики собираются с помощью системы мониторинга и оповещений Prometheus и отображаются в виде графиков, счетчиков и в других представлениях с помощью инструмента для визуализации и анализа данных Grafana.

Отображаемый отрезок времени на всех графиках по умолчанию: 6 часов.

Все метрики выводятся для списка подов или конкретного пода.

3.2.3.1. Описание функциональных клавиш Grafana

- **namespace** – выбор окружения;
- **prober_instance** – поды Prober;
- **transcoder_instance** – поды Transcoder;
- **uploader_instance** – поды Uploader;
- **taskmanager_instance** – поды Task Manager.

Для всех функциональных клавиш, кроме **namespace**, значение по умолчанию all.

3.2.3.2. Описание метрик Grafana

При выборе большого временного интервала или частом перезапуске подов на графиках и счётчиках относительно него могут отображаться все запущенные поды и их состояние. Для мониторинга последних следует выбирать более короткий интервал. Данное поведение позволяет отслеживать состояние конкретного пода в конкретный момент времени.

Разрывы на графиках и отсутствие информации возможны в случае, если после старта сервиса метрика ни разу не отсылалась или сервис не был запущен или не работает в следствии системной ошибки.

Метрики, которые регистрируют единичные всплески, могут быть не видны на графиках, где для его построения используется функции rate или increase, которые возвращают скорость изменения метрики на заданном интервале, при этом первая делит полученное значение на этот интервал. Для регистрации и отслеживания подобных изменений следует использовать счётчики.

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
1	Overall	Prober - количество ошибок по внутренним кодам (количество/с)	График	Общее количество ошибок по внутренним кодам ошибок.	<ul style="list-style-type: none"> code 	<p>Метрика: mp_prober_errors</p> <ul style="list-style-type: none"> Total: <ul style="list-style-type: none"> sum(sum(rate(mp_prober_errors{code!="0", namespace=~"\$namespace", pod=~"\$prober_instance"}[\$__rate_interval])) by (code)) {{code}}: <ul style="list-style-type: none"> sum(rate(mp_prober_errors{code!="0", namespace=~"\$namespace", pod=~"\$prober_instance"}[\$__rate_interval])) by (code)
2	Overall	Prober - общее количество ошибок по текстовым кодам после старта	Счетчик / Таблица	Общее количество ошибок по внутренним кодам ошибок с маппингом кодов на текстовые лейблы с момента последней перезагрузки.	<ul style="list-style-type: none"> code text_code 	<p>Метрика: mp_prober_errors</p> <ul style="list-style-type: none"> {{text_code}}: <ul style="list-style-type: none"> sum(mp_prober_errors{code!="0", pod=~"\$prober_instance"}) by (code, text_code)

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
3	Overall	Prober - статус подключения к сервисам	График	<p>Статус связи с сервисами:</p> <ul style="list-style-type: none"> • 1 - соединены • 0 - соединены отсутствуют. <p>Все графики суммируются, каждый следующий будет выглядеть, как своё значение плюс предыдущее.</p>	<ul style="list-style-type: none"> • exported_service: <ul style="list-style-type: none"> ○ 	<p>Метрика: mp_prober_connection_status</p> <ul style="list-style-type: none"> • {{exported_service}} - {{pod}}: mp_prober_connection_status{exported_service="kafka", pod=~"\$prober_instance"}

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
4	Overall	Prober - Kafka - статус подключения	Измеритель	Общее количество сервисов и количество сервисов, подключённых к Kafka в данный момент.	<ul style="list-style-type: none"> • exported_service: <ul style="list-style-type: none"> ○ 	<p>Метрика: mp_prober_connection_status</p> <ul style="list-style-type: none"> • количество сервисов: count(mp_prober_connection_status{exported_service="kafka", namespace=~"\$namespace", pod=~"\$prober_instance"}) • Подключённых сервисов: sum(mp_prober_connection_status{exported_service="kafka", namespace=~"\$namespace", pod=~"\$prober_instance"})
5	Prober	Количество ошибок (количество/с)	График	Группы ошибок по загрузке и обработке.	<ul style="list-style-type: none"> • code_aggregated: <ul style="list-style-type: none"> ○ ○ 	<p>Метрика: mp_prober_aggregated_error</p> <ul style="list-style-type: none"> • io: sum(rate(mp_prober_aggregated_error{code_aggregated="io", namespace=~"\$namespace", pod=~"\$prober_instance"}[\$__rate_interval])) >= 0 or on() vector(0) • processing: sum(rate(mp_prober_aggregated_error{code_aggregated="processing", namespace=~"\$namespace", pod=~"\$prober_instance"}[\$__rate_interval])) >= 0 or on() vector(0)

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
6	Prober	Статус обработки заданий	Изменение	Текущее состояние сервиса: обработка задания.	<ul style="list-style-type: none"> • value: <ul style="list-style-type: none"> ○ ○ 	<p>Метрика: mp_prober_state</p> <ul style="list-style-type: none"> • Processing: <pre>sum(mp_prober_state{ value="processing", namespace=~"\$namespace", pod=~"\$prober_instance" } >= 0 or on() vector(0))</pre> • Waiting: <pre>sum(mp_prober_state{ value="waiting", namespace=~"\$namespace", pod=~"\$prober_instance" } >= 0 or on() vector(0))</pre>
7	Prober	Количество событий ребалансировки (количество/с)	График	Средняя экстраполированная скорость изменения величины количества событий ребалансировки подписки на входной топик Kafka.	<ul style="list-style-type: none"> • pod • job 	<p>Метрика: mp_prober_rebalance</p> <ul style="list-style-type: none"> • {{pod}}: <pre>sum(rate(mp_prober_rebalance{ namespace=~"\$namespace", pod=~"\$prober_instance" }[\$__rate_interval])) by (pod, job)</pre>

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
8	Prober	Ошибки ребалансировки (количество/с)	График	Количество внештатных ситуаций в работе модуля Prober, связанное с ребалансировкой.	<ul style="list-style-type: none"> • pod • job • topic: <ul style="list-style-type: none"> ○ 	Метрика: mp_prober_rebalance_assigned_partitions <ul style="list-style-type: none"> • {{pod}}: <pre>sum(rate(mp_prober_rebalance_assigned_partitions{topic="prober.tasks", namespace=~"\$namespace", pod=~"\$prober_instance"}[\$__rate_interval])) by (pod, job)</pre>
9	Prober	Ошибки ребалансировки после старта	Счетчик	Количество внештатных ситуаций в работе модуля Prober, связанное с ребалансировкой с момента последней перезагрузки.	<ul style="list-style-type: none"> • topic: <ul style="list-style-type: none"> ○ 	Метрика: mp_prober_rebalance_assigned_partitions <ul style="list-style-type: none"> • {{pod}}: <pre>sum(mp_prober_rebalance_assigned_partitions{topic="prober.tasks", namespace=~"\$namespace", pod=~"\$prober_instance"})</pre>

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
10	Prober	Количество заданий за секунду (количество/с)	График	Средняя экстраполированная скорость изменения кол-ва заданий за определенный период. На графике представлено общее количество успешных заданий, не успешных заданий по всем подам, а также отдельно для каждого из них.	<ul style="list-style-type: none"> • pod • status: <ul style="list-style-type: none"> ○ 	Метрика: mp_prober_tasks <ul style="list-style-type: none"> • Total - Success: <pre>sum(rate(mp_prober_tasks { status="success", namespace=~"\$namespace", pod=~"\$prober_instance" } [\$_rate_interval]))</pre> • Total - Failed: <pre>sum(rate(mp_prober_tasks { status="failed", namespace=~"\$namespace", pod=~"\$prober_instance" } [\$_rate_interval]))</pre> • Success - {{pod}}: <pre>sum(rate(mp_prober_tasks { status="success", namespace=~"\$namespace", pod=~"\$prober_instance" } [\$_rate_interval])) by (pod)</pre> • Failed - {{pod}}: <pre>sum(rate(mp_prober_tasks { status="failed", namespace=~"\$namespace", pod=~"\$prober_instance" } [\$_rate_interval])) by (pod)</pre>
11	Prober	Общее количество заданий после старта	Счетчик	Общее количество заданий, количество успешных и не успешных заданий за время работы сервиса с момента последней перезагрузки.	<ul style="list-style-type: none"> • pod • status: <ul style="list-style-type: none"> ○ 	Метрика: mp_prober_tasks <ul style="list-style-type: none"> • Total: <pre>sum(mp_prober_tasks { namespace=~"\$namespace", pod=~"\$prober_instance" })</pre> • Success: <pre>sum(mp_prober_tasks { status="success", namespace=~"\$namespace", pod=~"\$prober_instance" })</pre> • Failed: <pre>sum(mp_prober_tasks { status="failed", namespace=~"\$namespace", pod=~"\$prober_instance" })</pre>

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
1 2	Prober	Средняя пропускная способность (мгб/с), 0.9 квантиль	График	Средняя пропускная способность модуля Prober по заданному квантилю, мегабит в секунду. Имеется в виду обобщенная скорость обработки видео контента, рассчитанная путем деления размера входного видео файла на полное время обработки задания.	<ul style="list-style-type: none"> • quantile: <ul style="list-style-type: none"> ○ 	<p>Метрика: mp_prober_throughput_mbps</p> <ul style="list-style-type: none"> • {{pod}}: mp_prober_throughput_mbps{quantile="0.9", namespace=~"\$namespace", pod=~"\$prober_instance"}
1 3	Prober	Среднее время обработки заданий (с), 0.9 квантиль	График	Среднее время обработки задания по заданному квантилю, в секундах.	<ul style="list-style-type: none"> • quantile: <ul style="list-style-type: none"> ○ 	<p>Метрика:</p> <p>mp_prober_task_processing_time_seconds</p> <ul style="list-style-type: none"> • {{pod}}: mp_prober_task_processing_time_seconds{quantile="0.9", namespace=~"\$namespace", pod=~"\$prober_instance"}

Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
14	Prober	График	Пропускная способность чтения из S3 по квантилям (мгб/с)	<ul style="list-style-type: none"> quantile: <ul style="list-style-type: none"> o o op_type: <ul style="list-style-type: none"> o 	Метрика: mp_prober_throughput_mbps_s3 <ul style="list-style-type: none"> {{pod}} - quantile - 0.9: mp_prober_throughput_mbps_s3{quantile="0.9", op_type="read", namespace=~"\$namespace", pod=~"\$prober_instance"} {{pod}} - quantile - 0.99: mp_prober_throughput_mbps_s3{quantile="0.99", op_type="read", namespace=~"\$namespace", pod=~"\$prober_instance"}
15	Prober	График	Время, потраченное на ожидание задания по заданному квантилю.	<ul style="list-style-type: none"> quantile: <ul style="list-style-type: none"> o 	Метрика: mp_prober_task_waiting_time_seconds <ul style="list-style-type: none"> {{pod}}: mp_prober_task_waiting_time_seconds{quantile="0.9", namespace=~"\$namespace", pod=~"\$prober_instance"}
16	Prober	График	Время чтения файлов из S3 в секундах.	<ul style="list-style-type: none"> op_type: <ul style="list-style-type: none"> o 	Метрика: mp_prober_s3_seconds <ul style="list-style-type: none"> {{pod}} - read: mp_prober_s3_seconds{op_type="read", namespace=~"\$namespace", pod=~"\$prober_instance"}
17	Prober	График	Объем чтения файлов из S3 в мегабайтах в секунду.	<ul style="list-style-type: none"> pod op_type: <ul style="list-style-type: none"> o 	Метрика: mp_prober_s3_bytes <ul style="list-style-type: none"> {{pod}} - read: sum(rate(mp_prober_s3_bytes{op_type="read", namespace=~"\$namespace", pod=~"\$prober_instance"}[\$__rate_interval])) by (pod)

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
18	Prober	Общий объем чтения файлов из S3, с момента перезапуска модуля (мгб)	Счетчик	Общий объем чтения файлов из S3 из в мегабайтах с момента перезапуска модуля.	<ul style="list-style-type: none"> • op_type: <ul style="list-style-type: none"> ○ 	Метрика: mp_prober_s3_bytes <ul style="list-style-type: none"> • Total: <pre>(sum(mp_prober_s3_bytes{op_type="read", namespace=~"\$namespace", pod=~"\$prober_instance"}) >=0 or on() vector(0))/1e+6</pre>

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
19	Prober	Сообщения в Kafka (количество/с)	График	Количество сообщений, полученных, необработанных и отправленных в Kafka.	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ • type: <ul style="list-style-type: none"> ○ 	<p>Метрика: mp_prober_kafka_writed_messages, mp_prober_unhandled_messages, mp_prober_kafka_read_messages</p> <ul style="list-style-type: none"> • Send - success: sum(rate(mp_prober_kafka_writed_messages{status="success", namespace=~"\$namespace", pod=~"\$prober_instance"}[\$__rate_interval])) >=0 or on() vector(0) • Send - failure: sum(rate(mp_prober_kafka_writed_messages{status="failed", namespace=~"\$namespace", pod=~"\$prober_instance"}[\$__rate_interval])) >=0 or on() vector(0) • Unhandled - unexpected: sum(rate(mp_prober_unhandled_messages{type="unexpected", namespace=~"\$namespace", pod=~"\$prober_instance"}[\$__rate_interval])) >=0 or on() vector(0) • Unhandled - unknown: sum(rate(mp_prober_unhandled_messages{type="unknown", namespace=~"\$namespace", pod=~"\$prober_instance"}[\$__rate_interval])) >=0 or on() vector(0) • Read - success: sum(rate(mp_prober_kafka_read_messages{status="success", namespace=~"\$namespace", pod=~"\$prober_instance"}[\$__rate_interval])) >=0 or on() vector(0) • Read - failure: sum(rate(mp_prober_kafka_read_messages{status="failed", namespace=~"\$namespace", pod=~"\$prober_instance"}[\$__rate_interval])) >=0 or on() vector(0)

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
20	Prober	Общее количество сообщений Kafka после старта	Счетчик	Общее количество сообщений, полученных, необработанных и отправленных в Kafka с момента перезапуска модуля.	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ • type: <ul style="list-style-type: none"> ○ 	<p>Метрика: mp_prober_kafka_writed_messages, mp_prober_unhandled_messages, mp_prober_kafka_read_messages</p> <ul style="list-style-type: none"> • Send - success: sum(mp_prober_kafka_writed_messages{status="success", namespace=~"\$namespace", pod=~"\$prober_instance"}) >=0 or on() vector(0) • Send - failure: sum(mp_prober_kafka_writed_messages{status="failed", namespace=~"\$namespace", pod=~"\$prober_instance"}) >=0 or on() vector(0) • Unhandled - unexpected: sum(mp_prober_unhandled_messages{type="unexpected", namespace=~"\$namespace", pod=~"\$prober_instance"}) >=0 or on() vector(0) • Unhandled - unknown: sum(mp_prober_unhandled_messages{type="unknown", namespace=~"\$namespace", pod=~"\$prober_instance"}) >=0 or on() vector(0) • Read - success: sum(mp_prober_kafka_read_messages{status="success", namespace=~"\$namespace", pod=~"\$prober_instance"}) >=0 or on() vector(0) • Read - failure: sum(mp_prober_kafka_read_messages{status="failed", namespace=~"\$namespace", pod=~"\$prober_instance"}) >=0 or on() vector(0)

Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
21	Prober	Количество назначенных партиций в Kafka	Счетчик	Количество назначенных партиций в Kafka	<ul style="list-style-type: none"> topic: <ul style="list-style-type: none"> ○ <p>Метрика: mp_prober_assigned_partitions</p> <ul style="list-style-type: none"> Assigned partitions: sum(mp_prober_assigned_partitions{topic="prober.tasks", namespace=~"\$namespace", pod=~"\$prober_instance"})

3.2.3.3. Описание метрик Prometheus

- gauge - масштабируемый диалоговый индикатор, который отображает текущее значение метрики в процентах или числах. Большим преимуществом Gauge в Grafana является то, что он может легко приспосабливаться к разным значениям, чтобы обеспечить оптимальное отображение информации на дашборде: может быть использован для отображения любых числовых данных, таких как счётчики, процентные доли, абсолютные значения и т.д., т.к. gauge измеряет числовое значение в определенный момент времени, а не считает скорость изменения значений, как Counter. Gauge в Grafana можно отображать не только в виде графика, но и в виде других типов визуализации, таких как текст, таблицы или специально созданные плагины.
- counter - счетчик, который увеличивается на каждое новое событие. Счетчики обычно могут только увеличиваться, но не уменьшаться, и сброс в ноль может потребоваться вручную.
- summary - собирает информацию о распределении значений заданной величины. Это позволяет собирать статистику о количестве значений, их среднем, медиане, процентилях и т.д. Для каждого Summary Prometheus собирает образцы (samples) для значений, которые необходимо агрегировать. Для каждого образца указывается значение и метка времени, когда оно было собрано. Затем Prometheus вычисляет ряд

статистических значений на основе этих образцов, включая сумму всех значений, количество собранных образцов и среднее значение. В Grafana представление Summary может использоваться для создания графиков и диаграмм, показывающих распределение значений величины. Графики Summary могут быть полезны для анализа производительности приложения, когда необходимо определить, какое количество запросов было выполнено за определенный период времени, и как быстро были выполнены запросы.

Ниже приведена таблица с описанием метрик Prometheus для сервиса Prober.

	Наименование метрики	Тип	Метки	Описание	Событие
1	mp_prober_tasks	counter	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ failed ○ success 	<p>Количество выполненных заданий за период по типам:</p> <ul style="list-style-type: none"> • failed - неуспешно; • success - успешно. 	<ul style="list-style-type: none"> • Задание на анализ медиафайла успешно выполнено. • Задание на анализ медиафайла выполнено с ошибкой.
2	mp_prober_s3_seconds	counter	<ul style="list-style-type: none"> • op_type: <ul style="list-style-type: none"> ○ read 	<p>Общее время чтения файлов из S3 в секундах.</p> <ul style="list-style-type: none"> • read - чтение файла; 	<p>Зафиксирована попытка чтения файла из хранилища S3.</p>
3	mp_prober_s3_bytes	counter	<ul style="list-style-type: none"> • op_type: <ul style="list-style-type: none"> ○ read 	<p>Объем чтения файлов в S3 в байтах.</p> <ul style="list-style-type: none"> • read - чтение файла; 	<p>Зафиксирована попытка чтения файла из хранилища S3.</p>

4	mp_prober_aggregated_error	counter	<ul style="list-style-type: none"> • code_aggregated: <ul style="list-style-type: none"> ○ processing ○ io 	Метрика по группам ошибок. <ul style="list-style-type: none"> • processing - обработка; • io - загрузка. 	Появилась новая ошибка, обработчик которой существует в Медиaplatforme.
5	mp_prober_kafka_writed_messages	counter	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ failed ○ success 	Общее количество отправленных сообщений в Kafka. <ul style="list-style-type: none"> • failed - неуспешно; • success - успешно. 	Зафиксирована попытка отправки сообщения в Kafka.
6	mp_prober_unhandled_messages	counter	<ul style="list-style-type: none"> • type: <ul style="list-style-type: none"> ○ unknown ○ unexpected 	Общее количество необработанных сообщений. <ul style="list-style-type: none"> • unknown - сервис не смог идентифицировать сообщение; • unexpected - сервис принял сообщение, которое ему не предназначено. 	Появился новый запрос, на который модуль не подписывался или который не удалось разобрать (получен не валидный json).

7	mp_prober_rebalance	counte r		Количество событий ре-балансировки подписки на входной топик Kafka за период.	<p>Зафиксировано событие ребалансировки подов: если один из подов модуля Prober перестает работать, Kafka замечает, что потребитель для некоторых партиций топика не отвечает, и начинает ребалансировку, перераспределяя партиции между другими подами модуля Prober. В этот момент все работающие поды модуля получают сигнал о начале ребалансировки. Количество таких событий и отражает данную метрику.</p> <p>Возможные причины роста метрики:</p> <ol style="list-style-type: none"> 1. Аппаратные проблемы. Необходимо проверить железные метрики CPU, GPU, RAM. 2. Проблемы со входным и файлами
---	----------------------------	-------------	--	---	---

					Необходимо проверить логи Kibana.
8	mp_prober_kafka_read_messages	counter	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ failed ○ success 	<p>Общее количество входящих сообщений.</p> <ul style="list-style-type: none"> • failed - неуспешно; • success - успешно. 	Зафиксирована попытка чтения сообщения из Kafka.
9	mp_prober_rebalance_assigned_partitions	counter	<ul style="list-style-type: none"> • topic: <ul style="list-style-type: none"> ○ prober.tasks 	<p>Количество внештатных ситуаций в работе модуля Prober, связанное с ребалансировкой. В частности, если во время ребалансировки модуль Prober ожидает подписку на одну партицию, но в процессе ребалансировки подписывается на более или менее одну партицию, то значение данной метрики увеличивается:</p> <ul style="list-style-type: none"> • prober.tasks - топик в Kafka, в который поступают задания на анализ медиафайлов. 	В результате ребалансировки партиций топика в Kafka модуль Prober ожидает подписку на определенное количество партиций, но получает подписку на большее или меньшее количество партиций.

1 0	mp_prober_connection_status	gauge	<ul style="list-style-type: none"> service: <ul style="list-style-type: none"> Kafka 	<p>Связь сервиса Prober с Kafka.</p> <ul style="list-style-type: none"> 1 - соединение присутствует; 0 - соединение отсутствует. 	Изменилось состояние связи с Kafka.
1 1	mp_prober_assigned_partitions	gauge	<ul style="list-style-type: none"> topic: <ul style="list-style-type: none"> prober.tasks 	<p>Количество назначенных партиций в Kafka:</p> <ul style="list-style-type: none"> prober.tasks - топик в Kafka, в который поступают задания на анализ медиафайлов. 	Изменение количества партиций в топике Kafka.
1 2	mp_prober_throughput_mbps	summary	<ul style="list-style-type: none"> quantile: <ul style="list-style-type: none"> 0.9 count sum 	<p>Средняя пропускная способность модуля по заданному квантилю, мегабит в секунду. Имеется в виду обобщенная скорость обработки видео контента, рассчитанная путем деления размера входного видео файла на полное время обработки задания.</p> <ul style="list-style-type: none"> sum - сумма всех значений; count - количество обновлений метрик. 	Начало или окончание обработки медиаконтента для 90% наиболее вероятных заданий.

1 3	mp_prober_task_processing_time_seconds	summary	<ul style="list-style-type: none"> • quantile: <ul style="list-style-type: none"> ○ 0.9 • count • sum 	Среднее время обработки задания по заданному квантилю, в секундах. <ul style="list-style-type: none"> • sum - сумма всех значений; • count - количество обновлений метрик. 	Обработка задания успешно завершена для 90% наиболее вероятных заданий.
1 4	mp_prober_throughput_mbps_s3	summary	<ul style="list-style-type: none"> • op_type: <ul style="list-style-type: none"> ○ read • quantile: <ul style="list-style-type: none"> ○ 0.9 ○ 0.99 	Пропускная способность S3 в секунду (Мбит/с) по заданному квантилю. <ul style="list-style-type: none"> • sum - сумма всех значений; • count - количество обновлений метрик. • read - запись файла. 	Начало или окончание загрузки данных из S3 в Prober по заданному квантилю.
1 5	mp_prober_task_waiting_time_seconds	summary	<ul style="list-style-type: none"> • quantile <ul style="list-style-type: none"> ○ 0.99 	Время, потраченное на ожидание задания по заданному квантилю. <ul style="list-style-type: none"> • sum - сумма всех значений; • count - количество обновлений метрик. 	Задание на анализ медиафайла находится в очереди и ожидает обработки для 99% наиболее вероятных заданий.

1 6	mp_prober_state	gauge	<ul style="list-style-type: none"> value: <ul style="list-style-type: none"> processing waiting 	<p>Текущее состояние сервиса: обработка задания.</p> <ul style="list-style-type: none"> processing - обработка задания; waiting- ожидание задания. 	<ul style="list-style-type: none"> Начало обработки и нового задания. Завершение обработки и текущего задания. Начало ожидания нового задания после завершения предыдущего.
1 7	mp_prober_errors	counter	<ul style="list-style-type: none"> code code_text 	<p>Счетчик ошибок по внутренним кодам ошибок с маппингом кодов на текстовые лейблы.</p> <ul style="list-style-type: none"> code - код ошибки; code_text - текст ошибки. 	<p>Появилась новая ошибка, обработчик которой существует в Медиаплатформе.</p>

3.2.3.4. Пример ответа микросервиса как экспортера Prometheus

URI для запроса метрик относительно значения переменной окружения

TELEMETRY_METRICS_ADDRESS:

- /metrics - метрики;
- /health - так называемый health-статус сервиса, т.е. признак того, что сервис жив и не "подвис".

HELP mp_prober_tasks Total number of tasks

TYPE mp_prober_tasks counter

mp_prober_tasks{status="failed"} 0

mp_prober_tasks{status="success"} 0

HELP mp_prober_s3_seconds S3 total transfer time, seconds

TYPE mp_prober_s3_seconds counter

```

mp_prober_s3_seconds{op_type="read"} 0
# HELP mp_prober_s3_bytes S3 total transfer volume, bytes
# TYPE mp_prober_s3_bytes counter
mp_prober_s3_bytes{op_type="read"} 0
# HELP mp_prober_aggregated_error Total error count aggregated by groups
# TYPE mp_prober_aggregated_error counter
mp_prober_aggregated_error{code_aggregated="processing"} 0
mp_prober_aggregated_error{code_aggregated="io"} 0
# HELP mp_prober_kafka_writed_messages Total number of sent messages
# TYPE mp_prober_kafka_writed_messages counter
mp_prober_kafka_writed_messages{status="failed"} 0
mp_prober_kafka_writed_messages{status="success"} 0
# HELP mp_prober_unhandled_messages Total number of unhandled messages
# TYPE mp_prober_unhandled_messages counter
mp_prober_unhandled_messages{type="unknown"} 0
mp_prober_unhandled_messages{type="unexpected"} 0
# HELP mp_prober_rebalance Total number of rebalance events
# TYPE mp_prober_rebalance counter
mp_prober_rebalance{POD_NAME=""} 1
# HELP mp_prober_kafka_read_messages Total number of incoming messages
# TYPE mp_prober_kafka_read_messages counter
mp_prober_kafka_read_messages{status="failed"} 0
mp_prober_kafka_read_messages{status="success"} 0
# HELP mp_prober_rebalance_assigned_partitions Total number cases of assignment to multiple or
zero partitions
# TYPE mp_prober_rebalance_assigned_partitions counter
mp_prober_rebalance_assigned_partitions{topic="prober.tasks"} 1
# HELP mp_prober_connection_status Connection status of message transport
# TYPE mp_prober_connection_status gauge
mp_prober_connection_status{service="kafka"} 1
# HELP mp_prober_assigned_partitions Count of assigned partitions
# TYPE mp_prober_assigned_partitions gauge

```

```

mp_prober_assigned_partitions{topic="prober.tasks"} 10
# HELP mp_prober_throughput_mbps Throughput per second (Mbps)
# TYPE mp_prober_throughput_mbps summary
mp_prober_throughput_mbps_count{POD_NAME=""} 0
mp_prober_throughput_mbps_sum{POD_NAME=""} 0
mp_prober_throughput_mbps{quantile="0.9"} Nan
# HELP mp_prober_task_processing_time_seconds Time spent on task handling
# TYPE mp_prober_task_processing_time_seconds summary
mp_prober_task_processing_time_seconds_count{POD_NAME=""} 0
mp_prober_task_processing_time_seconds_sum{POD_NAME=""} 0
mp_prober_task_processing_time_seconds{quantile="0.9"} Nan
# HELP mp_prober_throughput_mbps_s3 S3 throughput per second (Mbps)
# TYPE mp_prober_throughput_mbps_s3 summary
mp_prober_s3_throughput_mbps_count{op_type="read"} 0
mp_prober_s3_throughput_mbps_sum{op_type="read"} 0
mp_prober_throughput_mbps_s3{op_type="read",quantile="0.9"} Nan
mp_prober_throughput_mbps_s3{op_type="read",quantile="0.99"} Nan
# HELP mp_prober_task_waiting_time_seconds Time spent waiting for the task
# TYPE mp_prober_task_waiting_time_seconds summary
mp_prober_task_waiting_time_seconds_count{POD_NAME=""} 0
mp_prober_task_waiting_time_seconds_sum{POD_NAME=""} 0
mp_prober_task_waiting_time_seconds{quantile="0.9"} Nan
# HELP mp_prober_state Current 'sticky' state of the service. Retains 'active' value until the state is
requested.
# TYPE mp_prober_state gauge
mp_prober_state{value="waiting"} 1
mp_prober_state{value="processing"} 0
# HELP mp_prober_errors
# TYPE mp_prober_errors counter
mp_prober_errors{text_code="success"|"demultiplexing_error", code: 123455}

```

3.3. Transcoder

3.3.1. Назначение Transcoder

Сервис Transcoder предназначен для выполнения пофайлового транскодирования видео по запросам из очереди задач.

3.3.2. Обработка нескольких профилей

Transcoder имеет возможность закодировать входящий поток в несколько разных качеств, описанных профилями. При этом декодирование каждого кадра выполняется только один раз, а обработка и кодирование - столько раз, сколько профилей прописано в задаче на транскодинг.

Для каждого профиля настраивается тип медиа-контейнера (mp4, MPEG-TS, mp3 и т.д.), имя кодека, опции кодирования, настройки препроцессинга (например, разрешение и обрезка полей), возможность использовать в контейнере только выбранные дорожки (например, только одну аудио дорожку или одну аудио и одну видео дорожку).

Результирующие потоки будут сохранены в разные файлы. Количество файлов равно количеству профилей. При завершении обработки одной задачи транскодирования будет завершена запись всех файлов всех профилей.

3.3.3. Переменные среды Transcoder

В таблице ниже перечислены переменные окружения, которые используются для настройки компонента Transcoder.

Переменная	Тип	Формат	Диапазон возможных значений	Обязательно	Описание
Блок базовых настроек					
POD_NAME	String			Нет	Идентификатор пода Kubernetes, уникальное значение. Значение по умолчанию: отсутствует.

BASE_PATH	String	/file_path		Нет	<p>Базовый путь для входных и выходных файлов.</p> <p>Применяется для относительных путей в локальной файловой системе во входящих и исходящих сообщениях.</p> <p>Значение по умолчанию: отсутствует.</p>
CONNECTOR_TYPE	String		KAFKA	Да	<p>Брокер-сообщение, от которого Task Manager получает задачи.</p> <p>Используйте значение KAFKA.</p> <p>Значение по умолчанию: KAFKA.</p>
DELETE_SOURCES	Boolean		true, false или 0, 1	Нет	<p>Флаг, который показывает, нужно ли удалять исходное видео после успешного транскодирования.</p> <p>Допустимые значения:</p> <ul style="list-style-type: none"> • true - удалять исходное видео; • false - не удалять исходное видео. <p>Значение по умолчанию: false.</p>

FORCE_GPU	Bo ol		0, 1	Нет	<p>Переменная, определяющая выбор профиля транскодирования: GPU либо CPU. Так же влияет на чтение опции hw_device.</p> <p>Доступные варианты:</p> <ul style="list-style-type: none"> • FORCE_GPU не выставлена, тогда: <ul style="list-style-type: none"> ○ доступна настройка декодирования TRANSCODER_TASK.hw_device; ○ путь до профилей кодирования TRANSCODER_TASK.pfiles[*]; • FORCE_GPU=1, тогда: <ul style="list-style-type: none"> ○ доступна настройка декодирования TRANSCODER_TASK.hw_device; ○ Путь до профилей кодирования TRANSCODER_TASK.pfiles[*].gpu; • FORCE_GPU=0, тогда: <ul style="list-style-type: none"> ○ игнорируется настройка декодирования TRANSCODER_TASK.hw_device; ○ путь до профилей кодирования
-----------	----------	--	------	-----	---

					TRANSCODER_TASK.prifiles[*].cpu.
LOG_LEVEL	String		<ol style="list-style-type: none"> 1. none 2. fatal 3. critical 4. error 5. warning 6. notice 7. information 8. debug 9. trace 	Нет	<p>Уровень ведения журнала событий. Применяется как значение по умолчанию для всех служб, если система ведения журнала не сконфигурирована, или через командную строку компонента Transcoder.</p> <p>Допустимые значения:</p> <ul style="list-style-type: none"> • none - ведение журнала отключено; • fatal - неустранимая ошибка; • critical - критическая ошибка; • error - ошибка; • warning - предупреждение; • notice - уведомление; • information - информация; • debug - отладка; • trace - трассировка. <p>Значение по умолчанию: information.</p>

AWS_LOG_LEVEL	String		<ol style="list-style-type: none"> 1. none 2. fatal 3. critical 4. error 5. warning 6. notice 7. information 8. debug 9. trace 	Нет	Уровень логирования взаимодействия с облачным хранилищем S3. Значение по умолчанию: error.
PERSISTENT_CONNECTOR	Integer		<ol style="list-style-type: none"> 1. 1 2. 2 	Нет	Параметр, который указывает на доступность соединения хотя бы с одним брокером Kafka.
Блок переменных, необходимых, если CONNECTOR_TYPE=KAFKA					
KAFKA_BROKERS	String	host или host:port, значения разделяются запятыми		Да, если CONNECTOR_TYPE=KAFKA	Список брокеров Kafka в виде значений, разделённых запятыми, в формате <хост> или <хост>:<порт>. Значение по умолчанию: отсутствует.
KAFKA_GROUP_ID	String			Нет	Идентификатор группы Kafka (consumer group.id). Значение по умолчанию: 0.

KAFKA_TOPIC_PROBE	String			Да, если CONNECTOR_TYPE=KAFKA	Имя топика Kafka, куда Task Manager ставит задачи на анализ загруженного медиафайла для компонента Prober. Значение по умолчанию: KAFKA_TOPIC_PROBE.
KAFKA_TOPIC_TASK_MANAGER	String			Да, если CONNECTOR_TYPE=KAFKA	Имя топика Kafka, куда Prober отправляет сообщения о результатах анализа загруженного медиафайла: <ul style="list-style-type: none"> • завершение анализа медиафайла; • сбой при анализе медиафайла. Сообщения предназначены для считывания компонентом Task Manager. Значение по умолчанию: KAFKA_TOPIC_TASK_MANAGER.
KAFKA_SECURITY_PROTOCOL	String			Нет	Тип проверки подлинности и шифрования Kafka. Соответствует конфигурационному параметру Kafka: security.protocol.
KAFKA_SASL_MECHANISMS	String			Да, если задано KAFKA_SECURITY_PROTOCOL	Дополнительный параметр для проверки подлинности и шифрования Kafka. Соответствует конфигурационному параметру Kafka: sasl.mechanisms.
KAFKA_SASL_USERNAME	String			Да, если задано KAFKA_SECURITY_PROTOCOL	Дополнительный параметр для проверки подлинности и шифрования Kafka. Соответствует конфигурационному параметру Kafka: sasl.username.

KAFKA_SASL_PASSWORD	String			Да, если задано KAFKA_SECURITY_PROTOCOL	Дополнительный параметр для проверки подлинности и шифрования Kafka. Соответствует конфигурационному параметру Kafka: sasl.password.
Блок настроек телеметрии Это необязательные переменные — если не указано, сбор телеметрии осуществляться не будет.					
TELEMETRY_TRACES_EXPORTER_TYPE	String		<ol style="list-style-type: none"> 1. CONSOLE 2. NULL_SINK 3. FILE 	Нет	<p>Тип экспорта телеметрии. Возможные значения:</p> <ul style="list-style-type: none"> • CONSOLE; • NULL_SINK; • FILE; <p>Значение по умолчанию: NULL_SINK.</p>
TELEMETRY_TRACES_URI	String	/file_path или имя файла		Зависит от TELEMETRY_TRACES_EXPORTER_TYPE	<p>Адрес для экспорта телеметрии:</p> <ul style="list-style-type: none"> • для FILE - это имя файла: <p>Для CONSOLE и NULL_SINK эта переменная необязательна, для FILE - обязательна. Значение по умолчанию: отсутствует.</p>
TELEMETRY_METRICS_ADDRESS	String	host:port		Нет	<p>Адрес локального веб-сервера, по которому Prometheus сможет собрать метрики с конкретного экземпляра транскодера (Pod), в формате <хост>:<порт>. Значение по умолчанию: отсутствует.</p>
Блок настроек подключения к хранилищу медиаданных					
S3_KEY_ID	String			Да, если предполагается доступ к хранилищу S3	<p>Логин для подключения к хранилищу S3. Значение по умолчанию: отсутствует.</p>

S3_SECRET_KEY	String			Да, если предполагается доступ к хранилищу S3	Пароль для подключения к хранилищу S3. Значение по умолчанию: Отсутствует.
----------------------	--------	--	--	---	---

Переменные среды указываются в файле или файлах YAML. Это может быть как единый файл для всех компонентов Медиаплатформы, так и несколько файлов.

В зависимости от вариантов развёртывания переменные можно указать вручную или включить в настройки из других источников.

Ниже представлен пример настроек компонента Transcoder в нотации манифеста Kubernetes.

env:

Блок базовых настроек

- name: BASE_PATH
value: "/home/user/media"
- name: POD_NAME
value: "prober"
- name: LOG_LEVEL
value: "debug"
- name: AWS_LOG_LEVEL
value: "debug"
- name: PERSISTENT_CONNECTOR
value: 1

Настройки брокера сообщений

- name: CONNECTOR_TYPE
value: "KAFKA"
- name: KAFKA_BROKERS
value: 127.0.0.1:9093
- name: KAFKA_TOPIC_PROBE
value: KAFKA_TOPIC_PROBE
- name: KAFKA_TOPIC_TASK_MANAGER
value: KAFKA_TOPIC_TASK_MANAGER

```
- name: KAFKA_GROUP_ID
  value: 0
- name: KAFKA_SECURITY_PROTOCOL
  value: kafka_security_protocol
- name: KAFKA_SASL_MECHANISMS
  value: kafka_sasl_mechanisms
- name: KAFKA_SASL_USERNAME
  value: kafka_sasl_username
- name: KAFKA_SASL_PASSWORD
  value: kafka_sasl_password
```

Настройки подключения к хранилищу S3

```
- name: S3_ID_KEY
  value: s3_id_key
- name: S3_SECRET_KEY
  value: s3_secret_key
```

Настройка телеметрии

```
- name: TELEMETRY_METRICS_ADDRESS
  value: "0.0.0.0:8080"
- name: TELEMETRY_TRACES_EXPORTER_TYPE
  value: "CONSOLE"
- name: TELEMETRY_TRACES_URI
  value: "CONSOLE"
```

3.3.4. Сбор метрик

Отправителем метрик является сервис Transcoder, входящий в состав Медиаплатформы.

Метрики собираются с помощью системы мониторинга и оповещений Prometheus и отображаются в виде графиков, счетчиков и в других представлениях с помощью инструмента для визуализации и анализа данных Grafana.

Отображаемый отрезок времени на всех графиках по умолчанию: 6 часов.

Все метрики выводятся для списка подов или конкретного пода.

3.3.4.1. Описание функциональных клавиш Grafana

- **namespace** – выбор окружения;
- **prober_instance** – поды Prober;
- **transcoder_instance** – поды Transcoder;
- **uploader_instance** – поды Uploader;
- **taskmanager_instance** – поды Task Manager.

Для всех функциональных клавиш, кроме **namespace**, значение по умолчанию all.

3.3.4.2. Описание метрик Grafana

При выборе большого временного интервала или частом перезапуске подов на графиках и счётчиках относительно него могут отображаться все запущенные поды и их состояние. Для мониторинга последних следует выбирать более короткий интервал. Данное поведение позволяет отслеживать состояние конкретного пода в конкретный момент времени.

Разрывы на графиках и отсутствие информации возможны в случае, если после старта сервиса метрика ни разу не отсылалась или сервис не был запущен или не работает в следствии системной ошибки.

Метрики, которые регистрируют единичные всплески, могут быть не видны на графиках, где для его построения используется функции `rate` или `increase`, которые возвращают скорость изменения метрики на заданном интервале, при этом первая делит полученное значение на этот интервал. Для регистрации и отслеживания подобных изменений следует использовать счётчики.

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
1	Overall	Transcoder - общее количество ошибок по текстовым кодам после старта	Счетчик / Таблица	Общее количество ошибок по текстовым кодам ошибок с маппингом кодов на текстовые лейблы с момента последней перезагрузки.	<ul style="list-style-type: none"> code text_code transcoder_type 	<p>Метрика: mp_transcoder_errors - code, code_text, transcoder_type</p> <ul style="list-style-type: none"> <pre> {{{text_code}}}: sum(mp_transcoder_errors{code!="0", namespace=~"\$namespace", pod=~"\$transcoder_instance"}) by (code, text_code, transcoder_type) </pre>
2	Overall	Transcoder - количество ошибок по внутренним кодам (количество/с)	График	Общее количество ошибок по внутренним кодам ошибок.	<ul style="list-style-type: none"> code 	<p>Метрика: mp_transcoder_errors - code</p> <ul style="list-style-type: none"> <p>Total:</p> <pre> sum(sum(rate(mp_transcoder_errors{code!="0", namespace=~"\$namespace", pod=~"\$transcoder_instance"}[\$__rate_interval])) by (code)) </pre> <pre> {{{code}}}: sum(rate(mp_transcoder_errors{code!="0", namespace=~"\$namespace", pod=~"\$transcoder_instance"}[\$__rate_interval])) by (code) </pre>

3	Overall	Transcoder - статус подключения к сервисам	График	<p>Статус связи с сервисами :</p> <ul style="list-style-type: none"> • 1 - соединен • 0 - соединен отсутствует. <p>Все графики суммируются, каждый следующий будет выглядеть, как своё значение плюс предыдущее.</p>	<ul style="list-style-type: none"> • exported_service: ○ 	<p>Метрика: mp_transcoder_connection_status</p> <ul style="list-style-type: none"> • {{exported_service}} - {{pod}}: mp_transcoder_connection_status{exported_service="kafka", namespace=~"\$namespace", pod=~"\$transcoder_instance"}
---	---------	--	--------	--	---	---

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
4	Overall	Transcoder - kafka - статус подключения	Измеритель	Статус связи с сервисами : количество сервисов и количество подключённых к Kafka в данный момент сервисов	<ul style="list-style-type: none"> • exported_service: <ul style="list-style-type: none"> ○ 	<p>Метрика: mp_transcoder_connection_status</p> <ul style="list-style-type: none"> • количество сервисов: count(mp_transcoder_connection_status{exported_service="kafka", namespace=~"\$namespace", pod=~"\$transcoder_instance"}) • Подключённых сервисов: sum(mp_transcoder_connection_status{exported_service="kafka", namespace=~"\$namespace", pod=~"\$transcoder_instance"} >= 0 or on() vector(0))
5	Transcoder	Количество ошибок (количество/с)	График	Группы ошибок по загрузке и обработке.	<ul style="list-style-type: none"> • code_aggregated: <ul style="list-style-type: none"> ○ ○ 	<p>Метрика: mp_transcoder_aggregated_error_processing, io</p> <ul style="list-style-type: none"> • io: sum(rate(mp_transcoder_aggregated_error{code_aggregated="io", namespace=~"\$namespace", pod=~"\$transcoder_instance"}[\$__rate_interval])) >= 0 or on() vector(0) • processing: sum(rate(mp_transcoder_aggregated_error{code_aggregated="processing", namespace=~"\$namespace", pod=~"\$transcoder_instance"}[\$__rate_interval])) >= 0 or on() vector(0)

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
6	Transcoder	Статус обработки заданий	Измеритель	Текущее состояние сервиса: обработка задания.	value: <ul style="list-style-type: none"> • processing • waiting 	Метрика: mp_transcoder_state - processing, waiting <ul style="list-style-type: none"> • Processing: sum(mp_transcoder_state{ value="processing", namespace=~"\$namespace", pod=~"\$transcoder_instance" } >= 0 or on() vector(0)) • Waiting: sum(mp_transcoder_state{ value="waiting", namespace=~"\$namespace", pod=~"\$transcoder_instance" } >= 0 or on() vector(0))
7	Transcoder	Количество событий ребалансировки (количество/с)	График	Средняя экстраполированная скорость изменения величины количества событий ребалансировки подписки на входной топик Кафка.	<ul style="list-style-type: none"> • pod • job 	Метрика: mp_transcoder_rebalance <ul style="list-style-type: none"> • {{pod}}: sum(rate(mp_transcoder_rebalance{ namespace=~"\$namespace", pod=~"\$transcoder_instance" }[\$__rate_interval])) by (pod, job)

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
8	Transcoder	Ошибки ребалансировки (число/с)	График	Количество нестандартных ситуаций в работе модуля Transcoder, связанное с ребалансировкой.	<ul style="list-style-type: none"> • pod • job • topic: <ul style="list-style-type: none"> ○ 	<p>Метрика:</p> <pre>mp_transcoder_rebalance_assigned_partitions</pre> <ul style="list-style-type: none"> • {{pod}} - {{topic}}: <pre>sum(rate(mp_transcoder_rebalance_assigned_partitions{topic=~"transcoder.tasks.*", namespace=~"\$namespace", pod=~"\$transcoder_instance"}[\$__rate_interval])) by (topic, pod)</pre>
9	Transcoder	Ошибки ребалансировки после старта	Счетчик	Количество нестандартных ситуаций в работе модуля Transcoder, связанное с ребалансировкой с момента последней перезагрузки.	<ul style="list-style-type: none"> • topic: <ul style="list-style-type: none"> ○ 	<p>Метрика:</p> <pre>mp_transcoder_rebalance_assigned_partitions</pre> <ul style="list-style-type: none"> • Total: <pre>sum(mp_transcoder_rebalance_assigned_partitions{topic=~"transcoder.tasks.*", namespace=~"\$namespace", pod=~"\$transcoder_instance"})</pre>

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
10	Transcoder	Количество заданий за секунду (количество/с)	График	Средняя экстраполированная скорость изменения количества заданий за определенный период. На графике представлено общее количество успешных заданий, не успешных заданий по всем подам, а также отдельно для каждого из них.	<ul style="list-style-type: none"> • pod • status: <ul style="list-style-type: none"> ○ ○ 	<p>Метрика: mp_prober_tasks</p> <ul style="list-style-type: none"> • Total - Success: <pre>sum(rate(mp_prober_tasks{status="success", namespace=~"\$namespace", pod=~"\$transcoder_instance"}[\$__rate_interval]))</pre> • Total - Failed: <pre>sum(rate(mp_prober_tasks{status="failed", namespace=~"\$namespace", pod=~"\$transcoder_instance"}[\$__rate_interval]))</pre> • Success - {{pod}}: <pre>sum(rate(mp_prober_tasks{status="success", namespace=~"\$namespace", pod=~"\$transcoder_instance"}[\$__rate_interval])) by (pod)</pre> • Failed - {{pod}}: <pre>sum(rate(mp_prober_tasks{status="failed", namespace=~"\$namespace", pod=~"\$transcoder_instance"}[\$__rate_interval])) by (pod)</pre>

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
1	Transcoder	Общее количество заданий после старта	Счетчик	Общее количество заданий, количество успешных и не успешных заданий за время работы сервиса с момента последней перезагрузки.	<ul style="list-style-type: none"> • pod • status: <ul style="list-style-type: none"> ○ ○ 	<p>Метрика: mp_prober_tasks</p> <ul style="list-style-type: none"> • Total: sum(mp_prober_tasks { namespace=~"\$namespace", pod=~"\$transcoder_instance" }) • Success: sum(mp_prober_tasks { status="success", namespace=~"\$namespace", pod=~"\$transcoder_instance" }) • Failed: sum(mp_prober_tasks { status="failed", namespace=~"\$namespace", pod=~"\$transcoder_instance" })

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
1 2	Transcoder	Средняя пропускная способность (Мгб/с), 0.9 квантиль	График	Средняя пропускная способность сервиса по заданному квантилю, мегабит в секунду. Имеется в виду обобщенная скорость обработки видео контента, рассчитанная путем деления размера входного видео файла на полное время обработки задания.	<ul style="list-style-type: none"> • quantile : <ul style="list-style-type: none"> ○ 	Метрика: mp_transcoder_throughput_mbps <ul style="list-style-type: none"> • {{pod}}: mp_transcoder_throughput_mbps{quantile="0.9", namespace=~"\$namespace", pod=~"\$transcoder_instance"}
1 3	Transcoder	Среднее время обработки заданий (с), 0.9 квантиль	График	Среднее время обработки задания по заданному квантилю, в секундах.	<ul style="list-style-type: none"> • quantile : <ul style="list-style-type: none"> ○ 	Метрика: mp_transcoder_task_processing_time_seconds <ul style="list-style-type: none"> • {{pod}}: mp_transcoder_task_processing_time_seconds{quantile="0.9", namespace=~"\$namespace", pod=~"\$transcoder_instance"}

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
14	Transcoder	Пропускная способность чтения из S3 по квантилям (МГБ/с)	График	Пропускная способность S3 в секунду (МБТ/с) по заданному квантилю.	<ul style="list-style-type: none"> • quantile : <ul style="list-style-type: none"> ○ ○ • op_type : <ul style="list-style-type: none"> ○ ○ 	Метрика: mp_transcoder_throughput_mbps_s3 - write, read <ul style="list-style-type: none"> • {{pod}} - write - quantile - 0.9: mp_transcoder_throughput_mbps_s3{quantile="0.9", op_type="write", namespace=~"\$namespace", pod=~"\$transcoder_instance"} • {{pod}} - read - quantile - 0.9: mp_transcoder_throughput_mbps_s3{quantile="0.9", op_type="read", namespace=~"\$namespace", pod=~"\$transcoder_instance"} • {pod} - write - quantile - 0.99: mp_transcoder_throughput_mbps_s3{quantile="0.99", op_type="write", namespace=~"\$namespace", pod=~"\$transcoder_instance"} • {{pod}} - read - quantile - 0.99: mp_transcoder_throughput_mbps_s3{quantile="0.99", op_type="read", namespace=~"\$namespace", pod=~"\$transcoder_instance"}
15	Transcoder	Время ожидания задания (с), 0.9 квантиль	График	Время, потраченное на ожидание задания по заданному квантилю.	<ul style="list-style-type: none"> • quantile : <ul style="list-style-type: none"> ○ 	Метрика: mp_transcoder_task_waiting_time_seconds <ul style="list-style-type: none"> • {{pod}}: mp_transcoder_task_waiting_time_seconds{quantile="0.9", namespace=~"\$namespace", pod=~"\$transcoder_instance"}
16	Transcoder	Время чтения и записи файлов в S3 (с) после старта	График	Время чтения и записи файлов в S3 в секундах	<ul style="list-style-type: none"> • op_type : <ul style="list-style-type: none"> ○ 	метрика: mp_transcoder_s3_seconds - read <ul style="list-style-type: none"> • {{pod}} - read: mp_transcoder_s3_seconds{op_type="read", namespace=~"\$namespace", pod=~"\$transcoder_instance"}

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
17	Transcoder	Объем чтения и записи файлов из S3 (мгб/с)	График	Объем чтения и записи файлов в S3 из в байтах	<ul style="list-style-type: none"> • pod • op_type : <ul style="list-style-type: none"> ○ ○ 	Метрика: mp_transcoder_s3_bytes <ul style="list-style-type: none"> • {{pod}} - read: <pre>sum(rate(mp_transcoder_s3_bytes{op_type="read", namespace=~"\$namespace", pod=~"\$transcoder_instance"}[\$__rate_interval])) by (pod)</pre> • {{pod}} - write: <pre>sum(rate(mp_transcoder_s3_bytes{op_type="write", namespace=~"\$namespace", pod=~"\$transcoder_instance"}[\$__rate_interval])) by (pod)</pre>
18	Transcoder	Общий объем чтения файлов из S3 в мегабайтах с момента перезапуска модуля (мгб)	Счетчик	Общий объем чтения и записи файлов из S3 в мегабайтах с момента перезапуска модуля.	<ul style="list-style-type: none"> • pod • op_type : <ul style="list-style-type: none"> ○ ○ 	Метрика: mp_transcoder_s3_bytes <ul style="list-style-type: none"> • Read: <pre>(sum(mp_transcoder_s3_bytes{op_type="read", namespace=~"\$namespace", pod=~"\$transcoder_instance"}) >=0 or on() vector(0))/1e+6</pre> • Write: <pre>(sum(mp_transcoder_s3_bytes{op_type="write", namespace=~"\$namespace", pod=~"\$transcoder_instance"}) >=0 or on() vector(0))/1e+6</pre>

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
19	Transcoder	Сообщения в Kafka (количество/с)	График	Количество сообщений, полученных, необработанных и отправленных в Kafka.	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ ○ • type: <ul style="list-style-type: none"> ○ ○ 	<p>Метрики: mp_transcoder_kafka_writed_messages, mp_transcoder_unhandled_messages, mp_transcoder_kafka_read_messages</p> <ul style="list-style-type: none"> • Send - success: sum(rate(mp_transcoder_kafka_writed_messages{status="success", namespace=~"\$namespace", pod=~"\$transcoder_instance"}[\$__rate_interval])) >=0 or on() vector(0) • Send - failure: sum(rate(mp_transcoder_kafka_writed_messages{status="failed", namespace=~"\$namespace", pod=~"\$transcoder_instance"}[\$__rate_interval])) >=0 or on() vector(0) • Unhandled - unexpected: sum(rate(mp_transcoder_unhandled_messages{type="unexpected", namespace=~"\$namespace", pod=~"\$transcoder_instance"}[\$__rate_interval])) >=0 or on() vector(0) • Unhandled - unknown: sum(rate(mp_transcoder_unhandled_messages{type="unknown", namespace=~"\$namespace", pod=~"\$transcoder_instance"}[\$__rate_interval])) >=0 or on() vector(0) • Read - success: sum(rate(mp_transcoder_kafka_read_messages{status="success", namespace=~"\$namespace", pod=~"\$transcoder_instance"}[\$__rate_interval])) >=0 or on() vector(0) • Read - failure: sum(rate(mp_transcoder_kafka_read_messages{status="failed", namespace=~"\$namespace", pod=~"\$transcoder_instance"}[\$__rate_interval])) >=0 or on() vector(0)

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
20	Transcoder	Общее количество сообщений Kafka после старта	Счетчик	Общее количество сообщений, полученных, необработанных и отправленных в Kafka с момента перезапуска модуля.	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ ○ • type: <ul style="list-style-type: none"> ○ ○ 	<p>Метрики: mp_transcoder_kafka_writed_messages, mp_transcoder_unhandled_messages, mp_transcoder_kafka_read_messages</p> <ul style="list-style-type: none"> • Send - success: sum(mp_transcoder_kafka_writed_messages{status="success", namespace=~"\$namespace", pod=~"\$transcoder_instance"}) >=0 or on() vector(0) • Send - failure: sum(mp_transcoder_kafka_writed_messages{status="failed", namespace=~"\$namespace", pod=~"\$transcoder_instance"}) >=0 or on() vector(0) • Unhandled - unexpected: sum(mp_transcoder_unhandled_messages{type="unexpected", namespace=~"\$namespace", pod=~"\$transcoder_instance"}) >=0 or on() vector(0) • Unhandled - unknown: sum(mp_transcoder_unhandled_messages{type="unknown", namespace=~"\$namespace", pod=~"\$transcoder_instance"}) >=0 or on() vector(0) • Read - success: sum(mp_transcoder_kafka_read_messages{status="success", namespace=~"\$namespace", pod=~"\$transcoder_instance"}) >=0 or on() vector(0) • Read - failure: sum(mp_transcoder_kafka_read_messages{status="failed", namespace=~"\$namespace", pod=~"\$transcoder_instance"}) >=0 or on() vector(0)

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
2 1	Transcoder	Количество назначенных партиций в Kafka	График	Количество назначенных партиций в Kafka	<ul style="list-style-type: none"> topic: <ul style="list-style-type: none"> ○ 	Метрика: mp_transcoder_assigned_partitions <ul style="list-style-type: none"> Assigned partitions: sum(mp_transcoder_assigned_partitions{topic=~"transcoder.tasks.*", namespace=~"\$namespace", pod=~"\$transcoder_instance"})
2 2	Transcoder	Продолжительность медиафайла / время транскодирования, 0.9 квантиль (с)	График	Соотношение продолжительности медиафайла к времени транскодирования по заданному квантилю.	<ul style="list-style-type: none"> quantile: <ul style="list-style-type: none"> ○ 	Метрика: mp_transcoder_transcoding_rate <ul style="list-style-type: none"> {{pod}}: mp_transcoder_transcoding_rate{quantile="0.9", namespace=~"\$namespace", pod=~"\$transcoder_instance"}

3.3.4.3. Описание метрик Prometheus

- gauge - масштабируемый диалоговый индикатор, который отображает текущее значение метрики в процентах или числах. Большим преимуществом Gauge в Grafana является то, что он может легко приспосабливаться к разным значениям, чтобы обеспечить оптимальное отображение информации на дашборде: может быть использован для отображения любых числовых данных, таких как счётчики, процентные доли, абсолютные значения и т.д., т.к. gauge измеряет числовое значение в

определенный момент времени, а не считает скорость изменения значений, как Counter. Gauge в Grafana можно отображать не только в виде графика, но и в виде других типов визуализации, таких как текст, таблицы или специально созданные плагины.

- counter - счетчик, который увеличивается на каждое новое событие. Счетчики обычно могут только увеличиваться, но не уменьшаться, и сброс в ноль может потребоваться вручную.
- summary - собирает информацию о распределении значений заданной величины. Это позволяет собирать статистику о количестве значений, их среднем, медиане, процентилях и т.д. Для каждого Summary Prometheus собирает образцы (samples) для значений, которые необходимо агрегировать. Для каждого образца указывается значение и метка времени, когда оно было собрано. Затем Prometheus вычисляет ряд статистических значений на основе этих образцов, включая сумму всех значений, количество собранных образцов и среднее значение. В Grafana представление Summary может использоваться для создания графиков и диаграмм, показывающих распределение значений величины. Графики Summary могут быть полезны для анализа производительности приложения, когда необходимо определить, какое количество запросов было выполнено за определенный период времени, и как быстро были выполнены запросы.

Ниже приведена таблица с описанием метрик Prometheus для модуля Transcoder.

	Наименование метрики	Тип	Метки	Описание	Событие
1	mp_prober_tasks	counter	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ success ○ failed 	<p>Количество выполненных заданий за период по типам:</p> <ul style="list-style-type: none"> • success - успешно; • failed - неуспешно 	Появилось новое задание на транскодирование медиафайла.
2	mp_transcoder_s3_seconds	counter	<ul style="list-style-type: none"> • op_type: <ul style="list-style-type: none"> ○ read 	<p>Общее время передачи файлов из/в S3, в секундах.</p> <ul style="list-style-type: none"> • read - чтение файла. 	Зафиксирована попытка записи или чтения файла из хранилища S3.

3	mp_transcoder_s3_bytes	counter	<ul style="list-style-type: none"> op_type: <ul style="list-style-type: none"> write read 	<p>Объем передачи файлов из/в S3, в байтах.</p> <ul style="list-style-type: none"> write - запись файла; read - чтение файла. 	<p>Зафиксирована попытка записи или чтения файла из хранилища S3.</p>
4	mp_transcoder_aggregated_error	counter	<ul style="list-style-type: none"> code_aggregated: <ul style="list-style-type: none"> processing io 	<p>Метрика по группам ошибок.</p> <ul style="list-style-type: none"> processing - обработка; io - загрузка. 	<p>Появилась новая ошибка, обработчик которой существует в Медиаплатформе.</p>
5	mp_transcoder_kafka_writed_messages	counter	<ul style="list-style-type: none"> status: <ul style="list-style-type: none"> failed success 	<p>Общее количество отправленных сообщений в Apache Kafka.</p> <ul style="list-style-type: none"> failed - неуспешно; success - успешно. 	<p>Зафиксирована попытка отправки сообщения в Kafka.</p>
6	mp_transcoder_unhandled_messages	counter	<ul style="list-style-type: none"> type: <ul style="list-style-type: none"> unknown unexpected 	<p>Общее количество необработанных сообщений.</p> <ul style="list-style-type: none"> unknown - сервис не смог идентифицировать сообщение; unexpected - сервис принял сообщение, которое ему не предназначено. 	<p>Появился новый запрос, на который модуль не подписывался или который не удалось разобрать (получен не валидный json).</p>

7	mp_transcoder_rebalance	counter	-	Количество событий ребалансировки подписки на входной топик Kafka за период.	<p>Зафиксировано событие ребалансировки подов: если один из подов модуля Transcoder перестает работать, Kafka замечает, что потребитель для некоторых партиций топика не отвечает, и начинает ребалансировку, перераспределяя партиции между другими подами модуля Transcoder. В этот момент все работающие поды модуля получают сигнал о начале ребалансировки. Количество таких событий и отражает данную метрику. Возможные причины роста метрики:</p> <ol style="list-style-type: none"> 1. Аппаратные проблемы. Необходимо проверить железные метрики и CPU,
---	--------------------------------	---------	---	--	--

					GPU, RAM. 2. Проблемы со входными файлами. Необходимо проверить логи Kibana.
8	mp_transcoder_kafka_read_messages	counter	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ failed ○ success 	Общее количество входящих сообщений. <ul style="list-style-type: none"> • failed - неуспешно ; • success - успешно. 	Зафиксирована попытка чтения сообщения из Kafka.

9	mp_transcoder_rebalance_assigned_partitions	count er	<ul style="list-style-type: none"> • topic: <ul style="list-style-type: none"> ○ transcoder.tasks.cpu ○ transcoder.tasks.gpu 	<p>Количество внештатных ситуаций в работе модуля Transcoder, связанное с ребалансировкой. В частности, если во время ребалансировки модуль Transcoder ожидает подписку на одну партицию, но в процессе перебалансировки подписывается на более или менее одну партицию, то значение данной метрики увеличивается.</p> <ul style="list-style-type: none"> • transcoder.tasks.cpu - топик в Kafka, в который поступают задания для транскодирования на cpu; • transcoder.tasks.gpu - топик в Kafka, в который поступают задания для транскодирования на gpu. 	<p>В результате ребалансировки партиций топика в Kafka модуль Transcoder ожидает подписку на определенное количество партиций, но получает подписку на большее или меньшее количество партиций.</p>
---	--	-------------	--	---	---

1 0	mp_transcoder_connection_status	gauge	<ul style="list-style-type: none"> • service: <ul style="list-style-type: none"> ○ Kafka 	Связь сервиса Transcoder с Kafka. <ul style="list-style-type: none"> • 1 - соединени е присутствие; • 0 - соединени е отсутстует. 	Изменилось состояние связи с Kafka.
1 1	mp_transcoder_assigned_partitions	gauge	<ul style="list-style-type: none"> • topic: <ul style="list-style-type: none"> ○ transcod er.tasks.c pu ○ transcod er.tasks. gpu 	Количество назначенных партиций в Kafka. <ul style="list-style-type: none"> • transcoder.tasks.cpu - топик в Kafka, в который поступают задания для транскоди рования на cpu; • transcoder.tasks.gpu - топик в Kafka, в который поступают задания для транскоди рования на gpu. 	Изменение количества партиций в топике Kafka.

1 2	mp_transcoder_throughput_mbps	summary	<ul style="list-style-type: none"> • quantile: <ul style="list-style-type: none"> ○ 0.9 • count • sum 	<p>Средняя пропускная способность модуля по заданному квантилю, мегабит в секунду. Имеется в виду обобщенная скорость обработки медиаконтента, рассчитанная путем деления размера входного видео файла на полное время обработки задания.</p> <ul style="list-style-type: none"> • sum - сумма всех значений; • count - количество обновлений метрик. 	Начало или окончание обработки медиаконтента для 90% наиболее заданий.
1 3	mp_transcoder_task_processing_time_seconds	summary	<ul style="list-style-type: none"> • quantile: <ul style="list-style-type: none"> ○ 0.9 • count • sum 	<p>Среднее время обработки задания по заданному квантилю, в секундах.</p> <ul style="list-style-type: none"> • sum - сумма всех значений; • count - количество обновлений метрик. 	Обработка задания успешно завершена для 90% наиболее вероятных заданий.
1 4	mp_transcoder_transcoding_rate	summary	<ul style="list-style-type: none"> • quantile: <ul style="list-style-type: none"> ○ 0.9 • count • sum 	<p>Соотношение продолжительности и медиафайла к времени транскодирования.</p> <ul style="list-style-type: none"> • sum - сумма всех значений; • count - количество обновлений метрик. 	Запущена операция транскодирования медиафайла для 90% наиболее вероятных заданий.

1 5	mp_transcoder_throughput_mbps_s3	summary	<ul style="list-style-type: none"> • op_type: <ul style="list-style-type: none"> ○ write ○ read • quantile: <ul style="list-style-type: none"> ○ 0.5 ○ 0.8 ○ 0.9 ○ 0.95 ○ 0.99 • count • sum 	<p>Пропускная способность S3 в секунду (Мбит/с) по заданному квантилю.</p> <ul style="list-style-type: none"> • sum - сумма всех значений; • count - количество обновлений метрик. • write - запись файла; • read - чтение файла. 	<p>Начало или окончание загрузки данных из Transcoder в S3 и наоборот по заданному квантилю.</p>
1 6	mp_transcoder_task_waiting_time_seconds	summary	<ul style="list-style-type: none"> • quantile <ul style="list-style-type: none"> ○ 0.99 • count • sum 	<p>Время, потраченное на ожидание задания по заданному квантилю.</p> <ul style="list-style-type: none"> • sum - сумма всех значений; • count - количество обновлений метрик. 	<p>Задание на транскодирование медиафайла находится в очереди и ожидает обработки для 99% наиболее вероятных заданий.</p>

17	mp_transcoder_state	gauge	<ul style="list-style-type: none"> • value: <ul style="list-style-type: none"> ○ processing ○ waiting 	<p>Текущее состояние сервиса: обработка задания.</p> <ul style="list-style-type: none"> • processing - обработка задания; • waiting - ожидание задания. 	<ul style="list-style-type: none"> • Начало обработки нового задания. • Завершение обработки текущего задания. • Начало ожидания нового задания после завершения предыдущего.
18	mp_transcoder_errors	counter	<ul style="list-style-type: none"> • code • code_text • transcoder_type: <ul style="list-style-type: none"> ○ cpu ○ gpu ○ any 	<p>Счетчик ошибок по внутренним кодам ошибок с маппингом кодов на текстовые лейблы и по типу транскодера.</p> <ul style="list-style-type: none"> • code - код ошибки; • code_text - текст ошибки. 	<p>Появилась новая ошибка, обработчик которой существует в Медиаплатформе.</p>

3.3.4.4. Пример ответа микросервиса как экспортера Prometheus

URI для запроса метрик относительно значения переменной окружения

TELEMETRY_METRICS_ADDRESS:

- /metrics - метрики;
- /health - так называемый health-статус сервиса, т.е. признак того, что сервис жив и не "подвис".

HELP mp_prober_tasks Total number of tasks

```

# TYPE mp_prober_tasks counter
mp_prober_tasks{status="failed"} 0
mp_prober_tasks{status="success"} 0
# HELP mp_transcoder_s3_seconds S3 total transfer time, seconds
# TYPE mp_transcoder_s3_seconds counter
mp_transcoder_s3_seconds{op_type="read"} 0
# HELP mp_transcoder_s3_bytes S3 total transfer volume, bytes
# TYPE mp_transcoder_s3_bytes counter
mp_transcoder_s3_bytes{op_type="write"} 0
mp_transcoder_s3_bytes{op_type="read"} 0
# HELP mp_transcoder_aggregated_error Total error count aggregated by groups
# TYPE mp_transcoder_aggregated_error counter
mp_transcoder_aggregated_error{code_aggregated="processing"} 0
mp_transcoder_aggregated_error{code_aggregated="io"} 0
# HELP mp_transcoder_kafka_writed_messages Total number of sent messages
# TYPE mp_transcoder_kafka_writed_messages counter
mp_transcoder_kafka_writed_messages{status="failed"} 0
mp_transcoder_kafka_writed_messages{status="success"} 0
# HELP mp_transcoder_unhandled_messages Total number of unhandled messages
# TYPE mp_transcoder_unhandled_messages counter
mp_transcoder_unhandled_messages{type="unknown"} 0
mp_transcoder_unhandled_messages{type="unexpected"} 0
# HELP mp_transcoder_rebalance Total number of rebalance events
# TYPE mp_transcoder_rebalance counter
mp_transcoder_rebalance{ } 1
# HELP mp_transcoder_kafka_read_messages Total number of incoming messages
# TYPE mp_transcoder_kafka_read_messages counter
mp_transcoder_kafka_read_messages{status="failed"} 0
mp_transcoder_kafka_read_messages{status="success"} 0
# HELP mp_transcoder_rebalance_assigned_partitions Total number cases of assignment to
multiple or zero partitions
# TYPE mp_transcoder_rebalance_assigned_partitions counter

```

```

mp_transcoder_rebalance_assigned_partitions{topic="transcoder.tasks.cpu"} 0
# HELP mp_transcoder_connection_status Connection status of message transport
# TYPE mp_transcoder_connection_status gauge
mp_transcoder_connection_status{service="kafka"} 1
# HELP mp_transcoder_assigned_partitions Count of assigned partitions
# TYPE mp_transcoder_assigned_partitions gauge
mp_transcoder_assigned_partitions{topic="transcoder.tasks.cpu"} 1
# HELP mp_transcoder_throughput_mbps Throughput per second (Mbps)
# TYPE mp_transcoder_throughput_mbps summary
mp_transcoder_throughput_mbps_count{ } 0
mp_transcoder_throughput_mbps_sum{ } 0
mp_transcoder_throughput_mbps{quantile="0.9"} Nan
# HELP mp_transcoder_task_processing_time_seconds Time spent on task handling
# TYPE mp_transcoder_task_processing_time_seconds summary
mp_transcoder_task_processing_time_seconds_count{ } 0
mp_transcoder_task_processing_time_seconds_sum{ } 0
mp_transcoder_task_processing_time_seconds{quantile="0.9"} Nan
# HELP mp_transcoder_transcoding_rate Rate of video duration to transcoding time
# TYPE mp_transcoder_transcoding_rate summary
mp_transcoder_transcoding_rate_count{ } 0
mp_transcoder_transcoding_rate_sum{ } 0
mp_transcoder_transcoding_rate{quantile="0.9"} Nan
# HELP mp_transcoder_throughput_mbps_s3 S3 throughput per second (Mbps)
# TYPE mp_transcoder_throughput_mbps_s3 summary
mp_transcoder_throughput_mbps_s3_count{op_type="write"} 0
mp_transcoder_throughput_mbps_s3_sum{op_type="write"} 0
mp_transcoder_throughput_mbps_s3{op_type="write",quantile="0.9"} Nan
mp_transcoder_throughput_mbps_s3{op_type="write",quantile="0.99"} Nan
mp_transcoder_throughput_mbps_s3_count{op_type="read"} 0
mp_transcoder_throughput_mbps_s3_sum{op_type="read"} 0
mp_transcoder_throughput_mbps_s3{op_type="read",quantile="0.5"} Nan
mp_transcoder_throughput_mbps_s3{op_type="read",quantile="0.8"} Nan

```

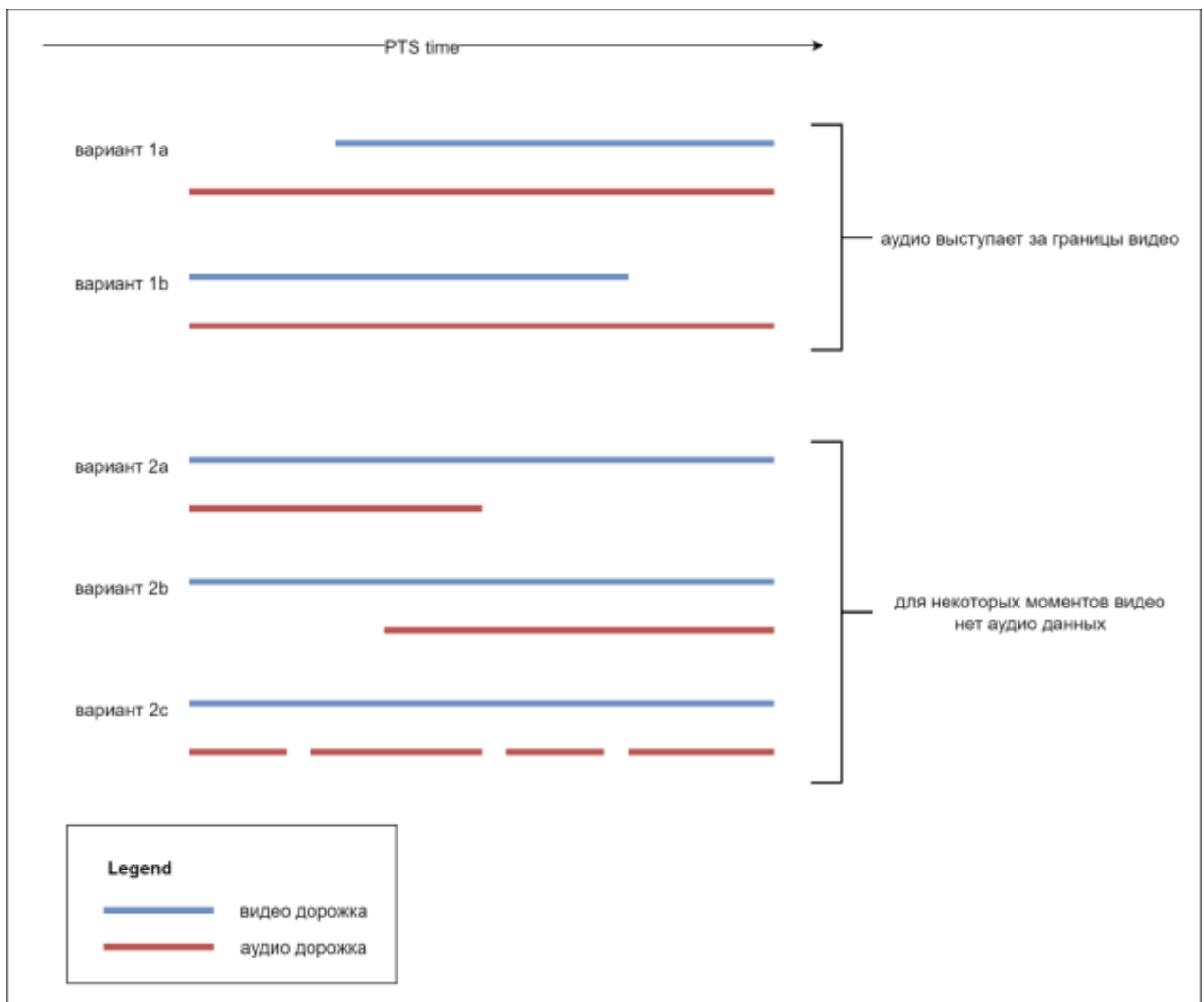
```

mp_transcoder_throughput_mbps_s3{op_type="read",quantile="0.9"} Nan
mp_transcoder_throughput_mbps_s3{op_type="read",quantile="0.95"} Nan
mp_transcoder_throughput_mbps_s3{op_type="read",quantile="0.99"} Nan
# HELP mp_transcoder_task_waiting_time_seconds Time spent waiting for the task
# TYPE mp_transcoder_task_waiting_time_seconds summary
mp_transcoder_task_waiting_time_seconds_count{ } 0
mp_transcoder_task_waiting_time_seconds_sum{ } 0
mp_transcoder_task_waiting_time_seconds{quantile="0.9"} Nan
# HELP mp_transcoder_state Current 'sticky' state of the service. Retains 'active' value until the state
is requested.
# TYPE mp_transcoder_state gauge
mp_transcoder_state{value="waiting"} 1
mp_transcoder_state{value="processing"} 0
# HELP mp_taskmanager_result
# TYPE mp_taskmanager_result counter
mp_taskmanager_result{code=102345,code_text="ok",transcoder_type="cpu"}

```

3.3.5. Алгоритм выравнивания аудио и видео дорожек

Transcoder в общем случае оперирует несколькими видео и аудио дорожками, которые могут не совпадать. Варианты различий приведены на рисунке ниже.



Для выравнивания дорожек Transcoder сначала определяет *основную* видео дорожку, относительно которой производит выравнивание. Алгоритм выбора по приоритету условий:

1. если ffmpeg предоставляет метаинформацию о том, что некоторая видео дорожка является лучшей, она будет также выбрана в качестве основной для выравнивания;
2. если ffmpeg предоставляет метаинформацию о длительности всех видеодорожек, в качестве основной будет выбрана самая длинная среди них;
3. если ffmpeg предоставляет метаинформацию о размере кадров всех видеодорожек, в качестве основной будет выбрана та, произведение ширины на высоту кадра которой является максимальным;
4. в ином случае в качестве основной выбирается видеодорожка с минимальным индексом (номер потока в контексте демуксера).

Выравнивание аудио дорожек относительно основной видео дорожки выполняется методом дополнения аудио "тишиной" либо отбрасыванием части аудио данных. С помощью ручной генерации аудио фреймов с тишиной и фильтра ffmpeg:

```
aresample=async=1:first_pts=X
```

Есть два основных случая, когда аудио выступает за границы видео, либо когда для некоторых моментов видео нет аудио данных. Комбинация этих вариантов выполняется соответственно несколькими последовательными выравниваниями. Рассмотрим варианты на диаграмме выше:

- вариант 1a. Лишнее аудио в начале клипа будет отброшено, если ffmpeg предоставил метаинформацию о начале основной видеодорожки (pts первого видео кадра). Иначе выравнивание не будет выполнено;
- вариант 1b. Лишнее аудио в конце клипа будет оставлено как есть, т.е. выравнивание не будет выполняться безусловно. Т.к. пока нет данных как поведет себя плеер в этом случае, но есть предположение, что плеер самостоятельно отбросит лишние аудио данные;
- вариант 2a. Недостающие аудио данные в конце клипа будут дополнены тишиной;
- вариант 2b. Недостающие аудио данные в начале клипа будут дополнены тишиной, если ffmpeg предоставил метаинформацию о начале "основной" видеодорожки (pts первого видео кадра). Иначе выравнивание не будет выполнено;
- вариант 2c. Недостающие аудио данные в середине клипа будут дополнены тишиной.

Фильтр aresample с параметром async=1 дополняет и заполняет тишиной промежутки в аудио основываясь на заданном samplerate. Т.е. фильтр знает, что в каждую конкретную секунду должно быть определенное количество аудио семплов. И, если их меньше, дополняет нулевыми семплами, а если больше, удаляет избыточные. Соответственно на выходе мы получаем аудио без "дырок", т.е. необходимое количество аудио семплов на всю продолжительность аудио дорожки.

3.3.6. Алгоритм масштабирования видео

3.3.6.1. Об алгоритме масштабирования видео

Модуль Transcoder преобразует исходное видео в несколько качеств, а также генерирует картинки thumbnail и trickmode. Сообщение [TRANSCODER_TASK](#) содержит указания, как нужно преобразовывать видео и какие картинки генерировать. Но это не прямые указания, а

описание желаемого результата. По крайней мере это касается размера и ориентации изображения. Сервис Transcoder в зависимости от входящего видео подбирает наилучший способ преобразования изображения, чтобы получить запрошенный результат.

Например, это означает, что если ориентация (Landscape, Portrait) исходного видео не совпадает с ориентацией выходного видео (filter_options.width и filter_options.height), то модуль Transcoder применит filter_options наоборот (большую сторону к большей, а меньшую к меньшей). Кроме того, всегда и безусловно модуль Transcoder пытается сохранить исходные пропорции видео и четную кратность выходного разрешения (чтобы ширина и высота делилась на 2, этого требуют многие кодеки). Из-за требования к четной кратности иногда DAR выходного видео может незначительно отличаться от DAR исходного видео. Также модуль Transcoder всегда делает пиксели квадратными у выходного видео (SAR 1:1).

Рассмотрим различные варианты, и предполагаемый результат:

		TRANSCODER_TASK				
		1280x720 [DAR 16:9]	720x1280 [DAR 9:16]	1024x768 [DAR 4:3]	768x1024 [DAR 3:4]	1000x1000 [DAR 1:1]
Исходное видео	1920x1080 [SAR 1:1 DAR 16:9]	1280x720 [SAR 1:1 DAR 16:9]	1280x720 [SAR 1:1 DAR 16:9]	1024x576 [SAR 1:1 DAR 16:9]	1024x576 [SAR 1:1 DAR 16:9]	1000x562 [SAR 1:1 DAR 500:281]
	1080x1920 [SAR 1:1 DAR 9:16]	720x1280 [SAR 1:1 DAR 9:16]	720x1280 [SAR 1:1 DAR 9:16]	576x1024 [SAR 1:1 DAR 9:16]	576x1024 [SAR 1:1 DAR 9:16]	562x1000 [SAR 1:1 DAR 281:500]
	1600x1200 [SAR 1:1 DAR 4:3]	960x720 [SAR 1:1 DAR 4:3]	960x720 [SAR 1:1 DAR 4:3]	1024x768 [SAR 1:1 DAR 4:3]	1024x768 [SAR 1:1 DAR 4:3]	1000x750 [SAR 1:1 DAR 4:3]
	1200x1600 [SAR 1:1 DAR 3:4]	720x960 [SAR 1:1 DAR 3:4]	720x960 [SAR 1:1 DAR 3:4]	768x1024 [SAR 1:1 DAR 3:4]	768x1024 [SAR 1:1 DAR 3:4]	750x1000 [SAR 1:1 DAR 3:4]
	2000x2000 [SAR 1:1 DAR 1:1]	720x720 [SAR 1:1 DAR 1:1]	720x720 [SAR 1:1 DAR 1:1]	768x768 [SAR 1:1 DAR 1:1]	768x768 [SAR 1:1 DAR 1:1]	1000x1000 [SAR 1:1 DAR 1:1]
	1440x1080 [SAR 4:3 DAR 16:9]	1280x720 [SAR 1:1 DAR 16:9]	1280x720 [SAR 1:1 DAR 16:9]	1024x576 [SAR 1:1 DAR 16:9]	1024x576 [SAR 1:1 DAR 16:9]	1000x562 [SAR 1:1 DAR 500:281]

Один и тот же алгоритм применяется как для профилей, так и для thumbnail и trickmode. Исключением является только то, что для thumbnail и trickmode делается автоповорот изображения в конце всех преобразований, если в метаданных исходного видео была информация о повороте.

Например:

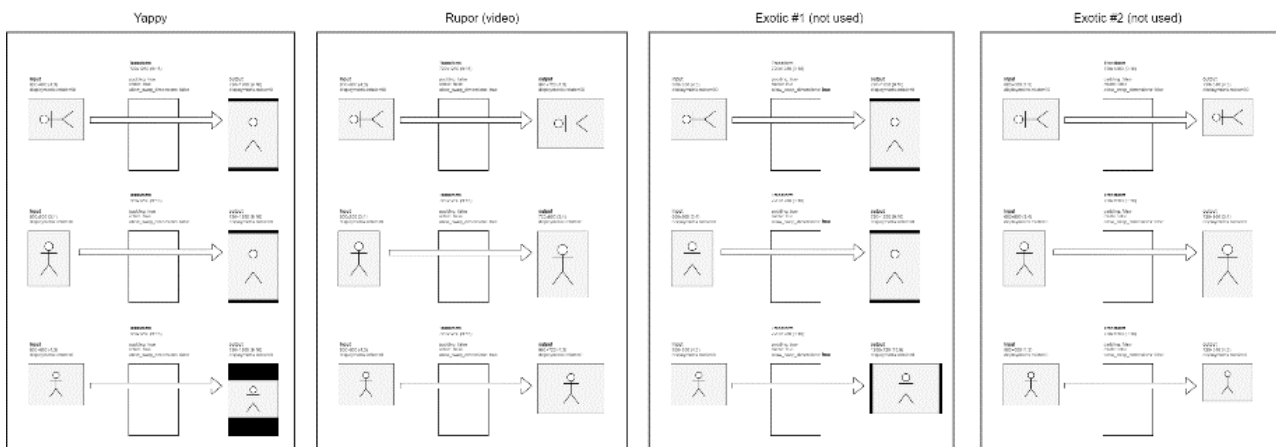
		TRANSCODER_TASK				
		1280x720 [DAR 16:9]	720x1280 [DAR 9:16]	1024x768 [DAR 4:3]	768x1024 [DAR 3:4]	1000x1000 [DAR 1:1]
исходное видео	1920x1080 [SAR 1:1 DAR 16:9 metadata: rotation 90]	video: 1280x720 [SAR 1:1 DAR 16:9 metadata: rotation 90]	1280x720 [SAR 1:1 DAR 16:9 metadata: rotation 90]	1024x576 [SAR 1:1 DAR 16:9 metadata: rotation 90]	1024x576 [SAR 1:1 DAR 16:9 metadata: rotation 90]	1000x562 [SAR 1:1 DAR 500:281 metadata: rotation 90]
	metadata: rotation 90]	thumbnail/trickmode: 720x1280 [SAR 1:1 DAR 9:16]	thumbnail/trickmode: 720x1280 [SAR 1:1 DAR 9:16]	thumbnail/trickmode: 576x1024 [SAR 1:1 DAR 9:16]	thumbnail/trickmode: 576x1024 [SAR 1:1 DAR 9:16]	thumbnail/trickmode: 562x1000 [SAR 1:1 DAR 281:500]

3.3.6.2. Особенности работы с настройками padding, rotate, allow_swap_dimensions

Описанный выше алгоритм актуален также для следующих настроек padding=false, rotate=false, allow_swap_dimensions=true.

Настройки опциональные, значения совпадают с настройками по умолчанию.

При изменении настроек алгоритм работы меняется, что проиллюстрировано на следующей схеме:



3.3.7. Допустимые аудио и видео временные метки в исходных файлах

Transcoder ожидает, что исходный пользовательский файл в достаточной степени корректно сформирован и не требует серьезных исправлений для транскодирования. В частности это касается временных меток (pts/dts) пакетов аудио и видео. Transcoder полагается на то, что каждому пакету аудио и видео задан корректный pts и dts, и что эти значения монотонно возрастают и соответствуют частоте сэмпирования аудио и частоте кадров видео. Если

исходный файл не соответствует этому требованию, транскодирование может завершиться с ошибкой.

Transcoder может попытаться исправить некоторые "проблемы" временных меток:

- экстраполяция pts и dts. Например, если у некоторых пакетов отсутствуют временные метки, Transcoder попытается вычислить их самостоятельно на основе pts и dts предыдущих пакетов и информации о частоте сэмплирования аудио и частоте кадров видео. Если же временные метки отсутствуют у первого пакета в исходном файле, транскодер не сможет это исправить;
- "дырки" или сдвиг аудио и видео дорожек относительно друг друга.

3.4. Опции энкодеров

Медиаплатформа поддерживает два видео энкодера:

1. libx264 для обработки на CPU;
2. h264_nvenc для обработки на GPU.

Через профили транскодирования можно передать все поддерживаемые видео энкодерами опции.

3.4.1. Опции, поддерживаемые h264_nvenc

Ниже представлен полный список опций, которые поддерживает h264_nvenc.

Option	Type	Possible values	Description
preset	enum		Encoding preset
		medium	hq 1 pass
		fast	hp 1 pass
		ll	low latency
		llhq	low latency hq
		llhp	low latency hp
		p1	fastest (lowest quality)
		p2	faster (lower quality)
		p3	fast (low quality)
		p4	medium (default)
		p5	slow (good quality)
		p6	slower (better quality)

		p7	slowest (best quality)
tune	enum		Tune the encoding params
		hq	High quality
		ll	Low latency
		ull	Ultra low latency
		lossless	Lossless
profile	enum	baseline, main, high, high444p	Encoding profile
level	enum	auto 1.0, 1.0b, 1.1, 1.2, 1.3 2.0, 2.1, 2.2 3.0, 3.1, 3.2 4.0, 4.1, 4.2 5.0, 5.1, 5.2 6.0, 6.1, 6.2	Specify level (as defined by Annex A).
rc	enum		Encoding constraints
		constqp	Constant QP mode
		vbr	Variable bitrate mode
		cbr	Constant bitrate mode
		vbr_minqp	Variable bitrate mode with MinQP (deprecated)
		cbr_ld_hq	Constant bitrate low delay high quality mode
		cbr_hq	Constant bitrate high quality mode
		vbr_hq	Variable bitrate high quality mode
gpu	int	[0;num_gpu]	GPU to be used for encoding
coder	enum		Entropy coding mode
		default	default value
		auto	Defined by encoder
		cabac	Context-adaptive binary arithmetic coding entropy coding mode.
		cavlc	Context-adaptive variable-length coding entropy coding mode.
b_ref_mode	enum		B-frames refernce mode
		disabled	B frames will not be used for reference
		each	Each B frame will be used for reference

		middle	Only (number of B frames)/2 will be used for reference
delay	int		Delay frame output by the given amount of frames
no-scenecut	bool		When lookahead is enabled set this to 1 to disable adaptive I-frame insertion at scene cuts
forced-idr	bool		If forcing keyframes force them as IDR frames
b_adapt	bool		When lookahead is enabled set this to 0 to disable adaptive B-frame decision
spatial-aq	bool		Set to 1 to enable Spatial AQ (Adaptive Quantization)
temporal-aq	bool		Set to 1 to enable Temporal AQ
zerolatency	bool		Set 1 to indicate zero latency operation (no reordering delay)
nonref_p	bool		Set this to 1 to enable automatic insertion of non-reference P-frames
strict_gop	bool		Set 1 to minimize GOP-to-GOP rate fluctuations
aud	bool		Use access unit delimiters
bluray-compat	bool		Bluray compatibility workarounds
init_qpP	int		Initial QP value for P frame
init_qpB	int		Initial QP value for B frame
init_qpI	int		Initial QP value for I frame
cq	float		Set target quality level (0 to 51 0 means automatic) for constant quality mode in VBR rate control
qp	int		Constant quantization parameter rate control method
qp_cb_offset	int		Quantization parameter offset for cb channel
qp_cr_offset	int		Quantization parameter offset for cr channel
a53cc	bool		Use A53 Closed Captions (if available)
dpb_size	int		Specifies the DPB size used for encoding (0 means automatic)
ldkfs	int		Low delay key frame scale; Specifies the Scene Change frame size increase allowed in case of single frame VBV and CBR
extra_sei	bool		Pass on extra SEI data (e.g. a53 cc) to be included in the bitstream
udu_sei	bool		Pass on user data unregistered SEI if available
intra-refresh	bool		Use Periodic Intra Refresh instead of IDR frames
single-slice-intra-refresh	bool		Use single slice intra refresh

constrained-encoding	bool		Enable constrainedFrame encoding where each slice in the constrained picture is independent of other slices
rc-lookahead	int		Enable constrainedFrame encoding where each slice in the constrained picture is independent of other slices
surfaces	int		Number of frames to look ahead for rate-control

3.4.2. Опции, поддерживаемые libx264

Ниже представлен полный список опций, которые поддерживает libx264:

Option	Type	Possible values	Description
preset	enum	ultrafast, superfast, veryfast, faster, fast, medium, slow, slower, veryslow	Encoding preset
tune	enum		Tune the encoding params
		film	Use for high quality movie content; lowers deblocking
		animation	Good for cartoons; uses higher deblocking and more reference frames
		grain	Preserves the grain structure in old, grainy film material
		stillimage	Good for slideshow-like content
		fastdecode	Allows faster decoding by disabling certain filters
		zerolatency	Good for fast encoding and low-latency streaming
		psnr	Improve PSNR metrics (mainly used for codec development)
		ssim	Improve SSIM metrics (mainly used for codec development)
profile	enum	main, high, high10, high422, high444	Encoding profile
level	enum		Specify level (as defined by Annex A)
		-1	auto
		1, 1.1, 1.2, 1.3, 2, 2.1, 2.2, 3, 3.1, 3.2, 4, 4.1, 4.2, 5 5.1	

wpredp	string		Weighted prediction for P-frames
a53cc	bool		Use A53 Closed Captions (if available)
x264opts	string		x264 options
crf	float		Select the quality for constant quality mode
crf_max	float		In CRF mode
qp	int		Constant quantization parameter rate control method
aq-mode	enum		Adaptive Quantization mode
		variance	Variance AQ
		autovariance	Auto-variance AQ
		autovariance-biased	Auto-variance AQ with bias to dark scenes
aq-strength	float		AQ strength. Reduces blocking and blurring in flat and textured areas
psy	bool		Use psychovisual optimizations
psy-rd	string		Strength of psychovisual optimization
rc-lookahead	int		Number of frames to look ahead for frametype and ratecontrol
weightb	bool		Weighted prediction for B-frames
weightp	enum	none, simple, smart, ssim	Weighted prediction analysis method
intra-refresh	bool		Use Periodic Intra Refresh instead of IDR frames
bluray-compat	bool		Bluray compatibility workarounds
b-bias	int		Influences how often B-frames are used
b-pyramid	enum		Keep some B-frames as references
		none	Do not use B-frames as reference
		strict	Strictly hierarchical pyramid
		normal	Non-strict (not Blu-ray compatible)
mixed-refs	bool		One reference per partition as opposed to one reference per macroblock
8x8dct	bool		High profile 8x8 transform
fast-pskip	bool		Enable fast decision to select between INTRA or INTER coding modes

aud	bool		Use access unit delimiters
mbtree	bool		Use macroblock tree ratecontrol
deblock	string		Loop filter parameters in <alpha:beta> form
cplxblur	float		Reduce fluctuations in QP (before curve compression)
partitions	enum	i4x4, i8x8, p4x4, p8x8, b8x8	A comma-separated list of partitions to consider. Minimal block size for I,P and B macroblocks
direct-pred	enum	auto, none, spatial, temporal	How previously evaluated motion vectors can be reused by encoder.
slice-max-size	int		Limit the size of each slice in bytes
avcintra-class	int		AVC-Intra class 50/100/200/300/480
motion-est	enum	dia, hex, umh, esa, tesa	Motion estimation method
forced-idr	bool		If forcing keyframes force them as IDR frames.
coder	enum		Entropy coding mode
		default	Default value
		auto	Defined by encoder
		cabac	Context-adaptive binary arithmetic coding.
		cavlc	Context-adaptive variable-length coding.
x264-params	enum	enum	Override the x264 configuration using a :-separated list of key=value parameters

3.5. Task Manager

3.5.1. Назначение сервиса Task Manager

Сервис TaskManager предназначен для:

- управления процессом обработки загруженных пользователями файлов путем чтения и передачи сообщений между другими сервисами (Uploader, Probe, Transcoder, Backend);
- реализации бизнес-логики, применение правил о том, как должны обрабатываться входящие видео файлы;
- уведомления сервиса Backend о статусе выполнения задач.

3.5.2. Переменные окружения Task Manager

В таблице ниже представлены переменные окружения, которые используются для настройки компонента Task Manager.

Переменная	Тип данных	Формат	Диапазон возможных значений	Обязательно	Описание
Блок базовых настроек					
LOG_LEVEL	Integer		1. 2. 3. 4. 5. 6. 7.	Нет	Уровень ведения журнала событий, в числовом формате. Допустимые значения: <ul style="list-style-type: none"> • -1 - трассировка; • 0 - отладка; • 1 - информация; • 2 - предупреждение; • 3 - ошибка; • 4 - неустраняемая ошибка; • 5 - тревога. Значение по умолчанию: 1.
CONNECTOR_TYPE	String		KAFKA	Нет	Брокер сообщение от которого Task Manager получает задачи. Используйте значение KAFKA. Значение по умолчанию: KAFKA.
DELIVERY_URI_PREFIX	String			Да	Префикс путей URI для подстановки в сообщение TASK_STATUS. Применяется для сопоставления внутренних путей к файлам во внешние ссылки. Значение по умолчанию: отсутствует.
Блок переменных, необходимых, если CONNECTOR_TYPE=KAFKA					
KAFKA_BROKERS	String	host или host:port, несколько значений разделяются		Да, если CONNECTOR_TYPE=KAFKA	Список брокеров Kafka в виде значений, разделённых запятыми, в формате <хост> или <хост>:<порт>. Значение по умолчанию: отсутствует.

		запятые			
		ми			
KAFKA_TOPIC_TASK_MANAGER	String			Да, если CONNECTOR_TYPE=KAFKA	<p>Имя топика Kafka, откуда Task Manager считывает сообщения компонентов Медиаплатформы (Uploader, Prober, Transcoder):</p> <ul style="list-style-type: none"> • Uploader: <ul style="list-style-type: none"> ○ начало загрузки исходного медиафайла во временное хранилище компонента Uploader; ○ завершение загрузки исходного медиафайла во временное хранилище компонента Uploader; ○ сбой загрузки исходного медиафайла во временное хранилище компонента Uploader; ○ завершение перемещения медиафайла в постоянное хранилище Медиаплатформы для анализа и обработки; ○ сбой перемещения медиафайла в постоянное хранилище Медиаплатформы для анализа и обработки. • Prober: <ul style="list-style-type: none"> ○ завершение анализа медиафайла;

					<ul style="list-style-type: none"> ○ сбой при анализе медиафайла. • Transcoder: <ul style="list-style-type: none"> ○ завершение создания эскиза (thumbnail) медиафайла; ○ сбой при создании эскиза (thumbnail) медиафайла; ○ завершение обработки медиафайла и/или извлечения аудиодорожки; ○ сбой обработки медиафайла и/или извлечения аудиодорожки. <p>Значение по умолчанию: KAFKA_TOPIC_TASK_MANAGER</p>
KAFKA_TOPIC_PROBE	String			Да, если CONNECTOR_TYPE=KAFKA	Имя топика Kafka, куда Task Manager ставит задачи на анализ загруженного медиафайла для компонента Probe. Значение по умолчанию: KAFKA_TOPIC_PROBE.
KAFKA_TOPIC_TRANSCODER	String			Да, если CONNECTOR_TYPE=KAFKA	Имя топика Kafka, куда Task Manager ставит для компонента Transcoder задачи на обработку медиафайла. Значение по умолчанию: KAFKA_TOPIC_TRANSCODER.
KAFKA_TOPIC_STATUS	String			Да, если CONNECTOR_TYPE=KAFKA	Имя топика Kafka, куда Task Manager отправляет сообщения о жизненном цикле загрузки и обработки медиафайла на основе данных, полученных от других компонентов Медиаплатформы: <ul style="list-style-type: none"> • начало загрузки исходного медиафайла; • завершение или сбой загрузки исходного медиафайла;

					<ul style="list-style-type: none"> • анализ (или сбой анализа) исходного медиафайла; • создание (или сбой создания) эскиза (thumbnail) медиафайла; • обработка (или сбой обработки) исходного медиафайла. <p>Топик <code>KAFKA_TOPIC_STATUS</code> доступен для чтения извне платформы. Значение по умолчанию: <code>KAFKA_TOPIC_STATUS</code>.</p>
KAFKA_TOPIC_TASKMANAGER_SETTINGS	String			Да, если <code>CONNECTOR_TYPE=KAFKA</code>	Топик Kafka, откуда Task Manager считывает профили обработки медиафайлов, которые использует для формирования задачи на обработку для компонента Transcoder. Значение по умолчанию: <code>KAFKA_TOPIC_TASKMANAGER_SETTINGS</code> .
KAFKA_GROUD_ID	Integer		0	Нет	В настоящее время не применяется. Всегда используйте значение 0.

Переменные среды указываются в файле или файлах YAML. Это может быть как единый файл для всех компонентов Медиаплатформы, так и несколько файлов.

В зависимости от вариантов развёртывания переменные можно указать вручную или включить в настройки из других источников.

Ниже представлен пример настроек компонента Task Manager.

env:

```
# Блок базовых настроек
- name: LOG_LEVEL
  value: "-1"
- name: DELIVERY_URI_PREFIX
  value: https://test.ru/
- name: CONNECTOR_TYPE
  value: "KAFKA"
```

```
# Блок настроек параметров шины сообщений
- name: KAFKA_BROKERS
  value: 127.0.0.1:9093
- name: KAFKA_TOPIC_TASK_MANAGER
  value: KAFKA_TOPIC_TASK_MANAGER
- name: KAFKA_TOPIC_PROBE
  value: KAFKA_TOPIC_PROBE
- name: KAFKA_TOPIC_TRANSCODER
  value: KAFKA_TOPIC_TRANSCODER
- name: KAFKA_TOPIC_STATUS
  value: KAFKA_TOPIC_STATUS
- name: KAFKA_TOPIC_TASKMANAGER_SETTINGS
  value: KAFKA_TOPIC_TASKMANAGER_SETTINGS
- name: KAFKA_GROUP_ID
  value: "0"
```

3.5.3. Сбор метрик

Отправителем метрик является модуль Task Manager, входящий в состав Медиаплатформы.

Метрики собираются с помощью системы `prom_taskmanager_connection_status` мониторинга и оповещений Prometheus и отображаются в виде графиков, счетчиков и в других представлениях с помощью инструмента для визуализации и анализа данных Grafana.

Отображаемый отрезок времени на всех графиках по умолчанию: 6 часов.

Все метрики выводятся для списка подов или конкретного пода.

3.5.3.1. Описание функциональных клавиш Grafana

- **namespace** – выбор окружения;
- **prober_instance** – поды Prober;
- **transcoder_instance** – поды Transcoder;
- **uploader_instance** – поды Uploader;
- **taskmanager_instance** – поды Task Manager.

Для всех функциональных клавиш, кроме namespace, значение по умолчанию all.

3.5.3.2. Описание метрик Grafana

При выборе большого временного интервала или частом перезапуске подов на графиках и счётчиках относительно него могут отображаться все запущенные поды и их состояние. Для мониторинга последних следует выбирать более короткий интервал. Данное поведение позволяет отслеживать состояние конкретного пода в конкретный момент времени.

Разрывы на графиках и отсутствие информации возможны в случае, если после старта сервиса метрика ни разу не отсылалась или сервис не был запущен или не работает в следствии системной ошибки.

Метрики, которые регистрируют единичные всплески, могут быть не видны на графиках, где для его построения используется функции `rate` или `increase`, которые возвращают скорость изменения метрики на заданном интервале, при этом первая делит полученное значение на этот интервал. Для регистрации и отслеживания подобных изменений следует использовать счётчики.

	Наименование дашборда	Наименование метрики	Тип метрики	Описание метрики	Параметры сбора данных	Серии данных
1	Task manager	Количество ошибок (колво/с)	График	Средняя экстраполированная скорость изменения количества ошибок при обработке сообщений за период.	-	Метрика: <code>mp_taskmanager_errors_total</code> <ul style="list-style-type: none"> Total: <code>sum(rate(mp_taskmanager_errors_total{namespace=~"\$namespace", pod=~"\$taskmanager_instance"}[\$_rate_interval]))</code>
2	Task manager	Общее количество ошибок после старта	Счетчик	Количество ошибок при обработке сообщений	-	Метрика: <code>mp_taskmanager_errors_total</code> <ul style="list-style-type: none"> TOTAL: <code>sum(mp_taskmanager_errors_total{namespace=~"\$namespace", pod=~"\$taskmanager_instance"}) >=0 or on() vector(0)</code>

				ий накопле нное значение за время работы сервиса с момента последн ей перезагр узки.		
3	Overa II	TaskManager - статус подключения	Графи к	Статус связи с сервиса ми: <ul style="list-style-type: none"> • 1 - с о е д и н е н и е п р и с у т с т в у е т ; • 0 - с о е 	<ul style="list-style-type: none"> • exported_ser vice: <ul style="list-style-type: none"> ○ Kaf ka ○ Pos tgre s 	Метрика: mp_taskmanager_connection_status <ul style="list-style-type: none"> • {{exported_service}}: mp_taskmanager_connection_status{ exported_service="Kafka", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} • {{exported_service}}: mp_taskmanager_connection_status{ exported_service="Postgres", namespace=~"\$namespace", pod=~"\$taskmanager_instance"}

				Д и н е н и е о т с у т с т в у е т .		
				Все графики суммиру ются, каждый следую щий будет выгляде ть, как своё значение плюс предыду щее.		
4	Task manag er	Количество прочитанных сообщений из Kafka по типу (кол- во/с)	Графи к	Количес тво прочита нных сообщен ий (успешн ых и не успешн ых) по типу сообщен ия.	<ul style="list-style-type: none"> • PROBE_TA SK_FINISH ED • THUMBNA IL_INFO • TRANSCO DER_TASK _FINISHED • TRANSFER _TASK_FIN ISHED • TRANSFER _TASK_ST ART 	Метрика: mp_taskmanager_kafka_read_message_types <ul style="list-style-type: none"> • {{message_type}}: sum(rate(mp_taskmanager_kafka_re ad_message_types{namespace=~"\$n amespace", pod=~"\$taskmanager_instance"}[\$_ _rate_interval])) by (message_type)

					<ul style="list-style-type: none"> • UPLOADER_TASK_CREATE 	
5	Task manager	Кол-во прочитанных сообщений из Kafka по типу после старта	Счетчик	Количество прочитанных сообщений (успешных и неуспешных) по типу сообщения.	<ul style="list-style-type: none"> • message_type <ul style="list-style-type: none"> ○ PROBE_TASK_FINISHED ○ THUMBNAIL_INFO ○ TRANSCODER_TASK_FINISHED ○ TRANSFER_TASK_FINISHED ○ UPLOADER_TASK_CREATE ○ TRANSFER_TASK_START 	Метрика: mp_taskmanager_kafka_read_message_types <ul style="list-style-type: none"> • TRANSFER_TASK_START: sum(mp_taskmanager_kafka_read_message_types{message_type="TRANSFER_TASK_START", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} >= 0 or on() vector(0)) by (message_type) • TRANSFER_TASK_FINISHED: sum(mp_taskmanager_kafka_read_message_types{message_type="TRANSFER_TASK_FINISHED", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} >= 0 or on() vector(0)) by (message_type) • UPLOADER_TASK_CREATE: sum(mp_taskmanager_kafka_read_message_types{message_type="UPLOADER_TASK_CREATE", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} >= 0 or on() vector(0)) by (message_type) • PROBE_TASK_FINISHED: sum(mp_taskmanager_kafka_read_message_types{message_type="PROBE_TASK_FINISHED", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} >= 0 or on() vector(0)) by (message_type) • TRANSCODER_TASK_FINISHED: sum(mp_taskmanager_kafka_read_message_types{message_type="TRANSCODER_TASK_FINISHED", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} >= 0 or on() vector(0)) by (message_type) • THUMBNAIL_INFO: sum(mp_taskmanager_kafka_read_message_types{message_type="TH

					<ul style="list-style-type: none"> ST AR T o UP LO AD ER _T AS K_ CR EA TE 	<p>UMBNAIL_INFO", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} >= 0 or on() vector(0) by (message_type)</p> <ul style="list-style-type: none"> • TOTAL - SUCCESS: sum(mp_taskmanager_kafka_read_messages{status="success", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} >=0 or on() vector(0)) • TOTAL - FAILED: sum(mp_taskmanager_kafka_read_messages{status="failed", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} >=0 or on() vector(0))
6	Task manager	Кол-во прочитанных и отправленных сообщений в Kafka (кол-во/с)	График	Количество прочитанных (успешных и неуспешных) и отправленных сообщений в Kafka, суммарно и для каждого топика отдельно.	<ul style="list-style-type: none"> • topic • status: <ul style="list-style-type: none"> o success o failed 	<p>Метрики: mp_taskmanager_kafka_read_messages, mp_taskmanager_kafka_writed_messages</p> <ul style="list-style-type: none"> • Read - success: sum(rate(mp_taskmanager_kafka_read_messages{status="success", namespace=~"\$namespace", pod=~"\$taskmanager_instance"}[\$_rate_interval])) >=0 or on() vector(0) • Read - failed: sum(rate(mp_taskmanager_kafka_read_messages{status="failed", namespace=~"\$namespace", pod=~"\$taskmanager_instance"}[\$_rate_interval])) >=0 or on() vector(0) • Writed - {{status}} - {{topic}}: sum(rate(mp_taskmanager_kafka_writed_messages{namespace=~"\$namespace", pod=~"\$taskmanager_instance"}[\$_rate_interval])) by (status,topic)
7	Task manager	Кол-во прочитанных и отправленных сообщений в Kafka после старта	Счетчик/Таблица	Количество прочитанных (успешных и неуспешных) и	<ul style="list-style-type: none"> • topic • status: <ul style="list-style-type: none"> o success o failed 	<p>Метрики: mp_taskmanager_kafka_writed_messages</p> <ul style="list-style-type: none"> • {{topic}} - {{status}}: sum(mp_taskmanager_kafka_writed_messages{namespace=~"\$namespace", pod=~"\$taskmanager_instance"}) by (topic, status)

				отправленных сообщений в Kafka с момента последней перезагрузки.		
8	Task manager	Среднее время выполнения запроса к БД (с)	График	Среднее время выполнения sql-запроса к БД.	<ul style="list-style-type: none"> status: <ul style="list-style-type: none"> success failed 	Метрика: mp_taskmanager_pg_response_time <ul style="list-style-type: none"> Success: <pre>(sum(rate(mp_taskmanager_pg_response_time_sum{status="success", namespace=~"\$namespace", pod=~"\$taskmanager_instance"}[\$_rate_interval])) / sum(rate(mp_taskmanager_pg_response_time_count{status="success", namespace=~"\$namespace", pod=~"\$taskmanager_instance"}[\$_rate_interval]))) >=0 or on() vector(0)</pre> Failed: <pre>(sum(rate(mp_taskmanager_pg_response_time_sum{status="failed", namespace=~"\$namespace", pod=~"\$taskmanager_instance"}[\$_rate_interval])) / sum(rate(mp_taskmanager_pg_response_time_count{status="failed", namespace=~"\$namespace", pod=~"\$taskmanager_instance"}[\$_rate_interval]))) >=0 or on() vector(0)</pre>
9	Task manager	Время выполнения запроса к БД удаления данных (гистограмма) (кол-во)	Гистограмма	Время выполнения успешно удаленного sql-запроса к БД удаления данных.	<ul style="list-style-type: none"> le query_type: <ul style="list-style-type: none"> delete status: <ul style="list-style-type: none"> failed 	Метрика: mp_taskmanager_pg_response_time_bucket <ul style="list-style-type: none"> {{le}}: sum by (le) <pre>(increase(mp_taskmanager_pg_response_time_bucket{query_type="delete", status="failed", le=~"[0-9].+", namespace=~"\$namespace", pod=~"\$taskmanager_instance"}[\$_rate_interval]))</pre>

1 0	Task manager	Время выполнения запроса к БД добавления данных (гистограмма) (кол-во)	Гистограмма	Время выполнения успешного sql-запроса к БД добавления данных.	<ul style="list-style-type: none"> le query_type: <ul style="list-style-type: none"> insert status: <ul style="list-style-type: none"> success 	Метрика: mp_taskmanager_pg_response_time_bucket <ul style="list-style-type: none"> {{le}}: sum by (le) (increase(mp_taskmanager_pg_response_time_bucket{query_type="insert", status="success", le=~"[0-9].+", namespace=~"\$namespace", pod=~"\$taskmanager_instance"}[\$_rate_interval]))
1 1	Task manager	Время выполнения запроса к БД выбора данных (гистограмма) (кол-во)	Гистограмма	Время выполнения успешного sql-запроса к БД выбора данных.	<ul style="list-style-type: none"> le query_type: <ul style="list-style-type: none"> select status: <ul style="list-style-type: none"> success 	Метрика: mp_taskmanager_pg_response_time_bucket <ul style="list-style-type: none"> {{le}}: sum by (le) (increase(mp_taskmanager_pg_response_time_bucket{query_type="select", status="success", le=~"[0-9].+", namespace=~"\$namespace", pod=~"\$taskmanager_instance"}[\$_rate_interval]))
1 2	Task manager	Время выполнения запроса к БД обновления данных (гистограмма) (кол-во)	Гистограмма	Время выполнения успешного sql-запроса к БД обновления данных.	<ul style="list-style-type: none"> le query_type: <ul style="list-style-type: none"> update status: <ul style="list-style-type: none"> success 	Метрика: mp_taskmanager_pg_response_time_bucket <ul style="list-style-type: none"> {{le}}: sum by (le) (increase(mp_taskmanager_pg_response_time_bucket{query_type="update", status="success", le=~"[0-9].+", namespace=~"\$namespace", pod=~"\$taskmanager_instance"}[\$_rate_interval]))
1 3	Task manager	Задачи отклоненные по ограничениям (кол-во/с)	График	Количество отклоненных задач модулем Task Manager (количество медиафайлов, отклоненных по ограничениям).	-	Метрика: mp_taskmanager_failed_tasks <ul style="list-style-type: none"> Total: sum(rate(mp_taskmanager_failed_tasks{namespace=~"\$namespace", pod=~"\$taskmanager_instance"}[\$_rate_interval])) >=0 or on() vector(0)

1 4	Task manager	Полученные задачи на транскодирование	Измеритель	Текущее количество успешно полученных задач на транскодирование по типу обработки и количество отклоненных задач модулем Task Manager (количество медиафайлов, отклоненных по ограничениям).	<ul style="list-style-type: none"> status: <ul style="list-style-type: none"> arrived transcoder_type: <ul style="list-style-type: none"> any gpu cpu 	Метрика: mp_taskmanager_tasks <ul style="list-style-type: none"> {{status}} - {{transcoder_type}}: mp_taskmanager_tasks{status="arrived", transcoder_type="any", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} {{status}} - {{transcoder_type}}: mp_taskmanager_tasks{status="arrived", transcoder_type="gpu", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} {{status}} - {{transcoder_type}}: mp_taskmanager_tasks{status="arrived", transcoder_type="cpu", namespace=~"\$namespace", pod=~"\$taskmanager_instance"}
1 5	Task manager	Задачи на транскодирование в процессе обработки	Измеритель	Текущее количество задач на транскодирование в процессе обработки по типу обработки.	<ul style="list-style-type: none"> status: <ul style="list-style-type: none"> processing transcoder_type: <ul style="list-style-type: none"> any gpu cpu 	Метрика: mp_taskmanager_tasks <ul style="list-style-type: none"> {{status}} - {{transcoder_type}}: mp_taskmanager_tasks{status="processing", transcoder_type="any", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} {{status}} - {{transcoder_type}}: mp_taskmanager_tasks{status="processing", transcoder_type="gpu", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} {{status}} - {{transcoder_type}}: mp_taskmanager_tasks{status="processing", transcoder_type="cpu", namespace=~"\$namespace", pod=~"\$taskmanager_instance"}
1 6	Task manager	Задачи на транскодирование в	Измеритель	Текущее количество	<ul style="list-style-type: none"> status: 	Метрика: mp_taskmanager_tasks <ul style="list-style-type: none"> {{status}} - {{transcoder_type}}: mp_taskmanager_tasks{status="finis

		статусе завершения		задач на транскодирование в статусе завершения по типу обработки.	<ul style="list-style-type: none"> ○ finished • transcoder_type: <ul style="list-style-type: none"> ○ any ○ gpu ○ cpu 	<pre>hed", transcoder_type="any", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} • {{status}} - {{transcoder_type}}: mp_taskmanager_tasks{status="finished", transcoder_type="gpu", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} • {{status}} - {{transcoder_type}}: mp_taskmanager_tasks{status="finished", transcoder_type="cpu", namespace=~"\$namespace", pod=~"\$taskmanager_instance"}</pre>
17	Task manager	Время ожидания обработки успешно полученных задач на транскодирование (гистограмма) (кол-во)	Гистограмма	Время ожидания обработки успешно полученных задач на транскодирование.	<ul style="list-style-type: none"> • le • status: <ul style="list-style-type: none"> ○ arrived 	<p>Метрика:</p> <pre>p_taskmanager_task_waiting_time_seconds_bucket • {{le}}: sum by (le) (increase(p_taskmanager_task_waiting_time_seconds_bucket{status="arrived", le=~"[0-9].+", namespace=~"\$namespace", pod=~"\$taskmanager_instance"}[\$__rate_interval])) >= 0 or on() vector(0)</pre>
18	Task manager	Время ожидания выполнения задач на транскодирование (гистограмма) (кол-во)	Гистограмма	Время ожидания выполнения задач на транскодирование.	<ul style="list-style-type: none"> • le • state: <ul style="list-style-type: none"> ○ processing 	<p>Метрика:</p> <pre>p_taskmanager_task_waiting_time_seconds_bucket • {{le}}: sum by (le) (increase(p_taskmanager_task_waiting_time_seconds_bucket{status="processing", le=~"[0-9].+", namespace=~"\$namespace", pod=~"\$taskmanager_instance"}[\$__rate_interval])) >= 0 or on() vector(0)</pre>
19	Task manager	Время ожидания завершения задач на транскодирование (гистограмма) (кол-во)	Гистограмма	Время ожидания завершения задач на транскодирование.	<ul style="list-style-type: none"> • le • state: <ul style="list-style-type: none"> ○ finished 	<p>Метрика:</p> <pre>p_taskmanager_task_waiting_time_seconds_bucket • {{le}}: sum by (le) (increase(p_taskmanager_task_waiting_time_seconds_bucket{status="finished", le=~"[0-9].+", namespace=~"\$namespace", pod=~"\$taskmanager_instance"}[\$__rate_interval])) >= 0 or on() vector(0)</pre>

20	Task manager	Завершенные задачи на транскодирование	Измеритель	Количество транскодированных медиафайлов по типу транскодера и статусу.	<ul style="list-style-type: none"> • status <ul style="list-style-type: none"> ○ success ○ failed • transcoder_type <ul style="list-style-type: none"> ○ any ○ gpu ○ cpu 	<p>Метрика: mp_taskmanager_result</p> <ul style="list-style-type: none"> • ok - any: mp_taskmanager_result{status="success", transcoder_type="any", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} >=0 or on() vector(0) • ok - gpu: mp_taskmanager_result{status="success", transcoder_type="gpu", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} >=0 or on() vector(0) • ok - cpu: mp_taskmanager_result{status="success", transcoder_type="cpu", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} >=0 or on() vector(0) • failed - any: mp_taskmanager_result{status="failed", transcoder_type="any", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} >=0 or on() vector(0) • failed - gpu: mp_taskmanager_result{status="failed", transcoder_type="gpu", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} >=0 or on() vector(0) • failed - cpu: mp_taskmanager_result{status="failed", transcoder_type="cpu", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} >=0 or on() vector(0)
21	Overalls	TaskManager - общее количество ошибок по текстовым кодам после старта	Счетчик / Таблица	Счетчик и ошибок по внутренним кодам ошибок ([int Ru por] release 2.1.4)	<ul style="list-style-type: none"> • code • text_code 	<p>Метрика: mp_taskmanager_errors - code, code_text</p> <ul style="list-style-type: none"> • {{code_text}}: sum(mp_taskmanager_errors{code!="0", namespace=~"\$namespace", pod=~"\$taskmanager_instance"}) by (code, code_text)

				<p>Сообщения об ошибках Медиаплатформы) с маппингом кодов на текстовые лейблы.</p>		
2 2	Overa II	TaskManager - общее количество ошибок по текстовым кодам	Измеритель	<p>Счетчик и ошибок по внутренним кодам ошибок с маппингом кодов на текстовые лейблы.</p>	<ul style="list-style-type: none"> • text_code 	<p>Метрика: mp_taskmanager_errors - code, code_text</p> <ul style="list-style-type: none"> • {{code_text}}: sum(rate(mp_taskmanager_errors{code!="0", namespace=~"\$namespace", pod=~"\$taskmanager_instance"}[\$__rate_interval])) by (code_text)
2 3	Overa II	TaskManager - статус подключения к сервисам	Измеритель	<p>Статус связи с сервисами:</p> <ul style="list-style-type: none"> • 1 - состояние инициализации сервиса 	<ul style="list-style-type: none"> • exported_service: <ul style="list-style-type: none"> ○ Kafka ○ Postgres 	<p>Метрика: mp_taskmanager_connection_status</p> <ul style="list-style-type: none"> • kafka: mp_taskmanager_connection_status{exported_service="Kafka", namespace=~"\$namespace", pod=~"\$taskmanager_instance"} • postgres: mp_taskmanager_connection_status{exported_service="Postgres", namespace=~"\$namespace", pod=~"\$taskmanager_instance"}

				у г с т в у е т ; • 0 - с о е д и н е н и е о т с у т с т в у е т .		
--	--	--	--	---	--	--

3.5.3.3. Описание метрик Prometheus

Ниже приведена таблица с описанием метрик Prometheus для модуля Task Manager.

- **histigram** - графическое представление данных в виде графика, слева, по оси Y, кол-во значений, попавших в бакет, справа, легенда (значения le) и размерность. Граничные значения (le) в гистограмме - это диапазоны значений, на которые разбиваются измерения. Гистограмма с граничными значениями позволяет нам увидеть распределение значений внутри каждого диапазона и оценить, насколько часто значения попадают в каждый из диапазонов. Если один из графиков начинает стремиться к нулю, значит значения в него не попадают.

- **gauge** - масштабируемый диалоговый индикатор, который отображает текущее значение метрики в процентах или числах. Большим преимуществом Gauge в Grafana является то, что он может легко приспосабливаться к разным значениям, чтобы обеспечить оптимальное отображение информации на дашборде: может быть использован для отображения любых числовых данных, таких как счётчики, процентные доли, абсолютные значения и т.д., т.к. gauge измеряет числовое значение в определенный момент времени, а не считает скорость изменения значений, как Counter. Gauge в Grafana можно отображать не только в виде графика, но и в виде других типов визуализации, таких как текст, таблицы или специально созданные плагины.
- **counter** - счетчик, который увеличивается на каждое новое событие. Счетчики обычно могут только увеличиваться, но не уменьшаться, и сброс в ноль может потребоваться вручную.
- **summary** - собирает информацию о распределении значений заданной величины. Это позволяет собирать статистику о количестве значений, их среднем, медиане, процентилях и т.д. Для каждого Summary Prometheus собирает образцы (samples) для значений, которые необходимо агрегировать. Для каждого образца указывается значение и метка времени, когда оно было собрано. Затем Prometheus вычисляет ряд статистических значений на основе этих образцов, включая сумму всех значений, количество собранных образцов и среднее значение. В Grafana представление Summary может использоваться для создания графиков и диаграмм, показывающих распределение значений величины. Графики Summary могут быть полезны для анализа производительности приложения, когда необходимо определить, какое количество запросов было выполнено за определенный период времени, и как быстро были выполнены запросы.

	Наименование метрики	Тип	Метки	Описание	Событие
1	mp_taskmanager_connection_status	gauge	<ul style="list-style-type: none"> • service: <ul style="list-style-type: none"> ○ Kafka ○ Postgres 	Статус связи с сервисами. <ul style="list-style-type: none"> • 1 - соединение присутствует; 	Изменилось состояние связи с сервисами.

				<ul style="list-style-type: none"> 0 - соединение отсутствует. 	
2	mp_taskmanager_errors	counter	<ul style="list-style-type: none"> code code_text 	<p>Счетчики ошибок по внутренним кодам ошибок ([int Rupor] release 2.1.4 Сообщения об ошибках Медиaplatformы)) с маппингом кодов на текстовые лейблы.</p> <ul style="list-style-type: none"> code - код ошибки; code_text - текст ошибки. 	<p>Появилась новая ошибка Медиaplatformы, обработчик которой существует в Медиaplatformе.</p>
3	mp_taskmanager_errors_total	counter	-	<p>Общее количество ошибок по статусам заданий модуля Task Manager.</p>	<p>Появилась новая ошибка по статусу задания модуля Task Manager.</p>
4	mp_taskmanager_failed_tasks	counter	-	<p>Количество отклоненных заданий модулем Task_Manager (количество медиафайлов, отклоненных по ограничениям).</p>	<p>Файл отклонен по ограничениям.</p>
5	mp_taskmanager_kafka_read_message_types	counter	<ul style="list-style-type: none"> message_type: 	<p>Количество прочитанных</p>	<p>Зафиксирована попытка</p>

			<ul style="list-style-type: none"> ○ PROBE_TASK_FINISHED ○ THUMBNAIL_INFO ○ TRANSCODER_TASK_FINISHED ○ TRANSFER_TASK_FINISHED ○ TRANSFER_TASK_START ○ UPLOADER_TASK_CREATE 	сообщений (успешных и неуспешных) по типу сообщения.	чтения сообщения из Kafka.
6	mp_taskmanager_kafka_read_messages	counter	<ul style="list-style-type: none"> • topic: <ul style="list-style-type: none"> ○ taskmanager • status: <ul style="list-style-type: none"> ○ failed ○ success 	Общее количество успешных и неуспешных прочитанных сообщений.	Зафиксирована попытка чтения сообщения из Kafka.
7	mp_taskmanager_kafka_wrote_messages	counter	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ success ○ failed • topic: <ul style="list-style-type: none"> ○ prober.tasks ○ status ○ transcoder.tasks.cpu ○ transcoder.tasks.gpu 	Количество отправленных сообщений в каждый топик. <ul style="list-style-type: none"> • success - успешно; • failed - неуспешно. 	Зафиксирована попытка отправки сообщения в Kafka.
8	mp_taskmanager_pg_response_time	histogram	<ul style="list-style-type: none"> • query_type: <ul style="list-style-type: none"> ○ delete ○ insert ○ select ○ update • status: <ul style="list-style-type: none"> ○ failed ○ success • le: <ul style="list-style-type: none"> ○ 0.01 ○ 0.05 ○ 0.1 ○ 0.5 ○ 1 ○ 5 ○ 10 ○ +Inf 	Время выполнения запросов в БД по типу и статусу (в секундах). <ul style="list-style-type: none"> • sum - сумма значений; • count - общее количество значений; 	Зафиксирована попытка отправки запроса в БД.

			<ul style="list-style-type: none"> • sum • count • bucket 	<ul style="list-style-type: none"> • bucket - счетчик гистограммы (интервал), куда записывается значение; • + inf - + бесконечность; • success - успешно; • failed - неуспешно. • le - граничащие значения, указывающие на то, насколько увеличится показатель гистограммы. 	
9	mp_taskmanager_tasks	gauge	<ul style="list-style-type: none"> • status: <ul style="list-style-type: none"> ○ arrived ○ finished ○ transcoding • transcoder_type: <ul style="list-style-type: none"> ○ any ○ cpu ○ gpu 	Текущее количество задач в работе по статусу. <ul style="list-style-type: none"> • arrived - заданное 	<ul style="list-style-type: none"> • Задание получено • Задание завершено

				<ul style="list-style-type: none"> получено; finished - задание завершено (эта метрика в норме должна стремиться к нулю); transcoding-задание в процессе обработки. <p>Текущее количество задач в работе по типам:</p> <ul style="list-style-type: none"> обработка в гри или сри; обработка в гри; обработка в сри. 	<ul style="list-style-type: none"> решено. Задание в процессе обработки.
10	mp_taskmanager_task_waiting_time_seconds	histogram	<ul style="list-style-type: none"> status: <ul style="list-style-type: none"> arrived finished processing le: <ul style="list-style-type: none"> 1 2.4839607111043738 6.170060814310146 	<p>Время ожидания обработки и выполнения задачи по статусу.</p> <ul style="list-style-type: none"> arrived - задание 	<ul style="list-style-type: none"> задание получено; задание завершено;

			<ul style="list-style-type: none"> ○ 15.326188647871062 ○ 38.06965045228558 ○ 94.56351600895424 ○ 234.89205847013176 ○ 583.4626445902387 ○ 1449.2982855592077 ○ 3599.99999999999 ○ +Inf <ul style="list-style-type: none"> • sum • count • bucket 	<p>получено;</p> <ul style="list-style-type: none"> • finished - задание завершено (эта метрика в норме должна стремиться к нулю); • processing - задание в процессе обработки; • sum - сумма значений; • count - общее количество значений; • le - граничащие значения, указывающие на то, насколько увеличится показатель 	<ul style="list-style-type: none"> • задание в процессе обработки.
--	--	--	--	--	---

				гистог раммы; <ul style="list-style-type: none"> • bucket - счетчик гистог раммы (интервал), куда записывается значение; 	
1 1	mp_taskmanager_result	gauge	<ul style="list-style-type: none"> • transcoder_type: <ul style="list-style-type: none"> ○ any ○ cpu ○ gpu • status: <ul style="list-style-type: none"> ○ success ○ failed 	Количество транскодированных медиафайлов по типу транскодера и статусу.	Завершено задание по транскодированию медиафайла.

3.5.3.4. Пример ответа микросервиса как экспортера Prometheus

URI для запроса метрик относительно значения переменной окружения

TELEMETRY_METRICS_ADDRESS:

- /metrics - метрики;
- /health - так называемый health-статус сервиса, т.е. признак того, что сервис жив и не "подвис".

```
# HELP mp_taskmanager_kafka_writed_messages connection status. metric value: 0 - disconnected,
1 - connected
# TYPE mp_taskmanager_connection_status gauge
mp_taskmanager_connection_status{service="Kafka"} 1
mp_taskmanager_connection_status{service="Postgres"} 1
# HELP mp_taskmanager_errors errors count
# TYPE mp_taskmanager_errors counter
mp_taskmanager_errors{code="1020201",code_text="internal_error"} 0
mp_taskmanager_errors{code="1020202",code_text="duration_exceeded"} 0
mp_taskmanager_errors{code="1020203",code_text="resolution_more_than_maximum"} 0
```

```

mp_taskmanager_errors{code="1020204",code_text="resolution_less_than_minumum"} 0
mp_taskmanager_errors{code="1020205",code_text="container_cannot_be_processed"} 0
mp_taskmanager_errors{code="1020206",code_text="codec_cannot_be_processed"} 0
mp_taskmanager_errors{code="1020207",code_text="video_bitrate_less_than_minimum"} 0
mp_taskmanager_errors{code="1020208",code_text="audio_bitrate_less_than_minimum"} 0
mp_taskmanager_errors{code="1020209",code_text="video_bitrate_more_than_maximum"} 0
mp_taskmanager_errors{code="1020210",code_text="audio_bitrate_more_than_maximum"} 0
# HELP mp_taskmanager_errors_total total errors count
# TYPE mp_taskmanager_errors_total counter
mp_taskmanager_errors_total 2
# HELP mp_taskmanager_failed_tasks tasks failed by TaskManager
# TYPE mp_taskmanager_failed_tasks counter
mp_taskmanager_failed_tasks 0
# HELP mp_taskmanager_kafka_read_message_types message types count read from Kafka
# TYPE mp_taskmanager_kafka_read_message_types counter
mp_taskmanager_kafka_read_message_types{message_type="PROBE_TASK_FINISHED"} 2
mp_taskmanager_kafka_read_message_types{message_type="TRANSFER_TASK_FINISHED"} 2
mp_taskmanager_kafka_read_message_types{message_type="TRANSFER_TASK_START"} 2
mp_taskmanager_kafka_read_message_types{message_type="UPLOADER_TASK_CREATE"} 4
# HELP mp_taskmanager_kafka_read_messages message count read from Kafka
# TYPE mp_taskmanager_kafka_read_messages counter
mp_taskmanager_kafka_read_messages{status="success",topic="taskmanager"} 10
# HELP mp_taskmanager_kafka_writed_messages message written to Kafka
# TYPE mp_taskmanager_kafka_writed_messages counter
mp_taskmanager_kafka_writed_messages{status="success",topic="prober.tasks"} 2
mp_taskmanager_kafka_writed_messages{status="success",topic="status"} 6
# HELP mp_taskmanager_pg_response_time postgres response time (seconds)
# TYPE mp_taskmanager_pg_response_time histogram
mp_taskmanager_pg_response_time_bucket{query_type="delete",status="failed",le="0.01"} 0
mp_taskmanager_pg_response_time_bucket{query_type="delete",status="failed",le="0.05"} 2
mp_taskmanager_pg_response_time_bucket{query_type="delete",status="failed",le="0.1"} 2
mp_taskmanager_pg_response_time_bucket{query_type="delete",status="failed",le="0.5"} 2

```

mp_taskmanager_pg_response_time_bucket{query_type="delete",status="failed",le="1"} 2
mp_taskmanager_pg_response_time_bucket{query_type="delete",status="failed",le="5"} 2
mp_taskmanager_pg_response_time_bucket{query_type="delete",status="failed",le="10"} 2
mp_taskmanager_pg_response_time_bucket{query_type="delete",status="failed",le="+Inf"} 2
mp_taskmanager_pg_response_time_sum{query_type="delete",status="failed"} 0.086931625
mp_taskmanager_pg_response_time_count{query_type="delete",status="failed"} 2
mp_taskmanager_pg_response_time_bucket{query_type="insert",status="success",le="0.01"} 0
mp_taskmanager_pg_response_time_bucket{query_type="insert",status="success",le="0.05"} 1
mp_taskmanager_pg_response_time_bucket{query_type="insert",status="success",le="0.1"} 2
mp_taskmanager_pg_response_time_bucket{query_type="insert",status="success",le="0.5"} 2
mp_taskmanager_pg_response_time_bucket{query_type="insert",status="success",le="1"} 2
mp_taskmanager_pg_response_time_bucket{query_type="insert",status="success",le="5"} 2
mp_taskmanager_pg_response_time_bucket{query_type="insert",status="success",le="10"} 2
mp_taskmanager_pg_response_time_bucket{query_type="insert",status="success",le="+Inf"} 2
mp_taskmanager_pg_response_time_sum{query_type="insert",status="success"}
0.10532016599999999
mp_taskmanager_pg_response_time_count{query_type="insert",status="success"} 2
mp_taskmanager_pg_response_time_bucket{query_type="select",status="success",le="0.01"} 0
mp_taskmanager_pg_response_time_bucket{query_type="select",status="success",le="0.05"} 2
mp_taskmanager_pg_response_time_bucket{query_type="select",status="success",le="0.1"} 2
mp_taskmanager_pg_response_time_bucket{query_type="select",status="success",le="0.5"} 2
mp_taskmanager_pg_response_time_bucket{query_type="select",status="success",le="1"} 2
mp_taskmanager_pg_response_time_bucket{query_type="select",status="success",le="5"} 2
mp_taskmanager_pg_response_time_bucket{query_type="select",status="success",le="10"} 2
mp_taskmanager_pg_response_time_bucket{query_type="select",status="success",le="+Inf"} 2
mp_taskmanager_pg_response_time_sum{query_type="select",status="success"}
0.08152604199999999
mp_taskmanager_pg_response_time_count{query_type="select",status="success"} 2
mp_taskmanager_pg_response_time_bucket{query_type="update",status="success",le="0.01"} 0
mp_taskmanager_pg_response_time_bucket{query_type="update",status="success",le="0.05"} 4
mp_taskmanager_pg_response_time_bucket{query_type="update",status="success",le="0.1"} 6
mp_taskmanager_pg_response_time_bucket{query_type="update",status="success",le="0.5"} 6

```

mp_taskmanager_pg_response_time_bucket{query_type="update",status="success",le="1"} 6
mp_taskmanager_pg_response_time_bucket{query_type="update",status="success",le="5"} 6
mp_taskmanager_pg_response_time_bucket{query_type="update",status="success",le="10"} 6
mp_taskmanager_pg_response_time_bucket{query_type="update",status="success",le="+Inf"} 6
mp_taskmanager_pg_response_time_sum{query_type="update",status="success"}
0.28859274999999995
mp_taskmanager_pg_response_time_count{query_type="update",status="success"} 6
# HELP mp_taskmanager_tasks current processing tasks
# TYPE mp_taskmanager_tasks gauge
mp_taskmanager_tasks{status="arrived",transcoder_type="any"} 0
mp_taskmanager_tasks{status="arrived",transcoder_type="cpu"} 0
mp_taskmanager_tasks{status="arrived",transcoder_type="gpu"} 0
mp_taskmanager_tasks{status="finished",transcoder_type="any"} 0
mp_taskmanager_tasks{status="finished",transcoder_type="cpu"} 0
mp_taskmanager_tasks{status="finished",transcoder_type="gpu"} 0
mp_taskmanager_tasks{status="processing",transcoder_type="any"} 0
mp_taskmanager_tasks{status="processing",transcoder_type="cpu"} 0
mp_taskmanager_tasks{status="processing",transcoder_type="gpu"} 0
# HELP mp_taskmanager_task_waiting_time_seconds tasks status time (seconds)
# TYPE mp_taskmanager_task_waiting_time_seconds histogram
p_taskmanager_task_waiting_time_seconds_bucket{status="arrived",le="1"} 2
p_taskmanager_task_waiting_time_seconds_bucket{status="arrived",le="2.4839607111043738"} 2
p_taskmanager_task_waiting_time_seconds_bucket{status="arrived",le="6.170060814310146"} 3
p_taskmanager_task_waiting_time_seconds_bucket{status="arrived",le="15.326188647871062"} 3
p_taskmanager_task_waiting_time_seconds_bucket{status="arrived",le="38.06965045228558"} 3
p_taskmanager_task_waiting_time_seconds_bucket{status="arrived",le="94.56351600895424"} 4
p_taskmanager_task_waiting_time_seconds_bucket{status="arrived",le="234.89205847013176"} 4
p_taskmanager_task_waiting_time_seconds_bucket{status="arrived",le="583.4626445902387"} 4
p_taskmanager_task_waiting_time_seconds_bucket{status="arrived",le="1449.2982855592077"} 4
p_taskmanager_task_waiting_time_seconds_bucket{status="arrived",le="3599.9999999999999"} 4
p_taskmanager_task_waiting_time_seconds_bucket{status="arrived",le="+Inf"} 4
p_taskmanager_task_waiting_time_seconds_sum{status="arrived"} 49.20234175

```

p_taskmanager_task_waiting_time_seconds_count{status="arrived"} 4
p_taskmanager_task_waiting_time_seconds_bucket{status="finished",le="1"} 0
p_taskmanager_task_waiting_time_seconds_bucket{status="finished",le="2.4839607111043738"}
0 p_taskmanager_task_waiting_time_seconds_bucket{status="finished",le="6.170060814310146"}
0
p_taskmanager_task_waiting_time_seconds_bucket{status="finished",le="15.326188647871062"}
0 p_taskmanager_task_waiting_time_seconds_bucket{status="finished",le="38.06965045228558"}
0 p_taskmanager_task_waiting_time_seconds_bucket{status="finished",le="94.56351600895424"}
0
p_taskmanager_task_waiting_time_seconds_bucket{status="finished",le="234.89205847013176"}
0 p_taskmanager_task_waiting_time_seconds_bucket{status="finished",le="583.4626445902387"}
2
p_taskmanager_task_waiting_time_seconds_bucket{status="finished",le="1449.2982855592077"}
2 p_taskmanager_task_waiting_time_seconds_bucket{status="finished",le="3599.999999999999"}
2
p_taskmanager_task_waiting_time_seconds_bucket{status="finished",le="+Inf"} 2
p_taskmanager_task_waiting_time_seconds_sum{status="finished"} 702.371079
p_taskmanager_task_waiting_time_seconds_count{status="finished"} 2
p_taskmanager_task_waiting_time_seconds_bucket{status="processing",le="1"} 0
p_taskmanager_task_waiting_time_seconds_bucket{status="processing",le="2.4839607111043738"}
} 0
p_taskmanager_task_waiting_time_seconds_bucket{status="processing",le="6.170060814310146"}
0
p_taskmanager_task_waiting_time_seconds_bucket{status="processing",le="15.326188647871062"}
} 4
p_taskmanager_task_waiting_time_seconds_bucket{status="processing",le="38.06965045228558"}
4
p_taskmanager_task_waiting_time_seconds_bucket{status="processing",le="94.56351600895424"}
4
p_taskmanager_task_waiting_time_seconds_bucket{status="processing",le="234.89205847013176"}
} 4
p_taskmanager_task_waiting_time_seconds_bucket{status="processing",le="583.4626445902387"}
}

4

```
p_taskmanager_task_waiting_time_seconds_bucket{status="processing",le="1449.2982855592077"} 4
```

```
p_taskmanager_task_waiting_time_seconds_bucket{status="processing",le="3599.9999999999999"} 4
```

```
p_taskmanager_task_waiting_time_seconds_bucket{status="processing",le="+Inf"} 4
```

```
p_taskmanager_task_waiting_time_seconds_sum{status="processing"} 35.896736709
```

```
p_taskmanager_task_waiting_time_seconds_count{status="processing"} 4
```

3.6. Настройка профилей транскодирования

3.6.1. Настройка профилей транскодирования

Для настройки профилей транскодирования необходимо поместить сообщение в топик `taskmanager.settings`. При запуске `TaskManager` 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`**.

3.6.2. Транскодирование видеофайлов в зависимости от типа глубины цвета с использованием CPU и GPU

Видео с 8-битной глубиной цвета использует 8 бит (1 байт) информации для представления каждого цвета пикселя. Это означает, что каждый цветовой канал (красный, зеленый, синий) может принимать значения от 0 до 255. В результате возможно представить 16,7 миллиона (256^3) возможных цветовых комбинаций. Видео с 10-битной глубиной цвета использует 10 бит информации для представления каждого цвета пикселя. Это означает, что каждый цветовой канал может принимать значения от 0 до 1023. В результате этого возможно представить более чем 1 миллиард (1024^3) возможных цветовых комбинаций. Однако, использование 10-битного видео требует большего объема памяти и вычислительных ресурсов. При получении Медиаплатформой HDR видео с глубиной цвета 10 бит модуль `Prober` осуществляет проверку соответствия ограничениям.

Между CPU и GPU транскодерами задачи распределяются в зависимости от загрузки транскодеров с учетом глубины цвета видео.

8-битные видео транскодируются с использованием CPU и GPU: секции `cpu` и `gpu` внутри секции `profiles`. Для таких видео формируется только один плейлист 8 бит: параметр `profile_type`.

10-битные видео в Медиаплатформе транскодируются только с использованием декодеров и фильтров CPU: в секции `cpu` внутри секции `profiles` должен быть установлен параметр `pix_bit_depth = 10`. Для таких видео формируется два плейлиста: 8 бит и 10 бит, параметр `profile_type`. При этом транскодирование будет происходить на GPU. Если инициализировать GPU версию этого профиля не удастся, произойдет срабатывание алгоритма аварийного отступления в CPU версию профиля.

В случае, если для профиля установлено транскодирование ВСЕХ найденных в источнике видеопотоков, произойдет фильтрация этих видеопотоков, и в результирующий файл будут включены (транскодированы) только потоки обладающие признаком HDR (глубина пикселя > 8 бит). Для всех остальных потоков в логи будет выдано предупреждение уровня WARNING:

```
"Skipping video stream #1 for profile 'best_hdr' due to source video pixel bits '8' < profile target pixel bits '10'."
```

3.6.3. Формат сообщения

Формат сообщения настройки Task Manager

```
{  
  "message_type": "TASKMANAGER_SETTINGS",  
  
  "ffmpeg_decoding_options": {  
    "threads": "1"  
  },  
  "master_playlist": true,  
  "hw_device": "cuda",  
  
  "profiles": [  
    {  
      "name": "outstanding_2160p_8bit",  
      "container_type": "hls",  
      "filename_extension": "m3u8",
```



```
"pix_bit_depth": 8,
"video": {
  "stream_selection": "best",
  "codec": "libx264",
  "filter_options": {
    "width": 3840,
    "height": 2160,
    "fps": {
      "num": 30,
      "den": 1
    },
    "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": "16M",
    "bufsize": "8M"
  }
},
"audio": {
  "stream_selection": "best",
  "codec": "aac",
  "filter_options": {
    "channels": 2,
```

```
    "samplerate": 48000
  },
  "ffmpeg_encoding_options": {
    "b": "256k"
  }
},
"cpu": {
  "name": "outstanding_2160p_8bit",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 8,
  "video": {
    "stream_selection": "best",
    "codec": "libx264",
    "filter_options": {
      "width": 3840,
      "height": 2160,
      "fps": {
        "num": 30,
        "den": 1
      },
    },
    "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": "16M",
        "bufsize": "8M"
    }
},
"audio": {
    "stream_selection": "best",
    "codec": "aac",
    "filter_options": {
        "channels": 2,
        "samplerate": 48000
    },
    "ffmpeg_encoding_options": {
        "b": "256k"
    }
}
},
"gpu": {
    "name": "outstanding_2160p_8bit",
    "container_type": "hls",
    "filename_extension": "m3u8",
    "pix_bit_depth": 8,
    "video": {
        "stream_selection": "best",
        "codec": "h264_nvenc",
        "filter_options": {
            "width": 3840,
            "height": 2160,
            "fps": {
                "num": 30,
                "den": 1
            },
        },
    },
}
```

```

    "overlay": false,
    "padding": false,
    "rotate": true,
    "allow_swap_dimensions": true
  },
  "ffmpeg_encoding_options": {
    "g": "60",
    "threads": "1",
    "profile": "main",
    "preset": "p1",
    "b": "16777216",
    "bufsize": "16777216",
    "maxrate": "16777216",
    "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"
  }
}
},
},

```

```
{
  "name": "high_1080p_8bit",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 8,
  "video": {
    "stream_selection": "best",
    "codec": "libx264",
    "filter_options": {
      "width": 1920,
      "height": 1080,
      "fps": {
        "num": 30,
        "den": 1
      },
      "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_1080p_8bit",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 8,
  "video": {
    "stream_selection": "best",
    "codec": "libx264",
    "filter_options": {
      "width": 1920,
      "height": 1080,
      "fps": {
        "num": 30,
        "den": 1
      },
    },
    "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_1080p_8bit",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 8,
  "video": {
    "stream_selection": "best",
    "codec": "h264_nvenc",
    "filter_options": {
      "width": 1920,
      "height": 1080,
```

```
"fps": {
  "num": 30,
  "den": 1
},
"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"
```



```

    }
  }
}
},
{
  "name": "medium_720p_8bit",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 8,
  "video": {
    "stream_selection": "best",
    "codec": "libx264",
    "filter_options": {
      "width": 1280,
      "height": 720,
      "fps": {
        "num": 30,
        "den": 1
      },
    },
    "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": "5M",
  }
}

```

```
        "bufsize": "2.5M"
    }
},
"audio": {
    "stream_selection": "best",
    "codec": "aac",
    "filter_options": {
        "channels": 2,
        "samplerate": 48000
    },
    "ffmpeg_encoding_options": {
        "b": "256k"
    }
},
"cpu": {
    "name": "medium_720p_8bit",
    "container_type": "hls",
    "filename_extension": "m3u8",
    "pix_bit_depth": 8,
    "video": {
        "stream_selection": "best",
        "codec": "libx264",
        "filter_options": {
            "width": 1280,
            "height": 720,
            "fps": {
                "num": 30,
                "den": 1
            },
        },
        "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": "5M",
        "bufsize": "2.5M"
    }
},
"audio": {
    "stream_selection": "best",
    "codec": "aac",
    "filter_options": {
        "channels": 2,
        "samplerate": 48000
    },
    "ffmpeg_encoding_options": {
        "b": "256k"
    }
}
},
"gpu": {
    "name": "medium_720p_8bit",
    "container_type": "hls",
    "filename_extension": "m3u8",
    "pix_bit_depth": 8,
    "video": {
        "stream_selection": "best",

```

```
"codec": "h264_nvenc",
"filter_options": {
  "width": 1280,
  "height": 720,
  "fps": {
    "num": 30,
    "den": 1
  },
  "overlay": false,
  "padding": false,
  "rotate": true,
  "allow_swap_dimensions": true
},
"ffmpeg_encoding_options": {
  "g": "60",
  "threads": "1",
  "profile": "main",
  "preset": "p1",
  "b": "5242880",
  "bufsize": "5242880",
  "maxrate": "5242880",
  "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"
    }
}
},
{
    "name": "low_480p_8bit",
    "container_type": "hls",
    "filename_extension": "m3u8",
    "pix_bit_depth": 8,
    "video": {
        "stream_selection": "best",
        "codec": "libx264",
        "filter_options": {
            "width": 854,
            "height": 480,
            "fps": {
                "num": 30,
                "den": 1
            },
            "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": "2.5M",
    "bufsize": "1.75M"
  }
},
"audio": {
  "stream_selection": "best",
  "codec": "aac",
  "filter_options": {
    "channels": 2,
    "samplerate": 48000
  },
  "ffmpeg_encoding_options": {
    "b": "256k"
  }
},
"cpu": {
  "name": "low_480p_8bit",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 8,
  "video": {
    "stream_selection": "best",
    "codec": "libx264",
    "filter_options": {
      "width": 854,
      "height": 480,
      "fps": {
        "num": 30,
        "den": 1
      }
    }
  }
}
```

```

    },
    "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": "2.5M",
    "bufsize": "1.75M"
  }
},
"audio": {
  "stream_selection": "best",
  "codec": "aac",
  "filter_options": {
    "channels": 2,
    "samplerate": 48000
  },
  "ffmpeg_encoding_options": {
    "b": "256k"
  }
}
},
"gpu": {
  "name": "low_480p_8bit",
  "container_type": "hls",

```

```
"filename_extension": "m3u8",
"pix_bit_depth": 8,
"video": {
  "stream_selection": "best",
  "codec": "h264_nvenc",
  "filter_options": {
    "width": 854,
    "height": 480,
    "fps": {
      "num": 30,
      "den": 1
    },
    "overlay": false,
    "padding": false,
    "rotate": true,
    "allow_swap_dimensions": true
  },
  "ffmpeg_encoding_options": {
    "g": "60",
    "threads": "1",
    "profile": "main",
    "preset": "p1",
    "b": "2621440",
    "bufsize": "2621440",
    "maxrate": "2621440",
    "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"
    }
  }
},
{
  "name": "outstanding_2160p_10bit",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 10,
  "video": {
    "stream_selection": "best",
    "codec": "libx264",
    "filter_options": {
      "width": 3840,
      "height": 2160,
      "fps": {
        "num": 30,
        "den": 1
      },
    },
    "overlay": false,
    "padding": false,
    "rotate": true,
    "allow_swap_dimensions": true,
    "pix_fmt": "yuv420p10le"
  }
}

```

```
},
"ffmpeg_encoding_options": {
  "g": "60",
  "threads": "4",
  "profile": "high10",
  "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": "outstanding_2160p_10bit",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 10,
  "video": {
    "stream_selection": "best",
    "codec": "libx264",
    "filter_options": {
      "width": 3840,
```

```
"height": 2160,
"fps": {
  "num": 30,
  "den": 1
},
"overlay": false,
"padding": false,
"rotate": true,
"allow_swap_dimensions": true,
"pix_fmt": "yuv420p10le"
},
"ffmpeg_encoding_options": {
  "g": "60",
  "threads": "4",
  "profile": "high10",
  "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"
  }
}
}
```

```

    }
  },
  {
    "name": "high_1080p_10bit",
    "container_type": "hls",
    "filename_extension": "m3u8",
    "pix_bit_depth": 10,
    "video": {
      "stream_selection": "best",
      "codec": "libx264",
      "filter_options": {
        "width": 1920,
        "height": 1080,
        "fps": {
          "num": 30,
          "den": 1
        },
        "overlay": false,
        "padding": false,
        "rotate": true,
        "allow_swap_dimensions": true,
        "pix_fmt": "yuv420p10le"
      },
      "ffmpeg_encoding_options": {
        "g": "60",
        "threads": "4",
        "profile": "high10",
        "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_1080p_10bit",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 10,
  "video": {
    "stream_selection": "best",
    "codec": "libx264",
    "filter_options": {
      "width": 1920,
      "height": 1080,
      "fps": {
        "num": 30,
        "den": 1
      },
    },
    "overlay": false,
    "padding": false,
    "rotate": true,
    "allow_swap_dimensions": true,
    "pix_fmt": "yuv420p10le"
  }
}
```

```

    },
    "ffmpeg_encoding_options": {
        "g": "60",
        "threads": "4",
        "profile": "high10",
        "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"
    }
}
},
{
    "name": "medium_720p_10bit",
    "container_type": "hls",
    "filename_extension": "m3u8",
    "pix_bit_depth": 10,
    "video": {
        "stream_selection": "best",
        "codec": "libx264",

```

```
"filter_options": {
  "width": 1280,
  "height": 720,
  "fps": {
    "num": 30,
    "den": 1
  },
  "overlay": false,
  "padding": false,
  "rotate": true,
  "allow_swap_dimensions": true,
  "pix_fmt": "yuv420p10le"
},
"ffmpeg_encoding_options": {
  "g": "60",
  "threads": "4",
  "profile": "high10",
  "preset": "superfast",
  "crf": "23",
  "maxrate": "5M",
  "bufsize": "2.5M"
}
},
"audio": {
  "stream_selection": "best",
  "codec": "aac",
  "filter_options": {
    "channels": 2,
    "samplerate": 48000
  },
  "ffmpeg_encoding_options": {
    "b": "256k"
```

```

    }
  },
  "cpu": {
    "name": "medium_720p_10bit",
    "container_type": "hls",
    "filename_extension": "m3u8",
    "pix_bit_depth": 10,
    "video": {
      "stream_selection": "best",
      "codec": "libx264",
      "filter_options": {
        "width": 1280,
        "height": 720,
        "fps": {
          "num": 30,
          "den": 1
        },
        "overlay": false,
        "padding": false,
        "rotate": true,
        "allow_swap_dimensions": true,
        "pix_fmt": "yuv420p10le"
      },
      "ffmpeg_encoding_options": {
        "g": "60",
        "threads": "4",
        "profile": "high10",
        "preset": "superfast",
        "crf": "23",
        "maxrate": "5M",
        "bufsize": "2.5M"
      }
    }
  }
}

```



```

    },
    "audio": {
      "stream_selection": "best",
      "codec": "aac",
      "filter_options": {
        "channels": 2,
        "samplerate": 48000
      },
      "ffmpeg_encoding_options": {
        "b": "256k"
      }
    }
  },
  {
    "name": "low_480p_10bit",
    "container_type": "hls",
    "filename_extension": "m3u8",
    "pix_bit_depth": 10,
    "video": {
      "stream_selection": "best",
      "codec": "libx264",
      "filter_options": {
        "width": 854,
        "height": 480,
        "fps": {
          "num": 30,
          "den": 1
        },
      },
      "overlay": false,
      "padding": false,
      "rotate": true,
    }
  }
}

```

```
    "allow_swap_dimensions": true,
    "pix_fmt": "yuv420p10le"
  },
  "ffmpeg_encoding_options": {
    "g": "60",
    "threads": "4",
    "profile": "high10",
    "preset": "superfast",
    "crf": "23",
    "maxrate": "2.5M",
    "bufsize": "1.75M"
  }
},
"audio": {
  "stream_selection": "best",
  "codec": "aac",
  "filter_options": {
    "channels": 2,
    "samplerate": 48000
  },
  "ffmpeg_encoding_options": {
    "b": "256k"
  }
},
"cpu": {
  "name": "low_480p_10bit",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 10,
  "video": {
    "stream_selection": "best",
    "codec": "libx264",
```

```
"filter_options": {
  "width": 854,
  "height": 480,
  "fps": {
    "num": 30,
    "den": 1
  },
  "overlay": false,
  "padding": false,
  "rotate": true,
  "allow_swap_dimensions": true,
  "pix_fmt": "yuv420p10le"
},
"ffmpeg_encoding_options": {
  "g": "60",
  "threads": "4",
  "profile": "high10",
  "preset": "superfast",
  "crf": "23",
  "maxrate": "2.5M",
  "bufsize": "1.75M"
}
},
"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
}
}
```

Таблица с описанием параметров

Параметр	Тип данных	Формат	Диапазон возможных значений	Обязательно	Описание
<code>message_type</code>	String		TASKMANAGER_SETTINGS	Да	Тип сообщения. Может принимать следующие значения: <ul style="list-style-type: none"> TASKMANAGER_SETTINGS – настройки компонента Task Manager.
<code>ffmpeg_decoding_options</code>	Struct	Группа параметров		Нет	Блок настроек Ffmpeg, передаваемых напрямую в декодер в формате "строка":"строка". Полный список возможных опций можно посмотреть здесь (внешний источник на английском языке). Например: threads - число потоков, которое может использовать декодер, если он поддерживает многопоточность. Возможные значения: <ul style="list-style-type: none"> auto или 0 для автоматического определения оптимального числа потоков; явно указанное число. Значение по умолчанию: auto.
<code>ffmpeg_decoding_options.threads</code>	String		<ul style="list-style-type: none"> auto [положительное целое число] 	Нет	Число потоков, которое может использовать декодер, если он поддерживает многопоточность. Возможные значения: <ul style="list-style-type: none"> auto или 0 для автоматического определения оптимального числа потоков; явно указанное число.

					Значение по умолчанию: auto.
master_playlist	Boolean	Флаг	<ol style="list-style-type: none"> 1. true 2. false 	Да	<p>Включение или отключение генерации общего мастер-плейлиста HLS для всех профилей, у которых параметр <code>container_type</code> имеет значение <code>hls</code>. Доступные значения:</p> <ul style="list-style-type: none"> • true - настройка включена; • false - настройка выключена.
hw_device	String		cuda	Нет	<p>Тип используемого аппаратного ускорения для обработки видео. Допустимое значение: <code>cuda</code>.</p> <p>Поддерживаются карты NVIDIA, драйверы которых совместимы с CUDA 11.6. Кроме того, аппаратное ускорение поддерживается только в режиме, когда декодирование и фильтрация (<code>scaling</code>) и кодирование видео выполняются на GPU. Т. е. частичная обработка на GPU в настоящее время недоступна. Это ограничивает возможный набор кодеков:</p> <ul style="list-style-type: none"> • декодирование: <ul style="list-style-type: none"> ○ <code>mpeg2</code>; ○ <code>av1</code>; ○ <code>h264</code>; ○ <code>h265/hevc</code>; ○ <code>mjpeg</code>; ○ <code>mpeg4</code>; ○ <code>vc1</code>; ○ <code>vp8</code>; ○ <code>vp9</code>. • кодирование: <ul style="list-style-type: none"> ○ <code>h264_nvenc</code>; ○ <code>hevc_nvenc</code>. <p>Аппаратное ускорение не применяется для обработки аудио и создания эскизов видео (<code>thumbnail</code>).</p>

profiles	Struct	Группа параметров		Да	<p>Блок, содержащий профиль или профили обработки медиаданных. Профили могут быть следующих видов:</p> <ul style="list-style-type: none"> • профиль обработки аудио и видео — название профиля может быть произвольным, профиль должен содержать вложенные блоки video и audio — для обработки аудио и видео соответственно; Блоки video и audio можно скомбинировать в одном профиле (видео с аудио) или в разные (видео без аудио или аудио без видео), тогда как блоки thumbnail и trickmode нельзя комбинировать с другими блоками в рамках одного профиля. • профиль создания эскиза (thumbnail) - всегда передаётся в блоке thumbnail; • профиль создания холста миниатюр для ознакомительного просмотра - всегда передаётся в блоке trickmode.
profiles.name	String		Произвольное значение	Да	Название профиля обработки медиаданных. Произвольное значение.
profiles.container_type	String		Список поддерживаемых текущей версией FFmpeg контейнеров можно командой ffmpeg -formats	Да	<p>Итоговый формат файла в рамках обозначений FFmpeg:</p> <ul style="list-style-type: none"> • профиль обработки видео - видеоформат, например, mp4; • профиль обработки (извлечения) аудио -

					<p>аудиоформат, например, mp3;</p> <ul style="list-style-type: none"> • профиль создания эскиза (thumbnail) - формат изображения, например, jpeg. <p>Получить список поддерживаемых текущей версией FFmpeg контейнеров можно командой ffmpeg -formats.</p> <p>Параметр должен быть указан для каждого профиля.</p>
profiles.filename_extension	String	Расширение файла		Нет	Имя расширения итогового файла. Используется при подстановке в шаблоне filename_template.
profiles.video	Struct	Группа параметров		Нет	Блок выходных параметров обрабатываемого видео.
profiles.video.stream_selection	String		<ol style="list-style-type: none"> 1. all 2. best 3. none 	Да	<p>Способ выбора видео- или аудиодорожки или дорожек для обработки и записи в итоговый файл:</p> <ul style="list-style-type: none"> • для видео - параметры выбора видеодорожки; • для аудио - параметры выбора аудиодорожки. <p>Доступны следующие значения:</p> <ul style="list-style-type: none"> • all - все дорожки этого типа; • best - одна «лучшая» дорожка в соответствии с критериями FFmpeg; • none - дорожки этого типа не обрабатываются (например: может быть использовано для профилей генерации аудио файла).

profiles.video.codec	String			Да	<p>Название кодека в формате обозначений FFmpeg. Тип кодека зависит от того, к какому блоку относится параметр:</p> <ul style="list-style-type: none"> • если параметр к обработке видео - видекодек; • если параметр относится к обработке аудио - аудиокодек. <p>Получить список поддерживаемых текущей версией FFmpeg кодеков можно командой <code>ffmpeg – encoders</code>.</p>
profiles.video.filter_options	String	Группа параметров		Да	<p>Блок параметров преобразования:</p> <ul style="list-style-type: none"> • для блока video - преобразование видеодорожки; • для блока audio - преобразование аудиодорожки.
profiles.video.filter_options.width	Integer			Да	Ширина видео в пикселях.
profiles.video.filter_options.height	Integer			Да	Высота видео в пикселях.
profiles.video.filter_options.fps	Struct	Группа параметров		Нет	Блок настроек FPS (Frames per second - кадры в секунду), определяется соотношением значений параметров num и den, входящих в блок. Итоговым значением блока после обработки будет значение соотношения num/den.
profiles.video.filter_options.fps.num	Integer	Положительное число		Да	Число кадров в единицу времени (количество секунд указано в параметре den).
profiles.video.filter_options.fps.den	Integer	Положительное число		Да	Количество секунд, в течение которых отображается количество кадров, определённых в параметре num.

profiles.video.filter_options.pix_fmt	String		Возможные значения можно получить из ffmpeg с помощью команды "ffmpeg -pix_fmts", например "yuv420p"	Нет	Формат пикселей видео. Итоговое видео будет преобразовано в указанный пиксель формат. Пустая строка означает автоматический подбор формата пикселей на основе формата пикселей исходного видео и возможностях кодека. Значение по умолчанию: пустая строка.
profiles.video.filter_options.padding	String	Флаг	true, false	Нет	Если true, части видео, которые образовались из-за разного соотношения сторон кадра на входе и выходе, заполняются черным цветом. Значение по умолчанию - false.
profiles.video.filter_options.rotate	String	Флаг	true, false	Нет	Если true, то при наличие в метаданных исходного видео значения rotate, отличного от нуля, делается поворот кадра. Значение по умолчанию - false.
profiles.video.filter_options.allow_swap_dimensions	String	Флаг	true, false	Нет	Если true, то разрешается сделать замену width <-> height, если это оптимальнее для входной картинки. Значение по умолчанию - true.
profiles.video.filter_options.overlay	String	Флаг	true, false	Нет	Если true, то видео, указанное в секции overlay (далее – водяной знак), будет наложено поверх основного видео профиля от левого угла. Предварительно, видео будет масштабировано по размеру, указанному для профиля, с сохранением оригинальных пропорций. Значение по умолчанию - false.
profiles.video.ffmpeg_encoding_options	Struct	Группа параметров		Нет	Блок настроек FFmpeg, передаваемых напрямую в кодировщик в формате "строка": "строка". Полный список возможных

					<p>опций можно посмотреть по ссылке https://ffmpeg.org/ffmpeg-codecs.html (на английском языке). Ниже представлены примеры опций.</p> <ul style="list-style-type: none"> • preset - имя пресета при кодировании видео в формате h.264. Для кодировщика libx264 список возможных значений: fast, superfast, ultrafast и т. п. Полный список значений можно посмотреть по адресу https://trac.ffmpeg.org/wiki/Encode/H.264 (на английском языке). Для кодировщика h264_nvenc (при работе с GPU) список возможных значений отличается: slow, medium, fast, hp, hq и т. п. Полный список значений можно посмотреть через командную строку: <code>ffmpeg -h encoder=h264_nvenc</code>. • b - значение фиксированного битрейта видео. Может быть выражено в бит/с, так и сокращением, например: 2М, 64К. В случае кодека libx264 число означает не бит/с, а Кбит/с. • threads - число потоков, которое может использовать кодировщик, если он поддерживает многопоточность.
--	--	--	--	--	--

					Возможные значения auto или 0 для автоматического определения оптимального числа потоков либо конкретное число. Значение по умолчанию: auto.
profiles.video.ffmpeg_encoding_options.g	Integer			Нет	Размещение ключевого кадра. Размер GOP устанавливает максимальное расстояние между ключевыми кадрами; если ноль, выходной поток будет только внутренним. Минимальное расстояние игнорируется, если оно не совпадает с размером GOP, и в этом случае ключевые кадры всегда будут появляться с фиксированным интервалом. Значение по умолчанию: не задано.
profiles.video.ffmpeg_encoding_options.preset	String		Для кодировщика libx264 список возможных значений: fast, superfast, ultrafast и т.п. Полный список значений можно посмотреть здесь: https://trac.ffmpeg.org/wiki/Encode/H.264 . Примечание: для кодировщика h264_nvenc (при работе с GPU) список возможных значений отличается: slow, medium, fast, hp, hq и т.п. Полный список значений можно посмотреть через командную строку ffmpeg -h encoder=h264_nvenc.	Нет	Имя пресета при кодировании видео в формате h.264. Для кодировщика libx264 список возможных значений: fast, superfast, ultrafast и т.п. Полный список значений можно посмотреть здесь: https://trac.ffmpeg.org/wiki/Encode/H.264 . Примечание: для кодировщика h264_nvenc (при работе с GPU) список возможных значений отличается: slow, medium, fast, hp, hq и т.п. Полный список значений можно посмотреть через командную строку ffmpeg -h encoder=h264_nvenc.
profiles.video.ffmpeg_encoding_options.b	String	Может быть		Нет	Значение фиксированного битрейта видео.

		как число, выраженное в бит/с, так и сокращения, например, “2М”, “64К”. Примечание: в случае кодека libx264 число означает не бит/с, а Кбит/с.			
profiles.video.ffmpeg_encoding_options.threads	String		<ul style="list-style-type: none"> • auto • [положительное целое число] 	Нет	Число потоков, которое может использовать кодировщик, если он поддерживает многопоточность. Возможные значения: <ul style="list-style-type: none"> • auto или 0 для автоматического определения оптимального числа потоков, • конкретное число. Значение по умолчанию: auto.
profiles.video.ffmpeg_encoding_options.profile	String		main	Нет	Параметр ffmpeg для кодека. Установлено значение main для всех профилей транскодирования для возможности декодирования h264 всеми браузерами и мобильными устройствами.
profiles.video.ffmpeg_encoding_options.bufsize	Integer	Может быть как число, выраже	0 - 8M	Нет	Размер буфера управления битрейтом.

		нное в бит/с, так и сокращения, например, “2М”, “64К“			
profiles.video.ffmpeg_encoding_options.maxrate	Integer	Может быть как число, выраженное в бит/с, так и сокращения, например, “2М”, “64К“	0 - 16М	Нет	Максимальный битрейт видео.
profiles.video.ffmpeg_encoding_options.tune	String			Нет	Настройка параметров декодирования.
profiles.video.ffmpeg_encoding_options.refs	Integer		0-16	Нет	Количество ключевые кадров компенсации движения, которые может использовать каждый P-кадр.
profiles.video.ffmpeg_encoding_options.no-scenecut	Integer		0-100	Нет	Значение, используемое кодировщиком для расчета, должен ли он форсировать ключевой кадр (вставка дополнительного I-кадра).
profiles.video.ffmpeg_encoding_options.coder	String			Нет	Метод кодирования.
profiles.video.ffmpeg_encoding_options.crf	Integer		0-63	Нет	Коэффициент фиксированной скорости передачи данных, компромисс между качеством и размером для режимов постоянного качества (без целевого битрейта) и ограниченного качества (с максимальным целевым битрейтом), более высокие числа указывают на более

					низкое качество и меньший размер результирующего файла.
profiles.audio	Struct	Группа параметров		Нет	Блок выходных параметров обрабатываемой аудиодорожки.
profiles.audio.stream_selection	String		<ol style="list-style-type: none"> 1. all 2. best 3. none 	Да	<p>Способ выбора видео- или аудиодорожки или дорожек для обработки и записи в итоговый файл:</p> <ul style="list-style-type: none"> • для видео - параметры выбора видеодорожки; • для аудио - параметры выбора аудиодорожки. <p>Доступны следующие значения:</p> <ul style="list-style-type: none"> • all - все дорожки этого типа; • best - одна «лучшая» дорожка в соответствии с критериями FFmpeg; • none - дорожки этого типа не обрабатываются (например, может быть использовано для профилей генерации аудио файла).
profiles.audio.codec	String	Группа параметров		Да	<p>Название кодека в формате обозначений FFmpeg. Тип кодека зависит от того, к какому блоку относится параметр:</p> <ul style="list-style-type: none"> • если параметр к обработке видео - видеокодек; • если параметр относится к обработке аудио - аудиокодек. <p>Получить список поддерживаемых текущей версией FFmpeg кодеков</p>

					можно командой ffmpeg – encoders.
profiles.audio.filter_options	String	Группа параметров		Да	Блок параметров преобразования: <ul style="list-style-type: none"> • для блока video - преобразование видеодорожки; • для блока audio - преобразование аудиодорожки.
profiles.audio.filter_options.channels	Integer		1.1 2.2	Да	Количество каналов аудиодорожки. Возможны следующие значения: <ul style="list-style-type: none"> • 1 - моно; • 2 - стерео.
profiles.audio.filter_options.samplerate	Integer	В Гц		Да	Частота аудио кадров.
profiles.audio.ffmpeg_encoding_options	Struct			Нет	Блок настроек FFmpeg, передаваемых напрямую в кодировщик в формате "строка": "строка". Полный список возможных опций можно посмотреть на сайте https://ffmpeg.org/ffmpeg-codecs.html (на английском языке). Например: <ul style="list-style-type: none"> • b - значение фиксированного битрейта аудио: может выраженным в бит/с, так и сокращением, например, 2М, 64К.
profiles.audio.ffmpeg_encoding_options.b	String	Может быть как число, выраженное в бит/с, так и сокращения, например,		Нет	Значение фиксированного битрейта аудио.

		“2М”, “64К”			
thumbnail	Struct	Группа параметров		Нет	Блок параметров генерации набора эскизов (thumbnail) видео.
thumbnail.number	Integer			Да	Количество эскизов (thumbnail), которое необходимо сгенерировать. Параметры number, offset и step в блоке thumbnail должны быть согласованы между собой так, чтобы временная позиция последней миниатюры не превысила 100% относительно полной длины видео.
thumbnail.offset	Float			Да	Смещение временной позиции первого эскиза (thumbnail) относительно начала видео, в процентах от полной длины видео. Параметры number, offset и step в блоке thumbnail должны быть согласованы между собой так, чтобы временная позиция последней миниатюры не превысила 100% относительно полной длины видео.
thumbnail.step	Float			Да	Смещение временных позиций эскизов (thumbnail) относительно друг друга в процентах от полной длины видео. Параметры number, offset и step в блоке thumbnail должны быть согласованы между собой так, чтобы временная позиция последней миниатюры не превысила 100% относительно полной длины видео.
thumbnail.width	Integer			Да	Ширина эскиза (thumbnail) в пикселях.
thumbnail.height	Integer			Да	Высота эскиза (thumbnail) в пикселях.

thumbnail.quality	Integer		от 1 до 31	Да	Уровень качества изображения. Допустимый диапазон от 1 до 31 включительно, где 1 — самое высокое качество/самый большой размер файла, а 31 — самое низкое качество/наименьший размер файла. Подробнее см. аналогичную опцию FFmpeg: https://trac.ffmpeg.org/wiki/Encode/MPEG-4 (на английском языке).
trickmode	Struct	Группа параметров		Нет	Блок информации о сгенерированных наборах миниатюр для ознакомительного просмотра.
trickmode.frame_width	Integer		от 4 до 16254	Да	Ширина эскиза для ознакомительного просмотра. Минимальное значение – 4 пикселя, максимальное значение – 16254, делённое на количество столбцов холста.
trickmode.frame_height	Integer		от 4 до 16254	Да	Высота миниатюры. Минимальное значение – 4 пикселя, максимальное значение – 16254, делённое на количество строк холста.
trickmode.columns	Integer			Да	Количество столбцов с эскизами на холсте для ознакомительного просмотра (т. е. количество миниатюр в одном ряду).
trickmode.rows	Integer			Да	Количество рядов с миниатюрами на холсте.
trickmode.frames	Integer			Да, если нет step	Общее количество сгенерированных эскизов для ознакомительного просмотра по всем холстам (результатирующим файлам). Параметры step и frames в блоке trickmode взаимоисключающие.

trickmode.step	Float		логично, если step равен длине сегмента видео (чанка), т.е. 2 секунды	Да, если нет frames	Шаг в секундах между двумя соседними эскизами для ознакомительного просмотра. Параметры step и frames в блоке trickmode взаимоисключающие.
trickmode.quality	Integer		от 1 до 31	Да	Уровень качества изображения. Допустимый диапазон от 1 до 31 включительно, где 1 — самое высокое качество/самый большой размер файла, а 31 — самое низкое качество/наименьший размер файла. Подробнее см. аналогичную опцию FFmpeg: https://trac.ffmpeg.org/wiki/Encode/MPEG-4 (на английском языке).
pix_bit_depth	Integer		8 или 10	Нет	Позволяет отметить профиль для транскодирования HDR видео (видео с битностью пикселя > 8).
probe	Struct	Группа параметров		Да, если нет step	Блок параметров модуля Prober.
probe.check_demux	Boolean	Флаг	1. true 2. false	Нет	Проверка доступности и корректной считываемости данных перед декодированием. Возможные значения: <ul style="list-style-type: none"> • true (истина); • false (ложь).
probe.check_decode	Boolean	Флаг	1. true 2. false	Нет	Проверка, что выбранные потоки с лучшим качеством (best) успешно декодируются. Возможные значения: <ul style="list-style-type: none"> • true (истина); • false (ложь).

3.7. Apache Kafka

Для взаимодействия компонентов Медиаплатформы используется брокер сообщений Kafka. Компонент-отправитель помещает сообщение в нужный топик, после чего получатель выполняет задачу в соответствии с параметрами, указанными в этом сообщении. По

выполнении задачи получатель помечает сообщение как прочитанное и переходит к следующему сообщению в топике.

Медиаплатформа оперирует топиками Kafka. Имена этих топиков задаются переменными окружения, которые одинаковы у всех служб (см. таблицу ниже).

Топик Kafka	Описание	Отправитель	Получатель
--------------------	-----------------	--------------------	-------------------

<p>KAFKA_TOPIC_TASK_MANAGER</p>	<p>В этот топик поступают сообщения об этапах загрузки и обработки исходных видео (а также об ошибках, возникающих в процессе обработки):</p> <ul style="list-style-type: none"> • сообщение о создании задачи на загрузку исходного видео; • сообщения о начале загрузки исходного видео; • сообщения о завершении и загрузке исходного видео в хранилище Медиаплатформы; • сообщение о завершении и работы сервиса Probe, которое содержит статус и информацию об исходном видео; • сообщения о создании эскиза (thumbnail) исходного видео 	<ul style="list-style-type: none"> • Uploader • Probe • Transcoder 	<p>Task Manager</p>
--	--	---	---------------------

	<p>(успешном или неуспешном);</p> <ul style="list-style-type: none"> • сообщения о завершении и обработки исходного видео (успешном или неуспешном). 		
KAFKA_TOPIC_TASKMANAGER_SETTINGS	<p>В этом топике хранятся настройки профилей обработки исходного видео — эти настройки считываются при запуске компонента Task Manager и обновляются при получении каждого нового сообщения в этот топик.</p>	<p>Может быть использован любой клиент, позволяющий отправить сообщение в данный топик. Содержимое сообщения настроек профилей обработки медиафайла задаётся на этапе первоначальной настройки Медиаплатформы.</p>	Task Manager
KAFKA_TOPIC_PROBE	<p>Топик, из которого Prober считывает задачи на анализ загруженного исходного видео.</p>	Task Manager	Prober

KAFKA_TOPIC_TRANSCODER	Топик, из которого Transcoder считывает задачи на обработку исходного видео. Если в системе присутствуют CPU и GPU транскодеры, то используются два топика – один для CPU транскодеров, другой для GPU транскодеров.	Task Manager	Transcoder
-------------------------------	--	--------------	------------

<p>KAFKA_TOPIC_STATUS</p>	<p>Топик, в который Task Manager отправляет сведения о статусах загрузки и обработки видео после сообщений от других компонентов (Uploader, Prober, Transcoder). Этот топик предназначен для чтения извне Медиаплатформы — в него поступают сообщения о следующих статусах:</p> <ul style="list-style-type: none"> • начало загрузки исходного видео; • завершение загрузки исходного (успешное или неуспешное); • создание эскиза (thumbnail) исходного видео (успешное или неуспешное); • завершение обработки исходного видео (успешное или неуспешное); 	<p>Task Manager</p>	<p>Сообщения топика KAFKA_TOPIC_STATUS предназначены для чтения извне платформы.</p>
----------------------------------	---	---------------------	--

1	<p>Task Manager считывает настройки в топике KAFKA_TOPIC_TASKMANAGER_SETTINGS при запуске компонента. Считываемое сообщение помещается в топик на этапе первоначальной настройки Медиаплатформы, параметр message_type (тип сообщения) в нём имеет значение TASKMANAGER_SETTINGS.</p>	<p>Может быть использован любой клиент, позволяющий отправить сообщение в данный топик. Содержит мое сообщение настроек профиля обработки медиафайла задаётся на этапе первоначальной настройки Медиаплатформы</p>	KAFKA_TOPIC_TASKMANAGER_SETTING S	Task Manager
2	<p>После получения задачи на загрузку исходного медиафайла, формирования идентификатора задачи (task id) и получения идентификатора загрузки (upload id) компонент Uploader отправляет в топик KAFKA_TOPIC_TASK_MANAGER сообщение UPLOADER_TASK, в котором параметр message_type (тип сообщения) имеет значение UPLOADER_TASK_CREATE (создание задачи загрузки).</p>	Uploader	KAFKA_TOPIC_TASK_MANAGER	Task Manager

3	<p>Компонент Task Manager отправляет в топик KAFKA_TOPIC_STATUS сообщение TASK_STATUS, в котором:</p> <ul style="list-style-type: none"> • параметр message_type (тип сообщения) имеет значение TASK_STATUS (статус задачи); • параметра state (статус) имеет значение UploaderStarted (загрузка начата). 	Task Manager	KAFKA_TOPIC_STAT US	Сообщения топика KAFKA_TOPIC_STATUS предназначены для чтения извне платформы
4	<p>Считав сообщение о завершении загрузки, компонент Task Manager отправляет в топик KAFKA_TOPIC_STATUS сообщение TASK_STATUS, в котором:</p> <ul style="list-style-type: none"> • параметр message_type (тип сообщения) имеет значение TASK_STATUS; • параметр state (статус) может иметь одно из следующих значений: <ul style="list-style-type: none"> ○ UploaderFinished (загрузка завершена); ○ UploaderFailed (сбой загрузки). 	Task Manager	KAFKA_TOPIC_STAT US	Сообщения топика KAFKA_TOPIC_STATUS предназначены для чтения извне платформы
5	<p>Uploader отправляет в топик KAFKA_TOPIC_TASK_MANAGER сообщение TRANSFER_TASK, в котором параметр message_type (тип сообщения) имеет значение TRANSFER_TASK_START (начал о перемещения).</p> <p>После получения этого сообщения Task Manager не отправляет статус задачи в топик KAFKA_TOPIC_STATUS.</p>	Uploader	KAFKA_TOPIC_TASK _MANAGER	Task Manager
6	<p>Uploader отправляет в топик KAFKA_TOPIC_TASK_MANAGER сообщение TRANSFER_TASK, в котором параметр message_type (тип сообщения) имеет значение TRANSFER_TASK_FINISHED (перемещение завершено).</p> <p>После получения этого сообщения Task Manager не отправляет статус задачи в топик KAFKA_TOPIC_STATUS.</p>	Uploader	KAFKA_TOPIC_TASK _MANAGER	Task Manager

7	<p>Task Manager отправляет в топик KAFKA_TOPIC_PROBE сообщение PROBE_TASK, содержащее задачу на анализ загруженного медиафайла. В этом сообщении параметр message_type (тип сообщения) имеет значение PROBE_TASK (задача на анализ медиафайла).</p>	Task Manager	KAFKA_TOPIC_PROBE	Prober
8	<p>После анализа (успешного или неуспешного) исходного медиафайла компонент Prober отправляет в топик KAFKA_TOPIC_TASK_MANAGER сообщение PROBE_TASKFINISHED, в котором параметр message_type имеет значение PROBE_TASKFINISHED (анализ завершён).</p>	Prober	KAFKA_TOPIC_TASK_MANAGER	Task Manager
9	<p>Получив сообщение об анализе медиафайла, Task Manager отправляет в топик KAFKA_TOPIC_STATUS сообщение TASK_STATUS, в котором:</p> <ul style="list-style-type: none"> • параметр message_type (тип сообщения) имеет значение TASK_STATUS; • параметр state (статус) может иметь одно из следующих значений: <ul style="list-style-type: none"> ○ ProbeFinished (анализ успешно завершён); ○ ProbeFailed (сбой анализа). 	Task Manager	KAFKA_TOPIC_STATUS	Сообщения топика KAFKA_TOPIC_STATUS предназначены для чтения извне платформы

1 0	<p>На основе настроек из топика KAFKA_TOPIC_TASKMANAGER_SETTINGS Task Manager отправляет в топик KAFKA_TOPIC_TRANSCODER сообщение TRANSCODER_TASK, содержащее задачу на обработку загруженного медиафайла. В этом сообщении параметр message_type (тип сообщения) имеет значение TRANSCODER_TASK. Task Manager считывает сообщение с настройками при запуске. Для постановки задач на обработку используется самое новое сообщение. В топике KAFKA_TOPIC_TASKMANAGER_SETTINGS это сообщение TASKMANAGER_SETTINGS. При получении задачи на обработку медиафайла Transcoder запускает три параллельных процесса:</p> <ul style="list-style-type: none"> • создание эскиза видео (thumbnail) (шаги 30-35); • транскодирование файла (шаги 36-38); • формирование трикмода (шаги 39-41). 	Task Manager	KAFKA_TOPIC_TRANSCODER	Transcoder
1 1	<p>Если задание на обработку исходного медиафайла содержит профиль для создания эскиза (thumbnail) медиафайла, после создания (или сбоя при создании) такого эскиза Transcoder в топик KAFKA_TOPIC_TASK_MANAGER отправляет сообщение THUMBNAIL_INFO, в котором параметр message_type (тип сообщения) имеет значение THUMBNAIL_INFO (сведения об эскизе).</p>	Transcoder	KAFKA_TOPIC_TASK_MANAGER	Task Manager

1 2	<p>После получения сообщения о создании — успешном или неуспешном — эскиза (thumbnail), Task Manager отправляет в топик KAFKA_TOPIC_STATUS сообщение TASK_STATUS, в котором:</p> <ul style="list-style-type: none"> • параметр message_type (тип сообщения) имеет значение TASK_STATUS (статус задачи); • параметр state (статус) имеет одно из следующих значений: <ul style="list-style-type: none"> ○ ThumbnailReady (эскиз готов); ○ ThumbnailFailed (сбой создания эскиза). 	Task Manager	KAFKA_TOPIC_STATUS	Сообщения топика KAFKA_TOPIC_STATUS предназначены для чтения извне платформы
1 3	<p>После обработки — успешной или неуспешной — медиафайла Transcoder отправляет в топик KAFKA_TOPIC_TASK_MANAGER сообщение TRANSCODER_TASKFINISHED, в котором параметр message_type (тип сообщения) имеет значение TRANSCODER_TASKFINISHED (обработка завершена).</p>	Transcoder	KAFKA_TOPIC_TASK_MANAGER	Transcoder
1 4	<p>После получения сообщения об обработке медиафайла (и извлечении аудиодорожки) Task Manager отправляет в топик KAFKA_TOPIC_STATUS сообщение TASK_STATUS, в котором:</p> <ul style="list-style-type: none"> • параметр message_type (тип сообщения) имеет значение TASK_STATUS; • параметр state (статус) может иметь следующие значения: <ul style="list-style-type: none"> ○ TranscoderFinished (обработка завершена); ○ TranscoderFailed (сбой обработки). <p>Задание получает общий статус FAILED в случае ошибки при работе хотя бы с одним профилем. При этом выходные файлы по корректно обработанным профилям сохраняются и в итоговом сообщении присутствуют ссылки на эти файлы.</p>	Task Manager	KAFKA_TOPIC_STATUS	Сообщения топика KAFKA_TOPIC_STATUS предназначены для чтения извне платформы

3.8. Мониторинг

Мониторинг за функционированием сервисов Медиаплатформы реализован в виде набора сервисов, подключающихся к модулям по протоколам OpenTelemetry и Prometheus. Для наблюдения трейсов используется сервис Zipkin, для метрик – Prometheus и Grafana.

3.8.1. Функциональная схема

В процессе функционирования сервисы Медиаплатформы предоставляют данные о производительности своих внутренних процессов:

1. данные о таймингах (продолжительности работы) тех или иных процессов (процедур) собираются модулем и самостоятельно выгружаются в виде метаданных для диаграммы Ганта (метод PUSH). Для этого модулю передаются настройки модуля Zipkin, который также разворачивается и настраивается в кластере. Единицей информации является трейс (trace) – элемент, отображающийся на диаграмме Ганта в виде отрезка, имеющий позицию начала относительно других трейсов, продолжительность и прочие метаданные, упрощающие чтение схемы. В настоящий момент трейсы организуются только отдельно для каждого компонента системы. В дальнейшем планируется связка отдельных трейсов разных компонентов в логически связанные блоки, отражающие бизнес-логику функционирования платформы;
2. данные о метриках (количественные показатели) компонентов системы собираются и обновляются каждым модулем в процессе его долгосрочного функционирования. Эти показатели становятся доступными для наблюдения после их сборки (метод PULL) модулем Prometheus, который также должен быть развернут в кластере. Для целей визуализации собранных Prometheus метрик используется модуль Grafana, тоже развернутый в кластере платформы.

3.8.2. Трейсы

Трейсы собираются двух типов:

1. Lifetime - применяется для отображения общего времени жизни того или иного объекта/процесса в контексте функционирования модуля.
2. Aggregator - применяется для учета суммарного времени работы тех или иных процедур/функций/методов в контексте функционирования модуля. На диаграмме Ганта трейс позиционируется по первому использованию процедуры/функции/метода.

Структура трейсов имеет многоуровневую вложенную структуру. В данном документе описываются только трейсы верхнего уровня тех модулей, которые реализуют функционал работы с телеметрией.

Модуль	Название трейса	Тип
Transcoder	pipeline	Lifetime
Prober	probe	Lifetime

3.8.3. Метрики в Grafana

3.8.3.1. Общее описание

Grafana — это платформа с открытым исходным кодом для визуализации, мониторинга и анализа данных. Grafana позволяет пользователям создавать дашборды с панелями, каждая из которых отображает определенные показатели в течение установленного периода времени. Дашборды составляются на основе данных, отправляемых сервисами Медиаплатформы.

Данные приведены в соответствующих разделах:

- Метрики Uploader;
- Метрики Prober;
- Метрики Transcoder;
- Метрики Task Manager.

В данном разделе приведены примеры дашбордов и их настройки.

3.8.3.2. Навигация дашборда


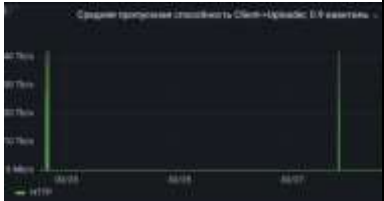
Сервисы развернуты на кластере Kubernetes в определенном namespace.

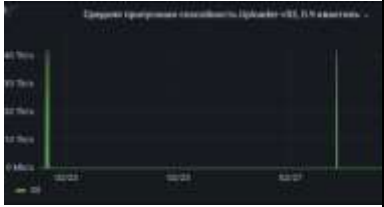

Название	Описание
Namespace	В раскрывающемся списке "namespace" находятся стенды, с которых поступают данные для графиков.
Mr_service	В раскрывающемся списке "mr_service" отражены поды модулей Медиаплатформы с одного стенда указанного в namespace.
All	Вкладка "All" выдает данные со всех подов выбранного модуля Медиаплатформы на одном стенде/namespace. На примере изображения у namespace rupor-beta-platform накопленное значение со всех транскодеров равно: 30 - общее количество заданий, 0 - количество неуспешных заданий.



3.8.3.3. Примеры графиков

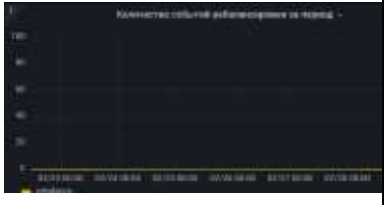
Дашборд	Название	Тип	Описание	Параметры	Пример изображения
Uploader	Количество неверных запросов за период (экстраполяция)	График	Средняя экстраполированная скорость изменения количества запросов на загрузку, не прошедших валидацию, за период 60 секунд.	Период сброса счетчика: 60 секунд	
	Общее количество неверных запросов	Счетчик	Количество запросов на загрузку, не прошедших валидацию за время работы сервиса с момента последней перезагрузки.	Накопленное значение	
	Количество загружаемых файлов за период (экстраполяция)	График	Средняя экстраполированная скорость изменения величины общих загрузок файлов (успешных и неуспешных) и незавершенных объединений чанков за определенный период загружаемых файлов. Если в начале этого периода (60 сек) началась загрузка файлов и прекратилась, то функция экстраполяции на основе имеющихся данных выводит график скорости загружаемых файлов до полного периода. График в примере показывает, что в 14:00:00 счетчик общего числа	Период сброса счетчика: 60 секунд	



			<p>загрузок изменялся со скоростью в 4 загрузки в минуту (60 сек).</p> <ul style="list-style-type: none"> • постоянное низкое значение означает неактивность сервера и нерациональное использование ресурсов. 		
	Общее количество загруженных файлов	Счетчик	Количество неуспешных загрузок (в том числе незавершенные объединения чанков) и общее количество загрузок с момента последней перезагрузки.	Накопленное значение	
	Количество загружаемых чанков за период (экстраполяция)	График	<p>Средняя экстраполированная скорость изменения величины общих загрузок чанков (успешных и неуспешных) за определенный период загружаемых чанков.</p> <p>Если в начале этого периода (60 сек) началась загрузка чанков и прекратилась, то функция экстраполяции на основе имеющихся данных выводит график скорости загружаемых чанков до полного периода. График в примере показывает, что в 14:15:30 счетчик</p>	Период сброса счетчика: 60 секунд	

			<p>общего числа загрузок изменялся со скоростью в 156 загрузок в минуту (60 сек).</p> <ul style="list-style-type: none"> значение этой метрики может помочь определить, насколько интенсивно нагружен сервис. 		
Общее количество загруженных чанков	Счетчик	Количество неуспешных загрузок и общее количество загрузок - накопленное значение за время работы сервиса с момента последней перезагрузки.	Накопленное значение		
Средняя пропускная способность Client->Uploader, 0.9 квантиль	График	<p>Средняя пропускная способность загрузки данных от клиента до Uploader (Мбит в секунду). Uploader анализирует размер и скорость загрузки каждого чанка, передает данные Prometheus, который статистически подсчитывает, что в 90% всех случаев скорость загрузки имела определенное значение (Мбит в секунду) в определенный промежуток времени.</p> <ul style="list-style-type: none"> значение этой метрики может помочь 	Случайные величины при 90% вероятности		

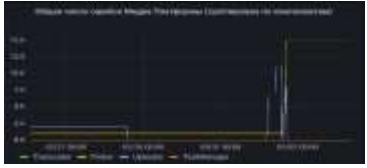
			определить среднюю нагрузку на сервис.		
	Средняя пропускная способность Uploader->S3, 0.9 квантиль	График	<p>Средняя пропускная способность загрузки данных из Uploader в S3 по заданному квантилю (Мбит в секунду). Uploader анализируют размер и скорость загрузки каждого чанка, передает данные Prometheus, который статистически подсчитывает, что в 90% всех случаев скорость загрузки имела определенное значение (Мбит в секунду) в определенный промежуток времени.</p> <ul style="list-style-type: none"> • может помочь определить среднюю нагрузку на сеть на разных участках микросервиса (загрузка файлов от клиента и загрузка файлов в S3). 	Случайные величины при 90% вероятности	
Task Manager	Количество ошибок за период (экстраполяция)	График	<p>Средняя экстраполированная скорость изменения количества ошибок при обработке сообщений за период.</p> <ul style="list-style-type: none"> • по мере увеличения 	Период сброса счетчика: 60 секунд	

			<p>значения этой метрики увеличивается количество ошибок;</p> <ul style="list-style-type: none"> • высокое значение метрики может указывать на проблемы с обработкой файлов; • в норме значение этой метрики должно стремиться к нулю, что будет означать, что ошибки в модуле Task Manager отсутствуют, файлы транскодируются. 		
Общее количество ошибок	Счетчик	Количество ошибок при обработке сообщений за время работы сервиса с момента последней перезагрузки.	Накопленное значение		
Количество заданий на транскодирование	График	<p>Количество необработанных заданий на транскодирование (размер очереди).</p> <ul style="list-style-type: none"> • высокое значение метрики может указывать на то, что число транскодеров меньше, чем необходимо, 	-		

			<p>для того, чтобы все задания начали обрабатываться сразу по мере поступления в кластер;</p> <ul style="list-style-type: none"> • в норме значение этой метрики должно быть не больше числа транскодеров в кластере. 		
Prober Transcod er	Состояние сервиса: обработка задания	График	Текущее состояние сервиса.	0 - ожидание 1 - обработка задания	
	Количество событий ребалансировки за период (экстраполяция)	График	<p>Средняя экстраполированная скорость изменения величины количества событий ребалансировки подписки на входной топик Kafka, за период в 60 секунд.</p> <ul style="list-style-type: none"> • высокое значение метрики может указывать на проблемы с Apache Kafka или кластером; • в норме значение этой метрики должно стремиться к нулю, что будет 	Период сброса счетчика: 60 секунд	

			<p>означать, что Kafka работает стабильно;</p> <ul style="list-style-type: none"> • значение должно быть 0 с редкими подъемами до 1. Оптимальный график - плато; • высокая скорость роста значения говорит о том, что поды выходят из строя слишком часто. 		
Количество заданий за период (экстраполяция)	График	<p>Средняя экстраполированная скорость изменения величины общих заданий (успешных и неуспешных) за определенный период загружаемых заданий.</p> <ul style="list-style-type: none"> • значение этой метрики может помочь определить, насколько интенсивно нагружен сервис. 	Период сброса счетчика: 60 секунд		
Общее количество заданий	Счетчик	Количество неуспешных заданий и общее количество заданий за время работы сервиса с	Накопленное значение		

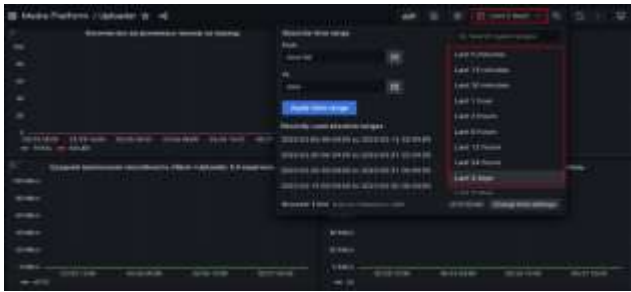
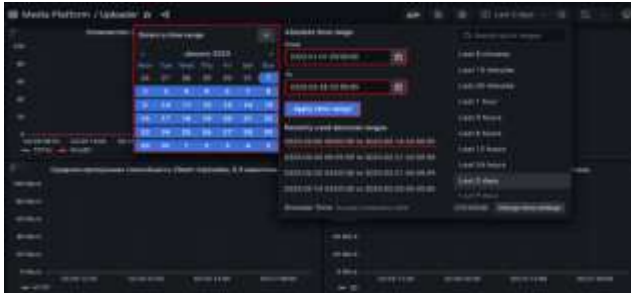
			момента последней перезагрузки.		
Средняя пропускная способность, 0.9 квантиль	График	<p>Средняя пропускная способность сервиса по заданному квантилю (Мбит в секунду).</p> <ul style="list-style-type: none"> • высокое значение метрики может указывать на высокую скорость сети; • низкое значение метрики может указывать на проблемы с сетью доступа в хранилище S3 (медленная обработка файлов из-за долгого ожидания загрузки файлов из хранилища S3). 	Случайные величины при 90% вероятности		
Среднее время обработки заданий, 0.9 квантиль	График	<p>Среднее время обработки задания по заданному квантилю в секундах.</p> <p>Если график в верхней строке таблицы (Средняя пропускная способность, 0.9 квантиль) показывает низкую пропускную способность, то среднее время обработки файла</p>	Случайные величины при 90% вероятности		


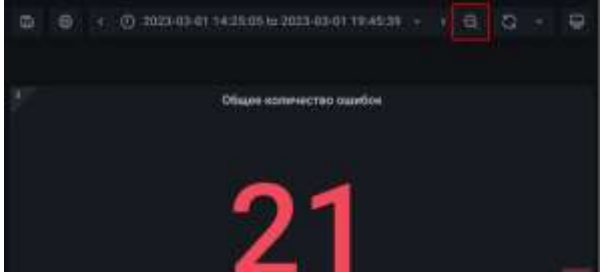

			<p>автоматически возрастет.</p> <ul style="list-style-type: none"> низкое значение метрики указывает на высокую скорость обработки файла. 		
Overall	Общее число ошибок Медиа Платформы (группировка по компонентам)	График	<p>Сумма ошибочных заданий по типу сервисов. Линия Transcoder показывает сумму failed заданий модуля Transcoder. Линия Prober показывает сумму failed задач модуля Prober. Линия Uploader показывает сумму failed загрузок файлов модуля Uploader. Линия Task_Manager показывает счетчик ошибок модуля Task_Manager.</p> <ul style="list-style-type: none"> в норме значение этой метрики должно стремиться к нулю, что будет означать отсутствие ошибок на сервисе; высокое значение метрики на одном из сервисов может указывать на проблемы с 	Период отсутствует. Возрастающий счетчик.	



			обработкой файлов на этом сервисе.		
--	--	--	------------------------------------	--	--

3.8.3.4. Настройки панели дашборда

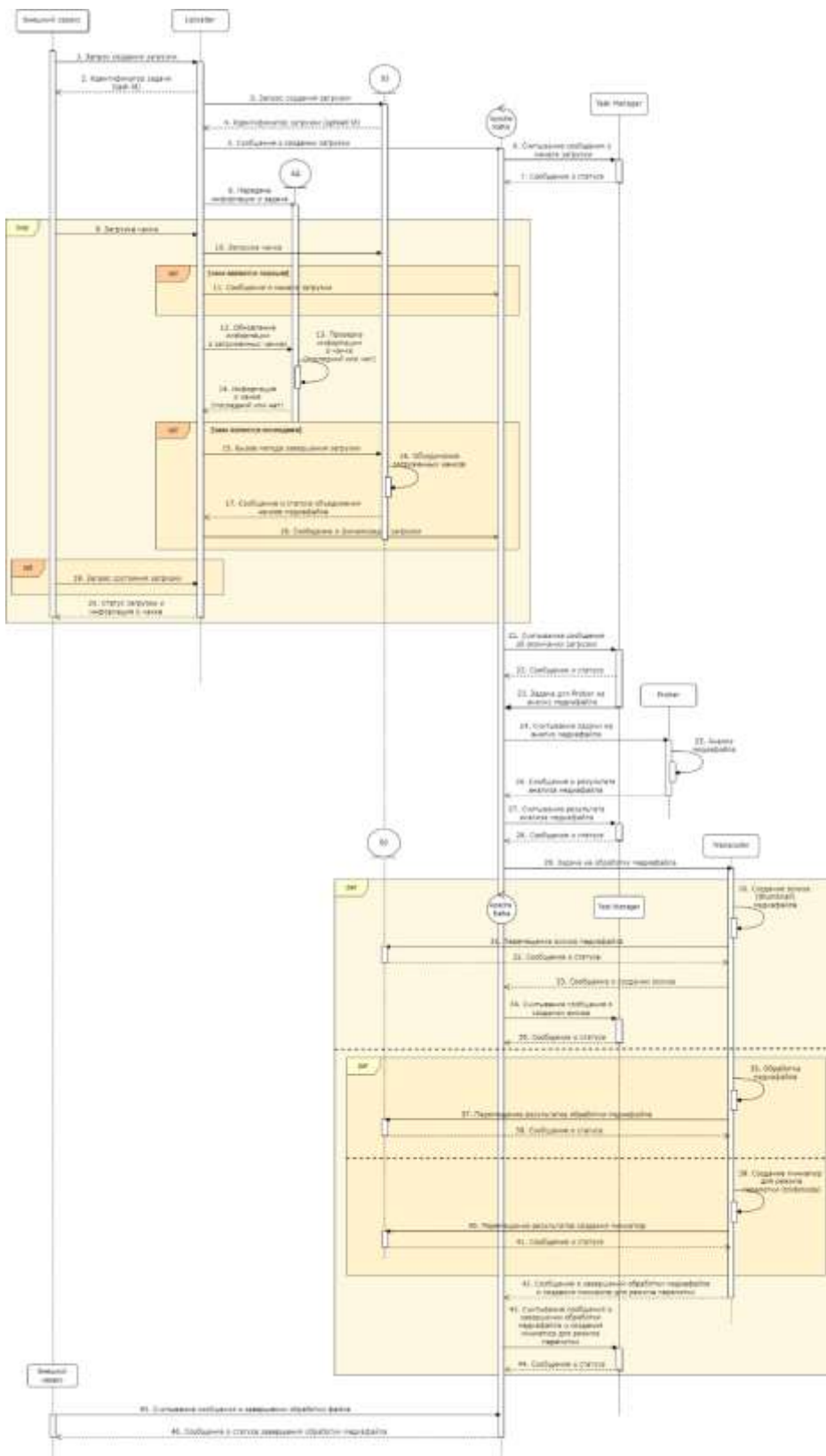
Для каждого графика существуют настройки диапазона времени, в течение которого собираются данные.

Название	Описание	Пример изображения
Относительный диапазон времени	<p>Пример относительного диапазонов времени:</p> <ul style="list-style-type: none"> • Последние 5 минут; • Последний 1 час; • Последние 2 дня; • Сегодня; • Вчера; • Позавчера. 	
Абсолютный диапазон времени	<p>Способы выбора времени:</p> <ul style="list-style-type: none"> • Пользователь вводит значения времени в формате YYYY-MM-DD HH:MM:SS, а затем нажимает кнопку "Apply time range" . • Grafana отображает календарь. Пользователь выбирает день или дни, которые хочет использовать в качестве текущего временного диапазона, а затем нажимает кнопку "Apply time range". <p>Grafana также отображает недавно использовавшиеся абсолютные диапазоны.</p>	

<p>Увеличение масштаба диапазона времени</p>	<p>Способ увеличения временного диапазона:</p> <ul style="list-style-type: none"> Пользователь выделяет правой кнопкой мыши нужный период времени на графике (например, 17:00), и период автоматически масштабируется до следующего временного периода. 	
<p>Уменьшение масштаба диапазона времени</p>	<p>Способ уменьшения временного диапазона:</p> <ul style="list-style-type: none"> Пользователь нажимает кнопку "уменьшить масштаб" в правом верхнем углу экрана, и период автоматически уменьшается до предыдущего временного периода. 	
<p>Цикл автоматического обновления метрик</p>	<p>Способ инициализации запросов и обновления данных информационной панели:</p> <ul style="list-style-type: none"> Пользователь нажимает на кнопку "обновить" в правом верхнем углу экрана. Происходит инициирование запросов и обновление данных информационной панели. 	

Настройки левой шкалы графика	Для каждого графика можно изменить опции отображения панели. Например, Grafana позволяет установить максимальное значение для показателей левой шкалы.		
	Шаги	Результат	Пример изображения
	Пользователь нажимает по названию графика и в раскрывающемся списке выбирает "Edit".	В левой части опций панели находятся поля min и max. Значения этих полей используются для расчета процентного порога.	
Пользователь изменяет значение в поле max с "auto" на значение "5" и нажимает кнопку "Apply"	После сохранения изменений у пользователя отображается 5 делений на левой шкале графика - 5 запросов на загрузку, не прошедших валидацию за период 60 секунд.		

4. Жизненный цикл обработки медиаданных



Описание схемы приведено в таблице ниже.

1	Из внешнего сервиса (front) в Uploader приходит запрос создания загрузки медиафайла.
2	Uploader формирует задачу и в ответе на запрос возвращает идентификатор задачи (task id).
3	Uploader отправляет запрос на создание загрузки в хранилище S3.
4	S3 в ответ на запрос возвращает в Uploader идентификатор загрузки (upload id).
5	После получения задачи на загрузку исходного медиафайла, формирования идентификатора задачи (task id) и получения идентификатора загрузки (upload id) компонент Uploader отправляет в топик KAFKA_TOPIC_TASK_MANAGER сообщение UPLOADER_TASK , в котором параметр message_type (тип сообщения) имеет значение UPLOADER_TASK_CREATE (создание задачи загрузки).
6	Task Manager подписан на топик KAFKA_TOPIC_TASK_MANAGER и считывает из Kafka сообщение о создании загрузки UPLOADER_TASK .
7	Компонент Task Manager отправляет в топик KAFKA_TOPIC_STATUS сообщение TASK_STATUS , в котором: <ul style="list-style-type: none"> • параметр message_type (тип сообщения) имеет значение TASK_STATUS (статус задачи); • параметра state (статус) имеет значение UploaderStarted (загрузка начата).
8	Uploader передает в БД информацию о задаче, включая: <ul style="list-style-type: none"> • метаданные медиафайла; • task id; • upload id; • общее количество чанков; • количество уже загруженных чанков.

9	<p>Из внешнего сервиса начинается загрузка в Uploader исходного медиафайла.</p> <p>При этом:</p> <ul style="list-style-type: none"> • можно загрузить медиафайл одним файлом - в этом случае происходит загрузка единственного чанка; • можно загрузить медиафайл несколькими частями - в этом случае происходит загрузка очередного чанка по порядку. <p>Шаги 9-20 повторяются для каждого чанка.</p>
10	<p>После загрузки чанка Uploader записывает его в хранилище S3 для дальнейшего анализа и обработки.</p>
11	<p>Только для первого чанка или для загрузки файла целиком (одним чанком):</p> <p>Uploader отправляет в топик KAFKA_TOPIC_TASK_MANAGER сообщение TRANSFER_TASK, в котором параметр message_type (тип сообщение) имеет значение TRANSFER_TASK_START (начало перемещения).</p> <p>После получения этого сообщения Task Manager не отправляет статус задачи в топик KAFKA_TOPIC_STATUS.</p>
12	<p>Uploader обновляет таблицу о загруженных чанках в БД, включая:</p> <ul style="list-style-type: none"> • описание чанка; • номер чанка; • размер чанка.
13	<p>В БД происходит проверка информации о загруженном чанке, является он последним для медиафайла или нет.</p>
14	<p>БД отправляет в Uploader информацию о чанке, является ли он последним для медиафайла.</p>
15	<p>REST API предоставляет возможность запросить сведения о состоянии загрузки (например, при загрузке медиафайлов блоками). Для этого используется запрос состояния загрузки</p> <p>Внешний сервис может запросить сведения о состоянии загрузки на любом этапе между шагами 9 и 20.</p>

16	<p>Только для последнего чанка:</p> <p>Если чанк является последним, Uploader вызывает в хранилище S3 метод завершения загрузки (если этот чанк единственный, метод все равно вызывается).</p>
17	<p>Только для последнего чанка:</p> <p>В хранилище происходит объединение загруженных чанков.</p>
18	<p>Только для последнего чанка:</p> <p>Хранилище S3 в ответ на вызов метода завершения загрузки отправляет в Uploader сообщение о статусе объединения чанков медиафайла.</p>
19	<p>Uploader отправляет в топик KAFKA_TOPIC_TASK_MANAGER сообщение TRANSFER_TASK, в котором параметр message_type (тип сообщения) имеет значение TRANSFER_TASK_FINISHED (перемещение завершено).</p> <p>После получения этого сообщения Task Manager не отправляет статус задачи в топик KAFKA_TOPIC_STATUS.</p>
20	<p>Uploader возвращает внешнему сервису ответ на запрос загрузки чанка, отправленный в п. 9:</p> <ul style="list-style-type: none"> • запрос загрузки медиафайла целиком; • запрос загрузки медиафайла частями. <p>Шаги 9-20 повторяются для каждого чанка.</p>
21	<p>Task Manager считывает сообщение из Kafka KAFKA_TOPIC_TASK_MANAGER сообщение TRANSFER_TASK об окончании загрузки медиафайла.</p>
22	<p>Считав сообщение о завершении загрузки, компонент Task Manager отправляет в топик KAFKA_TOPIC_STATUS сообщение TASK_STATUS, в котором:</p> <ul style="list-style-type: none"> • параметр message_type (тип сообщения) имеет значение TASK_STATUS; • параметр state (статус) может и меть одно из следующих значений: <ul style="list-style-type: none"> ○ UploaderFinished (загрузка завершена); ○ UploaderFailed (сбой загрузки).

23	Task Manager отправляет в топик KAFKA_TOPIC_PROBE сообщение PROBE_TASK , содержащее задачу на анализ загруженного медиафайла. В этом сообщении параметр message_type (тип сообщения) имеет значение PROBE_TASK (задача на анализ медиафайла).
24	Prober считывает сообщение от Task Manager о задаче на анализ медиафайла.
25	Prober анализирует исходный медиафайл.
26	После анализа (успешного или неуспешного) исходного медиафайла компонент Prober отправляет в топик KAFKA_TOPIC_TASK_MANAGER сообщение PROBE_TASK_FINISHED , в котором параметр message_type имеет значение PROBE_TASKFINISHED (анализ завершён).
27	Task Manager считывает сообщение из Kafka о результате анализа медиафайла.
28	<p>Получив сообщение об анализе медиафайла, Task Manager отправляет в топик KAFKA_TOPIC_STATUS сообщение TASK_STATUS, в котором:</p> <ul style="list-style-type: none"> • параметр message_type (тип сообщения) имеет значение TASK_STATUS; • параметр state (статус) может иметь одно из следующих значений: <ul style="list-style-type: none"> ○ ProbeFinished (анализ успешно завершён); ○ ProbeFailed (сбой анализа).

29	<p>На основе настроек из топика <code>KAFKA_TOPIC_TASKMANAGER_SETTINGS</code> Task Manager отправляет в топик <code>KAFKA_TOPIC_TRANSCODER</code> сообщение TRANSCODER_TASK, содержащее задачу на обработку загруженного медиафайла. В этом сообщении параметр <code>message_type</code> (тип сообщения) имеет значение <code>TRANSCODER_TASK</code>. Task Manager считывает сообщение с настройками при запуске. Для постановки задач на обработку используется самое новое сообщение. В топике <code>KAFKA_TOPIC_TASKMANAGER_SETTINGS</code> это сообщение TASKMANAGER_SETTINGS.</p> <p>При создании задания на транскодирование Task Manager копирует в сообщение <code>TRANSCODER_TASK</code> часть сообщения <code>TASKMANAGER_SETTINGS</code>.</p> <p>При получении задачи на обработку медиафайла Transcoder запускает три параллельных процесса:</p> <ul style="list-style-type: none"> • создание эскиза видео (thumbnail) (шаги 30-35); • транскодирование файла (шаги 36-38); • формирование трикмода (шаги 39-41).
30	<p>Transcoder создаёт эскиз или эскизы (thumbnail) медиафайла, если в настройках Task Manager содержится соответствующий профиль или профили.</p> <p>Обработка всех типов профилей:</p> <ul style="list-style-type: none"> • создание эскизов (thumbnail), • обработка медиафайлов (в т. ч. в разных качествах), • извлечение аудиодорожек, — <p>происходит параллельно.</p>
31	<p>Transcoder помещает созданный эскиз (thumbnail) в хранилище S3 в соответствии с установленными настройками.</p> <p>Для анализа и обработки исходных медиафайлов и для размещения результатов обработки может использоваться как один и тот же экземпляр хранилища S3, так и различные экземпляры. Какие хранилища будут использоваться, определяется настройками Медиаплатформы.</p>

32	Хранилище S3 передает сообщение о статусе в Transcoder.
33	<p>Если задание на обработку исходного медиафайла содержит профиль для создания эскиза (thumbnail) медиафайла, после создания (или сбоя при создании) такого эскиза Transcoder в топик KAFKA_TOPIC_TASK_MANAGER отправляет сообщение THUMBNAIL_INFO, в котором параметр message_type (тип сообщения) имеет значение THUMBNAIL_INFO (сведения об эскизе).</p> <p>Transcoder отправляет различные сообщения в зависимости от типа обработанных профилей:</p> <ul style="list-style-type: none"> • профили создания эскиза (thumbnail) — THUMBNAIL_INFO; • профили обработки медиафайла и/или извлечения аудиодорожки — TRANSCODER_TASK_FINISHED. <p>При этом сообщения THUMBNAIL_INFO отправляются по поводу каждого сформированного эскиза (thumbnail), тогда как сообщения TRANSCODER_TASK_FINISHED отправляются, когда обработаны все профили обработки медиафайлов (и/или извлечения аудиодорожки).</p>
34	Task Manager считывает сообщение из Kafka о результате создания эскизов.
35	<p>После получения сообщения о создании — успешном или неуспешном — эскиза (thumbnail), Task Manager отправляет в топик KAFKA_TOPIC_STATUS сообщение TASK_STATUS, в котором:</p> <ul style="list-style-type: none"> • параметр message_type (тип сообщения) имеет значение TASK_STATUS (статус задачи); • параметр state (статус) имеет одно из следующих значений: <ul style="list-style-type: none"> ○ ThumbnailReady (эскиз готов); ○ ThumbnailFailed (сбой создания эскиза).
36	Transcoder обрабатывает медиафайл.

37	<p>Transcoder помещает результат обработки в хранилище S3 в соответствии с установленными настройками.</p> <p>Для анализа и обработки исходных медиафайлов и для размещения результатов обработки может использоваться как один и тот же экземпляр хранилища S3, так и различные экземпляры. Какие хранилища будут использоваться, определяется настройкам и Медиаплатформы.</p>
38	<p>Хранилище S3 передает сообщение о статусе в Transcoder.</p>
39	<p>Transcoder создает миниатюры для режима перемотки (trickmode).</p>
40	<p>Transcoder помещает результат создания миниатюр для режима перемотки в хранилище S3.</p>
41	<p>Хранилище S3 передает сообщение о статусе в Transcoder.</p>
42	<p>После обработки — успешной или неуспешной — медиафайла Transcoder отправляет в топик KAFKA_TOPIC_TASK_MANAGER сообщение TRANSCODER_TASKFINISHED, в котором параметр message_type (тип сообщения) имеет значение TRANSCODER_TASKFINISHED (обработка завершена).</p> <p>Transcoder отправляет различные сообщения в зависимости от типа обработанных профилей:</p> <ul style="list-style-type: none"> • профили создания эскиза (thumbnail) — THUMBNAIL_INFO; • профили обработки медиафайла и/или извлечения аудиодорожки — TRANSCODER_TASK_FINISHED. <p>При этом сообщения THUMBNAIL_INFO отправляются по поводу каждого сформированного эскиза (thumbnail), тогда как сообщения TRANSCODER_TASK_FINISHED отправляются, когда обработаны все профили обработки медиафайлов (и/или извлечения аудиодорожки).</p>
43	<p>Task Manager считывает сообщение из Kafka о завершении обработки медиафайла и создания миниатюр для режима перемотки.</p>

44	<p>После получения сообщения об обработке медиафайла (и/или извлечения аудиодорожки) Task Manager отправляет в топик KAFKA_TOPIC_STATUS сообщение TASK_STATUS, в котором:</p> <ul style="list-style-type: none"> • параметр message_type (тип сообщения) имеет значение TASK_STATUS; • параметр state (статус) может иметь следующие значения: <ul style="list-style-type: none"> ○ TranscoderFinished (обработка завершена); ○ TranscoderFailed (сбой обработки). <p>Задание получает общий статус FAILED в случае ошибки при работе хотя бы с одним профилем. При этом выходные файлы по корректно обработанным профилям сохраняются и в итоговом сообщении присутствуют ссылки на эти файлы.</p>
45	<p>Внешний сервис считывает сообщение о завершении обработки медиафайла.</p>
46	<p>Kafka отправляет во внешний сервис сообщение о статусе завершения обработки медиафайла.</p>

4.1. Особенности работы Медиаплатформы

4.1.1. Информация о цветовом пространстве и диапазоне

Transcoder автоматически сохраняет информацию о цветовом пространстве и диапазоне входного файла, если такая информация в файле присутствует. Если информация отсутствует, то файл приводится к цветовому пространству bt.601, а диапазон – к mpeg_range, информация добавляется в метаданные выходного файла.

Transcoder поддерживает возможность отката способа транскодирования при невозможности инициализировать процесс транскодирования на GPU.

Чтобы задействовать эту возможность, необходимо в профиле транскодирования помимо секции gpu указать секцию cpu. Если такая резервная секция не указана, в случае сбоя инициализации GPU версии процесса транскодирования произойдет завершение обработки с ошибкой.

Для того, чтобы Transcoder начал инициализацию процесса транскодирования на GPU, транскодер должен быть запущен с переменной среды окружения FORCE_GPU, которая должна содержать индекс GPU устройства, доступного в конкретном месте деплоя Transcoder. Обычно индексирование GPU устройств начинается с нуля.

Аварийное отступление может произойти как для всего процесса транскодирования, так и частично, например, когда не получается декодировать материал на GPU, но его можно кодировать на CPU. В этом случае на CPU будут выполняться подзадачи декодирования и трансформации (масштабирование, поворот и т.п.), а кодирование в выходной формат будет осуществляться на GPU.

Срабатывание алгоритма аварийного отступления можно отследить по логам, которые в случае невозможности использования компонентов GPU для выполнения тех или иных подзадач, будут содержать сообщения уровня WARNING, например:

- Falling back to CPU with profile 'base_cpu' after error of initialization of GPU pipeline for profile 'base_gpu' with error: 'error message from GPU initializer'
- Failed to init GPU decoder thread for video stream #0. Falling back to CPU.

5. REST API

5.1. Создание задачи на загрузку и обработку медиафайлов

Чтобы инициировать загрузку и обработку исходного видео, необходимо создать соответствующую задачу в Медиаплатформе и присвоить ей уникальный идентификатор. Для этого используется POST-запрос создания задачи:

POST <сервер медиаплатформы>/v1

Параметры запроса

Параметр	Тип данных	Обязательно	Описание
checksum	String	Нет	Контрольная сумма MD5 загружаемого файла.
metadata	Struct	Нет	Блок дополнительных параметров. Каждый дополнительный параметр передаётся следующим образом. <pre>metadata: { "<дополнительный параметр_1>": {"<название дополнительного параметра_1>": "<значение дополнительного параметра_1>"}, "<дополнительный параметр_2>": {"<название дополнительного параметра_2>": "<значение дополнительного параметра_2>" } }</pre>
metadata.<дополнительные_параметры>	Struct	Нет	Блок дополнительных параметров в формате "название": "значение", например: "additionalProp1": {"name1": "value1"}, "additionalProp2": {"name2": "value2"} и т. п.
contentType	String	Нет	MIME-тип файла.
size	Integer	Да	Размер файла в байтах.
parts	Integer	Да	Количество ожидаемых частей медиафайла.
partsize	Integer	Да	Размер одной части.
uid	String	Нет	Уникальный идентификатор пользователя. Присваивается Медиаплатформой.

Пример запроса

```
{
  "checksum": "md5 9815f1499c22f1e8f300793c2079bed5",
  "metadata": {
    "originalFilename": "example.mp4"
  },
  "mimeType": "application/octet-stream",
  "size": 13879433,
  "parts":3,
  "partsize":5242880,"
  "uid": "813265a6-f674-11ec-bf7d-6a3d6058d262"
}
```

Параметры ответа

Параметр	Тип данных	Обязательно	Описание
id	String	Да, если запрос обработан без ошибок	Идентификатор задачи, созданный после запроса создания задачи загрузки и обработки видео.
size	Integer	Да, если запрос обработан без ошибок	Размер файла в байтах.
mimeType	String	Нет	MIME-тип файла.
uploadRanges	Array	Да, если запрос обработан без ошибок	Пронумерованный массив блоков загрузки видео (если видео будет загружаться не целиком, а по частям), пронумерованный, начиная с 0 (т. е. 0 — первый блок загружаемого видео).
state	String	Только если при обработке запроса возникла ошибка	Состояние запроса. Возвращается, только если запрос обработан с ошибкой. Имеет значение <code>error</code> .
error	String	Нет	Описание ошибки (на английском языке). Возвращается, только если в процессе обработки запроса возникла ошибка.

Пример успешного ответа

```
{
  "id": "NhMqtQipLSJPA6MYrcy7y",
  "size": 13879433,
  "mimeType": "application/octet-stream",
  "uploadRanges": [
```



```
[
  0,
  13879433
]
],
}
```

Пример ответа с ошибкой

```
{
  "status": "error",
  "error": "size is empty"
}
```

5.2. Загрузка медиафайла одним файлом

Запрос загрузки видео целиком отправляется методом POST

POST <сервер_медиаплатформы>/v1/<id>/file

где:

- <сервер_медиаплатформы> — адрес медиаплатформы, например: <https://test.ru>;
- <id> — идентификатор задачи на загрузку и обработку видео, созданный ранее (например: 8be68cfd-800a-469c-b319-015cf96ee5b3).

Пример ссылки:

<https://test.ru/v1/8be68cfd-800a-469c-b319-015cf96ee5b3/file>

При этом тело (body) в запросе пустое.

Параметры ответа

Параметр	Тип	Обязательно	Описание
checksumStatus	Integer	Нет	Результат проверки контрольной суммы, которая была получена в запросе на создание задачи загрузки и обработки видео. Возможные значения: <ul style="list-style-type: none"> • если значение больше нуля, проверка успешна; • если значение меньше нуля, проверка неуспешна.
error	String	Только если при обработке запроса возникла ошибка	Описание ошибки (на английском языке). Возвращается, только если в процессе обработки запроса возникла ошибка.
n	Integer	Да	Объём загруженного в байтах.
status	String	Да	Статус выполнения загрузки:

			<ul style="list-style-type: none"> • ok (успешно); • error (ошибка); • finished (завершено).
t	Integer	Да	Временная метка загрузки видео (затраченное время на обработку запроса) в формате UNIX time (UNIX-время), т. е. количество секунд, прошедших с полуночи (00:00:00 UTC) 1 января 1970 года.

Пример успешного ответа

```
{
  "status": "ok",
  "n": 444,
  "t": 9548933
}
```

Пример ответа с ошибкой

```
{
  "status": "error",
  "error": "record not found"
}
```

5.3. Загрузка медиафайлов блоками

Запрос загрузки видео блоками отправляется методом PUT:

PUT <сервер_медиаплатформы>/v1/<id>/<offset>

где:

- <сервер_медиаплатформы> — адрес медиаплатформы, например: <https://test.ru>;
- <id> — идентификатор задачи на загрузку и обработку видео, созданный ранее (например: 8be68cfd-800a-469c-b319-015cf96ee5b3);
- <offset> — номер блока загружаемого видео, начиная с 0 (т. е., 0 — это первый блок, 1 — второй блок и т. п.).

То есть, для каждого блока загружаемого видео отправляется отдельный запрос.

Пример запроса:

<https://test.ru/v1/8be68cfd-800a-469c-b319-015cf96ee5b3/3>

В теле запроса передаётся содержимое загружаемого блока видео в кодировке Base64.

Пример тела запроса

5.4. Запрос состояния загрузки

Запрос статуса загрузки видео отправляется методом GET:

GET <сервер_медиаплатформы>/v1/<id>

где:

- <сервер_медиаплатформы> — адрес медиаплатформы, например: <https://test.ru>;
- <id> — идентификатор задачи на загрузку и обработку видео, созданный ранее (например: 8be68cfd-800a-469c-b319-015cf96ee5b3).

При этом тело (body) в запросе пустое.

5.4.1.1. Параметры ответа

Параметр	Тип данных	Обязательно	Описание
checksum	String	Нет	Контрольная сумма MD5 загружаемого файла.
createdAt	Integer	Да	Временная метка создания задачи в формате UNIX time (UNIX-время), т. е. количество секунд, прошедших с полуночи (00:00:00 UTC) 1 января 1970 года.
id	String	Да	Идентификатор задачи, созданный после запроса создания задачи загрузки и обработки видео.
isFinished	Boolean	Нет	Завершена ли загрузка видео. Возможные варианты: <ul style="list-style-type: none">• true - загрузка завершена;• false - загрузка не завершена.
metadata	Struct	Нет	Блок дополнительных параметров. Каждый дополнительный параметр передаётся следующим образом. <pre>metadata: { "<дополнительный параметр_1>": {"<название дополнительного параметра_1>": "<значение дополнительного параметра_1>"}, "<дополнительный параметр_2>": {"<название дополнительного параметра_2>": "<значение дополнительного параметра_2>"} }</pre>
mimeType	String	Нет	MIME-тип файла.
resource	Struct	Нет	Блок данных о загружаемом видео.
size	Integer	Да	Размер файла в байтах.
ua	String	Нет	Строка описания клиентского приложения, например: "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.80 Safari/537.36".

uid	String	Нет	Уникальный идентификатор пользователя.
uploadRanges	Array	Да	Пронумерованный массив блоков загрузки видео (если видео будет загружаться не целиком, а по частям), пронумерованный, начиная с 0 (т. е. 0 — первый блок загружаемого видео).
state	String	Нет	Состояние запроса. Возвращается, только если запрос обработан с ошибкой. Имеет значение <code>error</code> .
error	String	Только если при обработке запроса возникла ошибка	Описание ошибки (на английском языке). Возвращается, только если в процессе обработки запроса возникла ошибка.

Пример успешного ответа

```
{
  "id": "NhMqtQipLSJPA6MYrcy7y",
  "size": 13879433,
  "mimeType": "application/octet-stream",
  "metadata": {
    "originalFilename": "example.mp4"
  },
  "uploadRanges": [
    [
      0,
      13879433
    ]
  ],
  "checksum": "md5 9815f1499c22f1e8f300793c2079bed5",
  "createdAt": 1657248664,
  "resource": {
    "size": 13879433
  }
}
```

Пример ответа с ошибкой

```
{
  "status": "error",
```

```
"error": "record not found"  
}
```

6. Взаимодействие между компонентами Медиаплатформы через Apache Kafka

6.1. Настройки профилей обработки исходного медиафайла

6.1.1. Настройка профилей транскодирования

Для настройки профилей транскодирования необходимо поместить сообщение в топик `taskmanager.settings`. При запуске `TaskManager` 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`.

6.1.2. Транскодирование видеофайлов в зависимости от типа глубины цвета с использованием CPU и GPU

Видео с 8-битной глубиной цвета использует 8 бит (1 байт) информации для представления каждого цвета пикселя. Это означает, что каждый цветовой канал (красный, зеленый, синий) может принимать значения от 0 до 255. В результате возможно представить 16,7 миллиона (256^3) возможных цветовых комбинаций. Видео с 10-битной глубиной цвета использует 10 бит информации для представления каждого цвета пикселя. Это означает, что каждый цветовой канал может принимать значения от 0 до 1023. В результате этого возможно представить более чем 1 миллиард (1024^3) возможных цветовых комбинаций. Однако, использование 10-битного видео требует большего объема памяти и вычислительных ресурсов. При получении Медиаплатформой HDR видео с глубиной цвета 10 бит модуль `Prober` осуществляет проверку соответствия ограничениям.

Между CPU и GPU транскодерами задачи распределяются в зависимости от загрузки транскодеров с учетом глубины цвета видео.

8-битные видео транскодируются с использованием CPU и GPU: секции `cpi` и `gpi` внутри секции `profiles`. Для таких видео формируется только один плейлист 8 бит: параметр `profile_type`.

10-битные видео в Медиаплатформе транскодируются только с использованием декодеров и фильтров CPU: в секции `cpi` внутри секции `profiles` должен быть установлен параметр `pix_bit_depth = 10`. Для таких видео формируется два плейлиста: 8 бит и 10 бит, параметр `profile_type`. При этом транскодирование будет происходить на GPU. Если

инициализировать GPU версию этого профиля не удастся, произойдет срабатывание алгоритма аварийного отступления в CPU версию профиля.

В случае, если для профиля установлено транскодирование ВСЕХ найденных в источнике видеопотоков, произойдет фильтрация этих видеопотоков, и в результирующий файл будут включены (транскодированы) только потоки обладающие признаком HDR (глубина пикселя > 8 бит). Для всех остальных потоков в логи будет выдано предупреждение уровня WARNING:

```
"Skipping video stream #1 for profile 'best_hdr' due to source video pixel bits '8' < profile target pixel bits '10'."
```

6.1.3. Формат сообщения

Формат сообщения настройки Task Manager

```
{  
  "message_type": "TASKMANAGER_SETTINGS",  
  
  "ffmpeg_decoding_options": {  
    "threads": "1"  
  },  
  "master_playlist": true,  
  "hw_device": "cuda",  
  
  "profiles": [  
    {  
      "name": "outstanding_2160p_8bit",  
      "container_type": "hls",  
      "filename_extension": "m3u8",  
      "pix_bit_depth": 8,  
      "video": {  
        "stream_selection": "best",  
        "codec": "libx264",  
        "filter_options": {  
          "width": 3840,  
          "height": 2160,  

```



```
"fps": {
  "num": 30,
  "den": 1
},
"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": "16M",
  "bufsize": "8M"
}
},
"audio": {
  "stream_selection": "best",
  "codec": "aac",
  "filter_options": {
    "channels": 2,
    "samplerate": 48000
  },
  "ffmpeg_encoding_options": {
    "b": "256k"
  }
},
"cpu": {
```

```
"name": "outstanding_2160p_8bit",
"container_type": "hls",
"filename_extension": "m3u8",
"pix_bit_depth": 8,
"video": {
  "stream_selection": "best",
  "codec": "libx264",
  "filter_options": {
    "width": 3840,
    "height": 2160,
    "fps": {
      "num": 30,
      "den": 1
    },
    "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": "16M",
    "bufsize": "8M"
  }
},
"audio": {
  "stream_selection": "best",
```

```

    "codec": "aac",
    "filter_options": {
      "channels": 2,
      "samplerate": 48000
    },
    "ffmpeg_encoding_options": {
      "b": "256k"
    }
  },
  "gpu": {
    "name": "outstanding_2160p_8bit",
    "container_type": "hls",
    "filename_extension": "m3u8",
    "pix_bit_depth": 8,
    "video": {
      "stream_selection": "best",
      "codec": "h264_nvenc",
      "filter_options": {
        "width": 3840,
        "height": 2160,
        "fps": {
          "num": 30,
          "den": 1
        },
      },
      "overlay": false,
      "padding": false,
      "rotate": true,
      "allow_swap_dimensions": true
    },
    "ffmpeg_encoding_options": {
      "g": "60",

```

```

        "threads": "1",
        "profile": "main",
        "preset": "p1",
        "b": "16777216",
        "bufsize": "16777216",
        "maxrate": "16777216",
        "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"
    }
}
},
{
    "name": "high_1080p_8bit",
    "container_type": "hls",
    "filename_extension": "m3u8",
    "pix_bit_depth": 8,
    "video": {
        "stream_selection": "best",

```

```
"codec": "libx264",
"filter_options": {
  "width": 1920,
  "height": 1080,
  "fps": {
    "num": 30,
    "den": 1
  },
  "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_1080p_8bit",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 8,
  "video": {
    "stream_selection": "best",
    "codec": "libx264",
    "filter_options": {
      "width": 1920,
      "height": 1080,
      "fps": {
        "num": 30,
        "den": 1
      },
    },
    "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_1080p_8bit",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 8,
  "video": {
    "stream_selection": "best",
    "codec": "h264_nvenc",
    "filter_options": {
      "width": 1920,
      "height": 1080,
      "fps": {
        "num": 30,
        "den": 1
      },
    },
    "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"
    }
}
},
{
    "name": "medium_720p_8bit",
    "container_type": "hls",

```



```
"filename_extension": "m3u8",
"pix_bit_depth": 8,
"video": {
  "stream_selection": "best",
  "codec": "libx264",
  "filter_options": {
    "width": 1280,
    "height": 720,
    "fps": {
      "num": 30,
      "den": 1
    },
    "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": "5M",
    "bufsize": "2.5M"
  }
},
"audio": {
  "stream_selection": "best",
  "codec": "aac",
  "filter_options": {
```

```
    "channels": 2,
    "samplerate": 48000
  },
  "ffmpeg_encoding_options": {
    "b": "256k"
  }
},
"cpu": {
  "name": "medium_720p_8bit",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 8,
  "video": {
    "stream_selection": "best",
    "codec": "libx264",
    "filter_options": {
      "width": 1280,
      "height": 720,
      "fps": {
        "num": 30,
        "den": 1
      },
    },
    "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": "5M",
        "bufsize": "2.5M"
    }
},
"audio": {
    "stream_selection": "best",
    "codec": "aac",
    "filter_options": {
        "channels": 2,
        "samplerate": 48000
    },
    "ffmpeg_encoding_options": {
        "b": "256k"
    }
}
},
"gpu": {
    "name": "medium_720p_8bit",
    "container_type": "hls",
    "filename_extension": "m3u8",
    "pix_bit_depth": 8,
    "video": {
        "stream_selection": "best",
        "codec": "h264_nvenc",
        "filter_options": {
            "width": 1280,
            "height": 720,
            "fps": {
                "num": 30,
                "den": 1
            }
        }
    }
}
}

```

```

    },
    "overlay": false,
    "padding": false,
    "rotate": true,
    "allow_swap_dimensions": true
  },
  "ffmpeg_encoding_options": {
    "g": "60",
    "threads": "1",
    "profile": "main",
    "preset": "p1",
    "b": "5242880",
    "bufsize": "5242880",
    "maxrate": "5242880",
    "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"
  }
}
}

```

```

},
{
  "name": "low_480p_8bit",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 8,
  "video": {
    "stream_selection": "best",
    "codec": "libx264",
    "filter_options": {
      "width": 854,
      "height": 480,
      "fps": {
        "num": 30,
        "den": 1
      },
    },
    "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": "2.5M",
    "bufsize": "1.75M"
  }
},

```

```
"audio": {
  "stream_selection": "best",
  "codec": "aac",
  "filter_options": {
    "channels": 2,
    "samplerate": 48000
  },
  "ffmpeg_encoding_options": {
    "b": "256k"
  }
},
"cpu": {
  "name": "low_480p_8bit",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 8,
  "video": {
    "stream_selection": "best",
    "codec": "libx264",
    "filter_options": {
      "width": 854,
      "height": 480,
      "fps": {
        "num": 30,
        "den": 1
      },
      "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": "2.5M",
  "bufsize": "1.75M"
}
},
"audio": {
  "stream_selection": "best",
  "codec": "aac",
  "filter_options": {
    "channels": 2,
    "samplerate": 48000
  },
  "ffmpeg_encoding_options": {
    "b": "256k"
  }
}
},
"gpu": {
  "name": "low_480p_8bit",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 8,
  "video": {
    "stream_selection": "best",
    "codec": "h264_nvenc",
    "filter_options": {
      "width": 854,
```

```
    "height": 480,
    "fps": {
      "num": 30,
      "den": 1
    },
    "overlay": false,
    "padding": false,
    "rotate": true,
    "allow_swap_dimensions": true
  },
  "ffmpeg_encoding_options": {
    "g": "60",
    "threads": "1",
    "profile": "main",
    "preset": "p1",
    "b": "2621440",
    "bufsize": "2621440",
    "maxrate": "2621440",
    "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"
    }
}
},
{
    "name": "outstanding_2160p_10bit",
    "container_type": "hls",
    "filename_extension": "m3u8",
    "pix_bit_depth": 10,
    "video": {
        "stream_selection": "best",
        "codec": "libx264",
        "filter_options": {
            "width": 3840,
            "height": 2160,
            "fps": {
                "num": 30,
                "den": 1
            },
            "overlay": false,
            "padding": false,
            "rotate": true,
            "allow_swap_dimensions": true,
            "pix_fmt": "yuv420p10le"
        },
        "ffmpeg_encoding_options": {
            "g": "60",
            "threads": "4",
            "profile": "high10",
            "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": "outstanding_2160p_10bit",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 10,
  "video": {
    "stream_selection": "best",
    "codec": "libx264",
    "filter_options": {
      "width": 3840,
      "height": 2160,
      "fps": {
        "num": 30,
        "den": 1
      },
    },
    "overlay": false,
    "padding": false,
```

```

        "rotate": true,
        "allow_swap_dimensions": true,
        "pix_fmt": "yuv420p10le"
    },
    "ffmpeg_encoding_options": {
        "g": "60",
        "threads": "4",
        "profile": "high10",
        "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"
    }
}
},
{
    "name": "high_1080p_10bit",
    "container_type": "hls",
    "filename_extension": "m3u8",
    "pix_bit_depth": 10,

```

```
"video": {
  "stream_selection": "best",
  "codec": "libx264",
  "filter_options": {
    "width": 1920,
    "height": 1080,
    "fps": {
      "num": 30,
      "den": 1
    },
    "overlay": false,
    "padding": false,
    "rotate": true,
    "allow_swap_dimensions": true,
    "pix_fmt": "yuv420p10le"
  },
  "ffmpeg_encoding_options": {
    "g": "60",
    "threads": "4",
    "profile": "high10",
    "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_1080p_10bit",
    "container_type": "hls",
    "filename_extension": "m3u8",
    "pix_bit_depth": 10,
    "video": {
      "stream_selection": "best",
      "codec": "libx264",
      "filter_options": {
        "width": 1920,
        "height": 1080,
        "fps": {
          "num": 30,
          "den": 1
        },
      },
      "overlay": false,
      "padding": false,
      "rotate": true,
      "allow_swap_dimensions": true,
      "pix_fmt": "yuv420p10le"
    },
    "ffmpeg_encoding_options": {
      "g": "60",
      "threads": "4",
      "profile": "high10",
      "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"
    }
}
},
{
    "name": "medium_720p_10bit",
    "container_type": "hls",
    "filename_extension": "m3u8",
    "pix_bit_depth": 10,
    "video": {
        "stream_selection": "best",
        "codec": "libx264",
        "filter_options": {
            "width": 1280,
            "height": 720,
            "fps": {
                "num": 30,
                "den": 1
            }
        },
    },

```

```
    "overlay": false,
    "padding": false,
    "rotate": true,
    "allow_swap_dimensions": true,
    "pix_fmt": "yuv420p10le"
  },
  "ffmpeg_encoding_options": {
    "g": "60",
    "threads": "4",
    "profile": "high10",
    "preset": "superfast",
    "crf": "23",
    "maxrate": "5M",
    "bufsize": "2.5M"
  }
},
"audio": {
  "stream_selection": "best",
  "codec": "aac",
  "filter_options": {
    "channels": 2,
    "samplerate": 48000
  },
  "ffmpeg_encoding_options": {
    "b": "256k"
  }
},
"cpu": {
  "name": "medium_720p_10bit",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 10,
```

```
"video": {
  "stream_selection": "best",
  "codec": "libx264",
  "filter_options": {
    "width": 1280,
    "height": 720,
    "fps": {
      "num": 30,
      "den": 1
    },
    "overlay": false,
    "padding": false,
    "rotate": true,
    "allow_swap_dimensions": true,
    "pix_fmt": "yuv420p10le"
  },
  "ffmpeg_encoding_options": {
    "g": "60",
    "threads": "4",
    "profile": "high10",
    "preset": "superfast",
    "crf": "23",
    "maxrate": "5M",
    "bufsize": "2.5M"
  }
},
"audio": {
  "stream_selection": "best",
  "codec": "aac",
  "filter_options": {
    "channels": 2,
    "samplerate": 48000
  }
}
```



```

    },
    "ffmpeg_encoding_options": {
        "b": "256k"
    }
}
},
{
    "name": "low_480p_10bit",
    "container_type": "hls",
    "filename_extension": "m3u8",
    "pix_bit_depth": 10,
    "video": {
        "stream_selection": "best",
        "codec": "libx264",
        "filter_options": {
            "width": 854,
            "height": 480,
            "fps": {
                "num": 30,
                "den": 1
            },
            "overlay": false,
            "padding": false,
            "rotate": true,
            "allow_swap_dimensions": true,
            "pix_fmt": "yuv420p10le"
        },
        "ffmpeg_encoding_options": {
            "g": "60",
            "threads": "4",
            "profile": "high10",

```

```
    "preset": "superfast",
    "crf": "23",
    "maxrate": "2.5M",
    "bufsize": "1.75M"
  }
},
"audio": {
  "stream_selection": "best",
  "codec": "aac",
  "filter_options": {
    "channels": 2,
    "samplerate": 48000
  },
  "ffmpeg_encoding_options": {
    "b": "256k"
  }
},
"cpu": {
  "name": "low_480p_10bit",
  "container_type": "hls",
  "filename_extension": "m3u8",
  "pix_bit_depth": 10,
  "video": {
    "stream_selection": "best",
    "codec": "libx264",
    "filter_options": {
      "width": 854,
      "height": 480,
      "fps": {
        "num": 30,
        "den": 1
      }
    },
  },
}
```

```

    "overlay": false,
    "padding": false,
    "rotate": true,
    "allow_swap_dimensions": true,
    "pix_fmt": "yuv420p10le"
  },
  "ffmpeg_encoding_options": {
    "g": "60",
    "threads": "4",
    "profile": "high10",
    "preset": "superfast",
    "crf": "23",
    "maxrate": "2.5M",
    "bufsize": "1.75M"
  }
},
"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
}

```

Таблица с описанием параметров

Параметр	Тип данных	Формат	Диапазон возможных значений	Обязательно	Описание
<code>message_type</code>	String		TASKMANAGER_SETTINGS	Да	Тип сообщения. Может принимать следующие значения:

					<ul style="list-style-type: none"> TASKMANAGER_SE TTINGS – настройки компонента Task Manager.
ffmpeg_decoding_options	Struct	Группа параметров		Нет	<p>Блок настроек FFmpeg, передаваемых напрямую в декодер в формате "строка":"строка". Полный список возможных опций можно посмотреть здесь (внешний источник на английском языке).</p> <p>Например: threads - число потоков, которое может использовать декодер, если он поддерживает многопоточность. Возможные значения:</p> <ul style="list-style-type: none"> auto или 0 для автоматического определения оптимального числа потоков; явно указанное число. <p>Значение по умолчанию: auto.</p>
ffmpeg_decoding_options.threads	String		<ul style="list-style-type: none"> auto [положительное целое число] 	Нет	<p>Число потоков, которое может использовать декодер, если он поддерживает многопоточность. Возможные значения:</p> <ul style="list-style-type: none"> auto или 0 для автоматического определения оптимального числа потоков; явно указанное число. <p>Значение по умолчанию: auto.</p>
master_playlist	Boolean	Флаг	<ol style="list-style-type: none"> true false 	Да	<p>Включение или отключение генерации общего мастер-плейлиста HLS для всех профилей, у которых параметр container_type имеет значение hls. Доступные значения:</p>

					<ul style="list-style-type: none"> • true - настройка включена; • false - настройка выключена.
hw_device	String		cuda	Нет	<p>Тип используемого аппаратного ускорения для обработки видео. Допустимое значение: cuda.</p> <p>Поддерживаются карты NVIDIA, драйверы которых совместимы с CUDA 11.6.</p> <p>Кроме того, аппаратное ускорение поддерживается только в режиме, когда декодирование и фильтрация (scaling) и кодирование видео выполняются на GPU. Т. е. частичная обработка на GPU в настоящее время недоступна. Это ограничивает возможный набор кодеков:</p> <ul style="list-style-type: none"> • декодирование: <ul style="list-style-type: none"> ○ mpeg2; ○ av1; ○ h264; ○ h265/hevc,; ○ mjpeg; ○ mpeg4; ○ vc1; ○ vp8; ○ vp9. • кодирование: <ul style="list-style-type: none"> ○ h264_nvenc; ○ hevc_nvenc. <p>Аппаратное ускорение не применяется для обработки аудио и создания эскизов видео (thumbnail).</p>
profiles	Struct	Группа параметров		Да	<p>Блок, содержащий профиль или профили обработки медиаданных. Профили могут быть следующих видов:</p> <ul style="list-style-type: none"> • профиль обработки аудио и видео — название профиля может быть произвольным, профиль должен

					<p>содержать вложенные блоки video и audio — для обработки аудио и видео соответственно; Блоки video и audio можно скомбинировать в одном профиле (видео с аудио) или в разные (видео без аудио или аудио без видео), тогда как блоки thumbnail и trickmode нельзя комбинировать с другими блоками в рамках одного профиля.</p> <ul style="list-style-type: none"> • профиль создания эскиза (thumbnail) - всегда передаётся в блоке thumbnail; • профиль создания холста миниатюр для ознакомительного просмотра - всегда передаётся в блоке trickmode.
profiles.name	String		Произвольное значение	Да	Название профиля обработки медиаданных. Произвольное значение.
profiles.container_type	String		Список поддерживаемых текущей версией FFmpeg контейнеров можно командой ffmpeg -formats	Да	<p>Итоговый формат файла в рамках обозначений FFmpeg:</p> <ul style="list-style-type: none"> • профиль обработки видео - видеоформат, например, mp4; • профиль обработки (извлечения) аудио - аудиоформат, например, mp3; • профиль создания эскиза (thumbnail) - формат изображения, например, jpeg. <p>Получить список поддерживаемых текущей версией FFmpeg контейнеров можно командой ffmpeg -formats.</p>

					Параметр должен быть указан для каждого профиля.
profiles.filename_extension	String	Расширение файла		Нет	Имя расширения итогового файла. Используется при подстановке в шаблоне filename_template.
profiles.video	Struct	Группа параметров		Нет	Блок выходных параметров обрабатываемого видео.
profiles.video.stream_selection	String		<ol style="list-style-type: none"> 1. all 2. best 3. none 	Да	<p>Способ выбора видео- или аудиодорожки или дорожек для обработки и записи в итоговый файл:</p> <ul style="list-style-type: none"> • для видео - параметры выбора видеодорожки; • для аудио - параметры выбора аудиодорожки. <p>Доступны следующие значения:</p> <ul style="list-style-type: none"> • all - все дорожки этого типа; • best - одна «лучшая» дорожка в соответствии с критериями FFmpeg; • none - дорожки этого типа не обрабатываются (например: может быть использовано для профилей генерации аудио файла).
profiles.video.codec	String			Да	<p>Название кодека в формате обозначений FFmpeg. Тип кодека зависит от того, к какому блоку относится параметр:</p> <ul style="list-style-type: none"> • если параметр к обработке видео - видеокодек; • если параметр относится к обработке аудио - аудиокодек.

					Получить список поддерживаемых текущей версией FFmpeg кодеков можно командой <code>ffmpeg – encoders</code> .
profiles.video.filter_options	String	Группа параметров		Да	Блок параметров преобразования: <ul style="list-style-type: none"> • для блока video - преобразование видеодорожки; • для блока audio - преобразование аудиодорожки.
profiles.video.filter_options.width	Integer			Да	Ширина видео в пикселях.
profiles.video.filter_options.height	Integer			Да	Высота видео в пикселях.
profiles.video.filter_options.fps	Struct	Группа параметров		Нет	Блок настроек FPS (Frames per second - кадры в секунду), определяется соотношением значений параметров num и den, входящих в блок. Итоговым значением блока после обработки будет значение соотношения num/den.
profiles.video.filter_options.fps.num	Integer	Положительное число		Да	Число кадров в единицу времени (количество секунд указано в параметре den).
profiles.video.filter_options.fps.den	Integer	Положительное число		Да	Количество секунд, в течение которых отображается количество кадров, определённых в параметре num.
profiles.video.filter_options.pix_fmt	String		Возможные значения можно получить из ffmpeg с помощью команды "ffmpeg - pix_fmts", например "yuv420p"	Нет	Формат пикселей видео. Итоговое видео будет преобразовано в указанный пиксель формат. Пустая строка означает автоматический подбор формата пикселей на основе формата пикселей исходного видео и возможностях кодека.

					Значение по умолчанию: пустая строка.
profiles.video.filter_options.padding	String	Флаг	true, false	Нет	Если true, части видео, которые образовались из-за разного соотношения сторон кадра на входе и выходе, заполняются черным цветом. Значение по умолчанию - false.
profiles.video.filter_options.rotate	String	Флаг	true, false	Нет	Если true, то при наличии в метаданных исходного видео значения rotate, отличного от нуля, делается поворот кадра. Значение по умолчанию - false.
profiles.video.filter_options.allow_swap_dimensions	String	Флаг	true, false	Нет	Если true, то разрешается сделать замену width <-> height, если это оптимальнее для входной картинки. Значение по умолчанию - true.
profiles.video.filter_options.overlay	String	Флаг	true, false	Нет	Если true, то видео, указанное в секции overlay (далее – водяной знак), будет наложено поверх основного видео профиля от левого угла. Предварительно, видео будет масштабировано по размеру, указанному для профиля, с сохранением оригинальных пропорций. Значение по умолчанию - false.
profiles.video.ffmpeg_encoding_options	Struct	Группа параметров		Нет	Блок настроек FFmpeg, передаваемых напрямую в кодировщик в формате "строка":"строка". Полный список возможных опций можно посмотреть по ссылке https://ffmpeg.org/ffmpeg-codecs.html (на английском языке). Ниже представлены примеры опций. <ul style="list-style-type: none"> • preset - имя пресета при кодировании видео в формате h.264. Для кодировщика libx264

					<p>список возможных значений: fast, superfast, ultrafast и т. п. Полный список значений можно посмотреть по адресу https://trac.ffmpeg.org/wiki/Encode/H.264 (на английском языке).</p> <p>Для кодировщика h264_nvenc (при работе с GPU) список возможных значений отличается: slow, medium, fast, hp, hq и т. п. Полный список значений можно посмотреть через командную строку: <code>ffmpeg -h encoder=h264_nvenc</code>.</p> <ul style="list-style-type: none"> • b - значение фиксированного битрейта видео. Может быть выражено в бит/с, так и сокращением, например: 2М, 64К. В случае кодека libx264 число означает не бит/с, а Кбит/с. • threads - число потоков, которое может использовать кодировщик, если он поддерживает многопоточность. Возможные значения auto или 0 для автоматического определения оптимального числа потоков либо конкретное число. <p>Значение по умолчанию: auto.</p>
--	--	--	--	--	--

profiles.video.ffmpeg_encoding_options.g	Integer			Нет	<p>Размещение ключевого кадра. Размер GOP устанавливает максимальное расстояние между ключевыми кадрами; если ноль, выходной поток будет только внутренним. Минимальное расстояние игнорируется, если оно не совпадает с размером GOP, и в этом случае ключевые кадры всегда будут появляться с фиксированным интервалом.</p> <p>Значение по умолчанию: не задано.</p>
profiles.video.ffmpeg_encoding_options.preset	String		<p>Для кодировщика libx264 список возможных значений: fast, superfast, ultrafast и т.п. Полный список значений можно посмотреть здесь: https://trac.ffmpeg.org/wiki/Encode/H.264.</p> <p>Примечание: для кодировщика h264_nvenc (при работе с GPU) список возможных значений отличается: slow, medium, fast, hp, hq и т.п. Полный список значений можно посмотреть через командную строку ffmpeg -h encoder=h264_nvenc.</p>	Нет	<p>Имя пресета при кодировании видео в формате h.264. Для кодировщика libx264 список возможных значений: fast, superfast, ultrafast и т.п. Полный список значений можно посмотреть здесь: https://trac.ffmpeg.org/wiki/Encode/H.264. Примечание: для кодировщика h264_nvenc (при работе с GPU) список возможных значений отличается: slow, medium, fast, hp, hq и т.п. Полный список значений можно посмотреть через командную строку ffmpeg -h encoder=h264_nvenc.</p>
profiles.video.ffmpeg_encoding_options.b	String	Может быть как число, выраженное в бит/с, так и сокращения, наприм		Нет	<p>Значение фиксированного битрейта видео.</p>

		ер, “2М”, “64К”. Примечание: в случае кодека libx264 число означает не бит/с, а Кбит/с.			
profiles.video.ffmpeg_encoding_options.threads	String		<ul style="list-style-type: none"> • auto • [положительное целое число] 	Нет	Число потоков, которое может использовать кодировщик, если он поддерживает многопоточность. Возможные значения: <ul style="list-style-type: none"> • auto или 0 для автоматического определения оптимального числа потоков, • конкретное число. Значение по умолчанию: auto.
profiles.video.ffmpeg_encoding_options.profile	String		main	Нет	Параметр ffmpeg для кодека. Установлено значение main для всех профилей транскодирования для возможности декодирования h264 всеми браузерами и мобильными устройствами.
profiles.video.ffmpeg_encoding_options.bufsize	Integer	Может быть как число, выраженное в бит/с, так и сокращения, например, “2М”, “64К”	0 - 8М	Нет	Размер буфера управления битрейтом.

profiles.video.ffmpeg_encoding_options.maxrate	Integer	Может быть как число, выраженное в бит/с, так и сокращения, например, “2М”, “64К”	0 - 16М	Нет	Максимальный битрейт видео.
profiles.video.ffmpeg_encoding_options.tune	String			Нет	Настройка параметров декодирования.
profiles.video.ffmpeg_encoding_options.refs	Integer		0-16	Нет	Количество ключевые кадров компенсации движения, которые может использовать каждый Р-кадр.
profiles.video.ffmpeg_encoding_options.no-scenecut	Integer		0-100	Нет	Значение, используемое кодировщиком для расчета, должен ли он форсировать ключевой кадр (вставка дополнительного I-кадра).
profiles.video.ffmpeg_encoding_options.coder	String			Нет	Метод кодирования.
profiles.video.ffmpeg_encoding_options.crf	Integer		0-63	Нет	Коэффициент фиксированной скорости передачи данных, компромисс между качеством и размером для режимов постоянного качества (без целевого битрейта) и ограниченного качества (с максимальным целевым битрейтом), более высокие числа указывают на более низкое качество и меньший размер результирующего файла.
profiles.audio	Struct	Группа параметров		Нет	Блок выходных параметров обрабатываемой аудиодорожки.
profiles.audio.stream_selection	String		1. all 2. best	Да	Способ выбора видео- или аудиодорожки или дорожек

			3. none		<p>для обработки и записи в итоговый файл:</p> <ul style="list-style-type: none"> • для видео - параметры выбора видеодорожки; • для аудио - параметры выбора аудиодорожки. <p>Доступны следующие значения:</p> <ul style="list-style-type: none"> • all - все дорожки этого типа; • best - одна «лучшая» дорожка в соответствии с критериями FFmpeg; • none - дорожки этого типа не обрабатываются (например, может быть использовано для профилей генерации аудио файла).
profiles.audio.codec	String	Группа параметров		Да	<p>Название кодека в формате обозначений FFmpeg. Тип кодека зависит от того, к какому блоку относится параметр:</p> <ul style="list-style-type: none"> • если параметр к обработке видео - видеокодек; • если параметр относится к обработке аудио - аудиокодек. <p>Получить список поддерживаемых текущей версией FFmpeg кодеков можно командой <code>ffmpeg – encoders</code>.</p>
profiles.audio.filter_options	String	Группа параметров		Да	<p>Блок параметров преобразования:</p> <ul style="list-style-type: none"> • для блока video - преобразование видеодорожки; • для блока audio - преобразование аудиодорожки.

profiles.audio.filter_options.channels	Integer		1.1 2.2	Да	Количество каналов аудиодорожки. Возможны следующие значения: <ul style="list-style-type: none"> • 1 - моно; • 2 - стерео.
profiles.audio.filter_options.samplerate	Integer	В Гц		Да	Частота аудио кадров.
profiles.audio.ffmpeg_encoding_options	Struct			Нет	Блок настроек FFmpeg, передаваемых напрямую в кодировщик в формате "строка": "строка". Полный список возможных опций можно посмотреть на сайте https://ffmpeg.org/ffmpeg-codecs.html (на английском языке). Например: <ul style="list-style-type: none"> • b - значение фиксированного битрейта аудио: может выраженным в бит/с, так и сокращением, например, 2М, 64К.
profiles.audio.ffmpeg_encoding_options.b	String	Может быть как число, выраженное в бит/с, так и сокращения, например, "2М", "64К"		Нет	Значение фиксированного битрейта аудио.
thumbnail	Struct	Группа параметров		Нет	Блок параметров генерации набора эскизов (thumbnail) видео.
thumbnail.number	Integer			Да	Количество эскизов (thumbnail), которое необходимо сгенерировать. Параметры number, offset и step в блоке thumbnail должны быть согласованы между

					собой так, чтобы временная позиция последней миниатюры не превысила 100% относительно полной длины видео.
thumbnail.offset	Float			Да	Смещение временной позиции первого эскиза (thumbnail) относительно начала видео, в процентах от полной длины видео. Параметры number, offset и step в блоке thumbnail должны быть согласованы между собой так, чтобы временная позиция последней миниатюры не превысила 100% относительно полной длины видео.
thumbnail.step	Float			Да	Смещение временных позиций эскизов (thumbnail) относительно друг друга в процентах от полной длины видео. Параметры number, offset и step в блоке thumbnail должны быть согласованы между собой так, чтобы временная позиция последней миниатюры не превысила 100% относительно полной длины видео.
thumbnail.width	Integer			Да	Ширина эскиза (thumbnail) в пикселях.
thumbnail.height	Integer			Да	Высота эскиза (thumbnail) в пикселях.
thumbnail.quality	Integer		от 1 до 31	Да	Уровень качества изображения. Допустимый диапазон от 1 до 31 включительно, где 1 — самое высокое качество/самый большой размер файла, а 31 — самое низкое качество/наименьший размер файла. Подробнее см. аналогичную опцию FFmpeg: https://trac.ffmpeg.org/

					wiki/Encode/MPEG-4 (на английском языке).
trickmode	Struct	Группа параметров		Нет	Блок информации о сгенерированных наборах миниатюр для ознакомительного просмотра.
trickmode.frame_width	Integer		от 4 до 16254	Да	Ширина эскиза для ознакомительного просмотра. Минимальное значение – 4 пикселя, максимальное значение – 16254, делённое на количество столбцов холста.
trickmode.frame_height	Integer		от 4 до 16254	Да	Высота миниатюры. Минимальное значение – 4 пикселя, максимальное значение – 16254, делённое на количество строк холста.
trickmode.columns	Integer			Да	Количество столбцов с эскизами на холсте для ознакомительного просмотра (т. е. количество миниатюр в одном ряду).
trickmode.rows	Integer			Да	Количество рядов с миниатюрами на холсте.
trickmode.frames	Integer			Да, если нет step	Общее количество сгенерированных эскизов для ознакомительного просмотра по всем холстам (результатирующим файлам). Параметры step и frames в блоке trickmode взаимоисключающие.
trickmode.step	Float		логично, если step равен длине сегмента видео (чанка), т.е. 2 секунды	Да, если нет frames	Шаг в секундах между двумя соседними эскизами для ознакомительного просмотра. Параметры step и frames в блоке trickmode взаимоисключающие.
trickmode.quality	Integer		от 1 до 31	Да	Уровень качества изображения. Допустимый диапазон от 1 до 31 включительно, где 1 — самое высокое качество/самый большой размер файла, а 31 — самое низкое

					качество/наименьший размер файла. Подробнее см. аналогичную опцию FFmpeg: https://trac.ffmpeg.org/wiki/Encode/MPEG-4 (на английском языке).
pix_bit_depth	Integer		8 или 10	Нет	Позволяет отметить профиль для транскодирования HDR видео (видео с битностью пикселя > 8).
probe	Struct	Группа параметров		Да, если нет step	Блок параметров модуля Prober.
probe.check_demux	Boolean	Флаг	1. true 2. false	Нет	Проверка доступности и корректной считываемости данных перед декодированием. Возможные значения: <ul style="list-style-type: none"> • true (истина); • false (ложь).
probe.check_decode	Boolean	Флаг	1. true 2. false	Нет	Проверка, что выбранные потоки с лучшим качеством (best) успешно декодируются. Возможные значения: <ul style="list-style-type: none"> • true (истина); • false (ложь).

6.2. Задание на загрузку медиафайла

Модуль Uploader отправляет уведомления в брокер сообщений Kafka о начале загрузки файла клиента.

Формат сообщения

Пример сообщения о создании задачи загрузки в хранилище

```
{
  "message_type": "UPLOADER_TASK_CREATE",
  "task_id": "e3338b4a-2771-11ec-9621-0242ac130002",
  "file": {
    "size": 123545,
    "original_filename": "video.mp4",
```

```

    "local_path": "/tmp/e3338b4a-2771-11ec-9621-0242ac130002/video.mp4"
  },
  "metadata": {
    "mimetype": "video/mp4"
  },
  "device": {
    "client_ip": "127.0.0.1",
    "user_agent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/98.0.4758.80 Safari/537.36"
  },
  "errors": []
}

```

Пример сообщения о неудаче в создании задачи загрузки в хранилище

```

{
  "message_type": "TASK_STATUS",
  "status": "UploaderFailed",
  "task_id": "01GKTQGJ6J70PF080JFTD0WJPC",
  "timestamp": 1670565611,
  "utc_time": "2022-12-09T06:00:11Z",
  "error": "Uploader. unexpected EOF",
  "errors": [
    {
      "error_code": "1010402",
      "error_message": "Uploader. unexpected EOF"
    }
  ]
}

```

Таблица с описанием параметров

Параметр	Тип	Формат	Диапазон возможных значений	Обязательность использования	Описание

message_type	String	client_ip	UPLOADER_TASK_CREATE	Да	Тип сообщения. Может принимать следующие значения: <ul style="list-style-type: none"> UPLOADER_TASK_CREATE - задание на загрузку файла создано.
task_id	String	UUID		Да	Глобальный идентификатор задачи в формате UUID, например: 30f709e8-5a5a-483e-9bcf-449067e261ef.
file	Struct	Группа параметров		Да	Блок описания загружаемого файла.
file.size	Integer		1 - 1 073 741 824 <i>Максимальный размер загружаемого файла - 1Гб</i>	Да	Размер файла в байтах.
file.original_filename	String	Название файла		Да	Исходное название видеофайла.
file.local_path	String	/file_path		Да	Абсолютный путь к файлу во временном локальном хранилище.
metadata	Struct	Группа параметров		Да	Блок метаданных медиафайла.
metadata.mimetype	String			Да	MIME-тип файла.
device	Struct	Группа параметров		Да	Блок сведений о клиентском приложении, через которое загружается видео.
device.client_ip	String	host		Да	IP-адрес клиента.
device.user_agent	String			Да	Строка описания клиентского приложения, например: "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.80 Safari/537.36".
error	String	На английском языке		Да, если есть ошибка	Подробное сообщение об ошибке (на английском языке). Поле присутствует, только если

					значение поля status (статус) имеет значение FAILED (сбой).
errors	Array	Группа параметров		Да, если есть ошибка	Массив, описывающий ошибки.
error_code	Integer	7 знаков	1010101 - 1999999	Да, если есть ошибка	Код ошибки.
error_message	String	Текст на английском языке		Да, если есть ошибка	Сообщение об ошибке.

6.3. Загрузка медиафайла

О начале и завершении загрузки медиафайла Uploader уведомляет Task Manager, отправляя сообщение в очередь KAFKA_TOPIC_TASK_MANAGER компонента Kafka. Сообщение содержит следующие параметры (см. таблицу ниже).

Формат сообщения

Пример сообщения о начале переноса видео в постоянное хранилище

```
{
  "message_type": "TRANSFER_TASK_START",
  "task_id": "e3338b4a-2771-11ec-9621-0242ac130002",
  "error": "",
  "status": "OK",
  "source": {
    "provider": "fs",
    "s3_region": "",
    "host_uri": "",
    "bucket": "",
    "file_uri": "/tmp/e3338b4a-2771-11ec-9621-0242ac130002/video.mp4"
  },
  "destination": {
    "provider": "s3",
    "s3_region": "ru",
    "host_uri": "test.ru:9000",
    "bucket": "hls",

```

```
    "file_uri": "user/upload/video.mp4"
  },
  "error": "Error database connection",
  "errors": [
    {
      "error_code": 1010402,
      "error_message": "Error database connection"
    }
  ]
}
```

Пример сообщения о завершении переноса видео во временное хранилище

```
{
  "message_type": "TRANSFER_TASK_FINISHED",
  "task_id": "e3338b4a-2771-11ec-9621-0242ac130002",
  "error": "",
  "status": "OK",
  "source": {
    "provider": "fs",
    "s3_region": "",
    "host_uri": "",
    "bucket": "",
    "file_uri": "/tmp/pT5iEn4CMaq7WUPoO7Doj/video.mp4"
  },
  "destination": {
    "provider": "s3",
    "s3_region": "ru",
    "host_uri": "test.ru:9000",
    "bucket": "hls",
    "file_uri": "user/upload/video.mp4"
  }
  "error": "Error database connection",
  "errors": [
```

```

{
  "error_code": 1010402,
  "error_message": "Error database connection"
}
]
}

```

Таблица с описанием параметров

Название	Тип	Формат	Диапазон возможных значений	Обязательно	Описание
message_type	String		<ol style="list-style-type: none"> TRANSFER_TASK_START TRANSFER_TASK_FINISHED 	Да	Тип сообщения. Может принимать следующие значения: <ul style="list-style-type: none"> TRANSFER_TASK_START - перенос файла начат; TRANSFER_TASK_FINISHED - загрузка файла завершена.
task_id	String	UUID		Да	Глобальный идентификатор задачи в формате UUID, например: 30f709e8-5a5a-483e-9bcf-449067e261ef.
error	String	На английском языке		Да, если есть ошибка	Подробное сообщение об ошибке (на английском языке). Поле присутствует, только если значение поля status (статус) имеет значение FAILED (сбой).
errors	Array	Группа параметров		Да, если есть ошибка	Информация о найденных ошибках в виде массива.
errors.error_code	String	7 знаков	1010101 - 1999999	Да, если есть ошибка	Код ошибки.
errors.error_message	String			Да, если есть ошибка	Сообщение об ошибке.

status	String		<ol style="list-style-type: none"> OK FAILED 	Да	<p>Результат выполнения задачи. Может принимать следующие значения:</p> <ul style="list-style-type: none"> OK - успешное завершение задачи; FAILED - перенос файла не удался.
source	Struct	Группа параметров		Да	Блок параметров доступа к видео.
source.provider	String		<ol style="list-style-type: none"> fs s3 	Да	<p>Тип хранилища:</p> <ul style="list-style-type: none"> s3 - объектное хранилище S3 (Object storage S3) — это облачный сервис, позволяющий хранить файлы любого типа и объема; fs - локальный диск (file system, FS).
source.file_uri	String	/file_path		Да	<p>Путь к файлу в хранилище.</p> <ul style="list-style-type: none"> для s3 - это абсолютный путь к файлу внутри контейнера (bucket); для fs - это либо путь к файлу относительно переменной окружения BASE_PATH, либо абсолютный путь.
source.s3_region	String			Нет	<p>Название региона хранилища S3 (если используется это хранилище). Значение, которое должно совпадать с аналогичным значением на стороне хранилища S3. Допустимо вводить любые значения, однако подразумевается, что здесь будет указан код страны (например, ru-central). Если в качестве хранилища используется FS, поле не используется.</p>

source.bucket	String		0..9, a..z, A..Z	Нет	Имя контейнера (bucket) в хранилище S3. Если в качестве хранилища используется FS, поле не используется.
source.host_uri	String	/file_path		Нет	Сетевой путь к службе S3 в формате {схема}://{хост IP}[:{порт}]. Если в качестве хранилища используется FS, поле не используется.
destination	Struct	Группа параметров		Да	Блок, описывающий параметры пути переноса файла.
destination.provider	String		<ol style="list-style-type: none"> 1. fs 2. s3 	Да	<p>Тип хранилища:</p> <ul style="list-style-type: none"> • s3 - объектное хранилище S3 (Object storage S3) — это облачный сервис, позволяющий хранить файлы любого типа и объема; • fs - локальный диск (file system, FS).
destination.file_uri	String	/file_path		Да	<p>Путь к файлу в хранилище.</p> <ul style="list-style-type: none"> • для s3 это абсолютный путь к файлу внутри контейнера (bucket); • для fs это либо путь к файлу относительно переменной окружения BASE_PATH, либо абсолютный путь.

destination.s3_region	String			Нет	Название региона хранилища S3 (если используется это хранилище). Значение, которое должно совпадать с аналогичным значением на стороне хранилища S3. Допустимо вводить любые значения, однако подразумевается, что здесь будет указан код страны (например, ru-central). Если в качестве хранилища используется FS, поле не используется.
destination.bucket	String		0..9, a..z, A..Z	Нет	Имя контейнера (bucket) в хранилище S3. Если в качестве хранилища используется FS, поле не используется.
destination.host_uri	String	/file_path		Нет	Сетевой путь к службе S3 в формате {схема}://{хост IP}[:{порт}]. Если в качестве хранилища используется FS, поле не используется.
errors	Array	Группа параметров		Нет	Массив ошибок.
error_code	Integer	7 знаков	1010101 - 1999999	Да, если есть ошибка	Код ошибки.
error_message	String	Текст на английском языке		Да, если есть ошибка	Сообщение об ошибке.

6.4. Задание на анализ исходного медиафайла

Получив от Task Manager сообщение с задачей на анализ исходного медиафайла, модуль Prober начинает анализ, обрабатывая поочередно сообщения из топика `KAFKA_TOPIC_PROBE`, и отправляет соответствующее сообщение в очередь `KAFKA_TOPIC_TASK_MANAGER` компонента Kafka. Параметры сообщения приведены в таблице ниже. Задачи в модуль отправляются поочередно, за это отвечает балансировщик в

составе модуля. Транспорт может быть дополнен работой через командную строку. Формат сообщения при этом не поменяется.

KAFKA_TOPIC_PROBE, KAFKA_TOPIC_TASK_MANAGER - переменные окружения модуля Prober.

Для сообщения Kafka должен быть добавлен header **type=PROBE_TASK**.

Формат сообщения

Пример задачи на анализ для компонента Prober

```
{
  "message_type": "PROBE_TASK",
  "task_id": "e3338b4a-2771-11ec-9621-0242ac130002",
  "source": {
    "provider": "s3",
    "bucket": "",
    "s3_region": "ru",
    "host_uri": "https://test.ru",
    "file_uri": "video.mp4"
  },
  "telemetry_trace_context": {
    "trace_id": "AABBCCDDEEFF00112233445566778899",
    "parent_span_id": "ABCDEF0123456789"
  }
  "check_demux": true,
  "check_decode": true,
  "check_decode_sec": 100
}
```

Таблица с описанием параметров

Параметр	Тип данных	Формат	Диапазон возможных значений	Обязательно	Описание
----------	------------	--------	-----------------------------	-------------	----------

message_type	String		PROBE_T ASK	Да	Тип сообщения. Может принимать следующие значения: <ul style="list-style-type: none"> • PROBE_TASK - задача для компонента Prober.
task_id	String	UUID		Да	Глобальный идентификатор задачи в формате UUID, например: 30f709e8-5a5a-483e-9bcf-449067e261ef.
source	Struct	Группа параметров		Да	Блок параметров доступа к видео.
source.provider	String		1. fs 2. s3	Да	Тип хранилища: <ul style="list-style-type: none"> • s3 - объектное хранилище S3 (Object storage S3) — это облачный сервис, позволяющий хранить файлы любого типа и объема; • fs - локальный диск (file system, FS).
source.bucket	String			Обязательно, если используется хранилище S3	Имя контейнера (bucket) в хранилище S3.
source.s3_region	String			Обязательно, если используется хранилище S3	Название региона хранилища S3 (если используется это хранилище). Значение, которое должно совпадать с аналогичным значением на стороне хранилища S3. Допустимо вводить любые значения, однако подразумевается, что здесь будет указан код страны (например, ru-central).
source.host_uri	String	host:port		Обязательно, если используется хранилище S3	Сетевой путь к службе S3 в формате {схема}://{хост IP}[:{порт}].

source.file_uri	String	/file_path		Да	<p>Путь к файлу в хранилище.</p> <ul style="list-style-type: none"> Для s3 это абсолютный путь к файлу внутри контейнера (bucket); Для fs это либо путь к файлу относительно переменной окружения BASE_PATH, либо абсолютный путь.
telemetry_trace_context.trace_id	String	Ненулевое HEX-число. Максимальная длина HEX-числа - 16 байт		Да, если передаётся блок telemetry_trace_context	Идентификатор трассировки - строка, содержащая ненулевое HEX-число. Максимальная длина HEX-числа - 16 байт.
check_demux	Boolean	Флаг	<ol style="list-style-type: none"> true false 	Нет	<p>Проверка доступности и корректной считываемости данных перед декодированием.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> true (истина); false (ложь).
check_decode	Boolean	Флаг	<ol style="list-style-type: none"> true false 	Нет	<p>Проверка, что выбранные потоки с лучшим качеством (best) успешно декодируются.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> true (истина); false (ложь).
check_decode_sec	Integer	В секундах		Нет	<p>Максимальная длительность декодируемого участка в секундах. Игнорируется, если check_decode = false.</p> <p>Значения, равные и меньше 0, отключают декодирование, оставляя только проверку наличия соответствующего декодера.</p>

6.5. Результат анализа исходного медиафайла

Завершив анализ исходного видео, компонент Prober отправляет в топик KAFKA_TOPIC_TASK_MANAGER компонента Kafka сообщение с результатом анализа. Ниже представлено ответное сообщение на Probe_Task, описывающее результат работы сервиса Probe.

Формат сообщения

```
{
  "message_type": "PROBE_TASK_FINISHED",
  "task_id": "01H8GTNTVPAS9Q0",
  "status": "OK",
  "errors": [],
  "location": {
    "provider": "s3",
    "bucket": "media-platform",
    "s3_region": "ru-central",
    "host_uri": "https://example.net",
    "file_uri": "01H8GTNTVPAS9Q0D/16_10_3840x2400.mp4"
  },
  "file_size": 569498952,
  "container_short_names": "mov,mp4,m4a,3gp,3g2,mj2",
  "streams": [
    {
      "stream_type": "VIDEO",
      "stream_idx": 0,
      "duration": 201.39999389648438,
      "codec": "h264",
      "bitrate": 22307545,
      "width": 3840,
      "height": 2400,
      "pix_fmt": "yuv420p",
      "sample_aspect_ratio": {
        "num": 655,
```

```
        "den": 666
    },
    "display_aspect_ratio": {
        "num": 524,
        "den": 333
    },
    "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": 201.32571411132812,
    "codec": "mp3",
    "bitrate": 320001,
    "samplerate": 44100,
    "channels": 2
}
],
"best_video_stream": 0,
"best_audio_stream": 1,
"demux_ok": true,
"start_timestamp_ms": 1692782120601,
"duration_us": 17616480
}
```


Это сообщение имеет следующие параметры (см. таблицу ниже).

Таблица с описанием параметров

Параметр	Тип данных	Формат	Диапазон возможных значений	Обязательно	Описание
message_type	String		PROBE_TASK	Да	Тип сообщения. Может принимать следующие значения: <ul style="list-style-type: none"> PROBE_TASK - задача для компонента Prober.
task_id	String	UUID		Да	Глобальный идентификатор задачи в формате UUID, например: 30f709e8-5a5a-483e-9bcf-449067e261ef.
status	String		<ol style="list-style-type: none"> OK FAILED 	Да	Результат выполнения задачи. Может принимать следующие значения: <ul style="list-style-type: none"> OK - успешное завершение задачи; FAILED - перенос файла не удался.
error	String	На английском языке		Да, если есть ошибка	Подробное сообщение об ошибке (на английском языке). Поле присутствует, только если значение поля status (статус) имеет значение FAILED (сбой).
errors	Array	Группа параметров		Да, если есть ошибка	Информация о найденных ошибках в виде массива.
errors.error_code	String	7 знаков	1010101 - 1999999	Да, если есть ошибка	Код ошибки.
errors.error_message	String			Да, если есть ошибка	Сообщение об ошибке.
location	Struct	Группа параметров		Да	Блок параметров доступа к видео.
location.provider	String		<ol style="list-style-type: none"> fs s3 	Да	Тип хранилища: <ul style="list-style-type: none"> s3 - объектное хранилище S3 (Object storage S3) — это облачный сервис, позволяющий хранить файлы

					любого типа и объема; <ul style="list-style-type: none"> fs - локальный диск (file system, FS).
location.host_uri	String	host:port		Обязательно, если используется хранилище S3	Сетевой путь к службе S3 в формате {схема}://{хост IP}[:{порт}].
location.bucket	String		0..9, a..z, A..Z	Нет	Имя контейнера (bucket) в хранилище S3. Если в качестве хранилища используется FS, поле не используется.
location.s3_region	String			Нет	Название региона хранилища S3 (если используется это хранилище). Значение, которое должно совпадать с аналогичным значением на стороне хранилища S3. Допустимо вводить любые значения, однако подразумевается, что здесь будет указан код страны (например, ru-central). Если в качестве хранилища используется FS, поле не используется.
file_size	Integer	Положительное число	1 - 1 073 741 824 <i>Максимальный размер загружаемого файла - 1Гб</i>	Да	Размер файла в байтах.
container_short_names	String			Да	Список допустимых коротких имен контейнера через запятую.
streams	Array			Да	Массив информации о видео-аудиопотоках.
streams.stream_type	String		1. VIDEO 2. AUDIO	Да	Тип потока: <ul style="list-style-type: none"> VIDEO - видеопоток; AUDIO - аудиопоток.

streams.stream_idx	Integer	Положительное число		Да	Индекс (порядковый номер) потока в контейнере.
streams.duration	Float	s.ms		Да	Время видео/аудиодорожки в формате <секунды>.<миллисекунды>. Пример: 19.734.
streams.codec	String	Группа параметров		Да	Название кодека в формате обозначений FFmpeg. Тип кодека зависит от того, к какому блоку относится параметр: <ul style="list-style-type: none"> • если параметр к обработке видео - видеокодек; • если параметр относится к обработке аудио - аудиокодек. Получить список поддерживаемых текущей версией FFmpeg кодеков можно командой <code>ffmpeg - encoders</code> .
streams.bitrate	Integer	Положительное число		Да	Битрейт потока.
streams.width	Integer	Положительное число	от 4 до 16254	Да, если "stream_type": "VIDEO"	Ширина миниатюры в пикселях
streams.height	Integer	Положительное число	от 4 до 16254	Да, если "stream_type": "VIDEO"	Высота миниатюры в пикселях
streams.sample_aspect_ratio	Struct			Да, если "stream_type": "VIDEO"	Соотношение сторон пикселя.
streams.sample_aspect_ratio.num	Integer	Положительное число		Да	Числитель в соотношении сторон пикселя.
streams.sample_aspect_ratio.den	Integer	Положительное число		Да	Знаменатель в соотношении сторон пикселя.
streams.display_aspect_ratio	Struct			Да, если "stream_type": "VIDEO"	Соотношении сторон видео кадра.

streams.display_aspect_ratio.num	Integer	Положительное число		Да	Числитель, в соотношении сторон видео — числитель/знаменатель.
streams.display_aspect_ratio.den	Integer	Положительное число		Да	Знаменатель в соотношении сторон видео — числитель/знаменатель.
streams.frame_rate	Struct			Да, если "stream_type": "VIDEO"	Блок, описывающий частоту кадров в единицу времени. Число кадров в секунду выражается соотношением. Например: <ul style="list-style-type: none"> • при частоте 30 кадров в секунду это соотношение 30/1; • при частоте 29,97 кадра в секунду — 30000/1001.
streams.frame_rate.num	Integer	Положительное число		Да	Числитель в соотношении, которое определяет число кадров в секунду.
streams.frame_rate.den	Integer	Положительное число		Да	Знаменатель в соотношении, которое определяет число кадров в секунду.
streams.decode_open_ok	Boolean			Нет	Декодер соответствующего потока найден.
streams.decode_try_ok	Boolean			Нет	Начало стрима успешно декодировано.
streams.decode_ok	Boolean			Нет	Стрим успешно декодирован.
streams.samplerate	Integer	В Гц		Да, если "stream_type": "AUDIO"	Частота аудио кадров.
streams.channels	Integer		1.1 2.2	Да, если "stream_type": "AUDIO"	Количество каналов аудиодорожки. Возможны следующие значения: <ul style="list-style-type: none"> • 1 - моно; • 2 - стерео
streams.pix_fmt	String			Нет	Формат пикселей видео, определяется для видео потоков. Возможные значения можно получить из ffmpeg с помощью команды

					"ffmpeg -pix_fmts", например "yuv420p".
streams.color_range	String			Нет	Параметры цветового пространства исходного видео, определяется для видео потоков.
streams.color primaries	String			Нет	
streams.color_trc	String			Нет	
streams.color_space	String			Нет	
streams.chroma_location	String			Нет	
best_video_stream	Integer	Положительное число		Нет	Индекс лучшего видеопотока в контейнере.
best_audio_stream	Integer	Положительное число		Нет	Индекс лучшего аудиопотока в контейнере.
demux_ok	Boolean			Нет	Все пакеты были успешно прочитаны перед отправкой на декодер.
start_timestamp_ms	Integer	Положительное число		Да	Отметка времени начала обработки, выраженная в UNIX-time, то есть в миллисекундах, прошедших с 1 января 1970 года. Замеряется при помощи системных часов. Подвержена коррекциям системного времени: <ul style="list-style-type: none"> • NTP, Network Time Protocol (протокол системного времени); • leap second (секунда координации) • и так далее.
duration_us	Integer	Положительное число		Да	Продолжительность обработки, выраженная в микросекундах. Замеряется при помощи монотонных часов и не подвержена коррекциям системного времени.

6.6. Задание на обработку исходного медиафайла

Получив результаты анализа исходного видео от компонента Prober, Task Manager отправляет в топик `transcoder.tasks` компонента Kafka сообщение с задачей на обработку видео для Transcoder. Сервис Transcoder подключается к брокеру Kafka, обрабатывая поочередно

сообщения из топика `KAFKA_TOPIC_TRANSCODER`. Транспорт может быть дополнен работой через командную строку. Формат сообщения при этом не меняется.

- Форматы упаковки/кодирования аудио (audio sample format) и видео (video pixel format) итогового видеофайла выбираются автоматически. Используются соответствующие значения из исходного видео, если они поддерживаются выбранным кодеком. Если нет, форматы перекодируются в соответствии со значениями по умолчанию для выбранного кодека.
- PAR (pixel aspect ratio), или в терминологии FFmpeg - SAR (sample aspect ratio), итогового видео всегда приводится к 1/1.
- Для сообщения Kafka должен быть добавлен header `"type=TRANSCODER_TASK"`.

Формат сообщения v1.0

```
{
  "message_type": "TRANSCODER_TASK",
  "task_id": "e3338b4a-2771-11ec-9621-0242ac130002",

  "source": {
    "provider": "s3",
    "bucket" : "some-bucket",
    "s3_region": "ru",
    "host_uri": "https://test.ru",
    "file_uri": "video.mp4"
  },

  "destination": {
    "provider": "s3",
    "s3_region": "ru",
    "host_uri": "https://test.ru",
    "bucket": "some-bucket",
    "dir_uri": "e3338b4a-2771-11ec-9621-0242ac130002",
    "filename_template":
    "${source_filename}_${profile_name}_${container_type}.${filename_extension}",
    "thumbnail_template": "THUMBNAIL_${source_filename}_${time}_${index}.jpg",
```

```
        "master_playlist_template": "${source_filename}_playlist.m3u8"
    },

    "hw_device": "cuda",

    "ffmpeg_decoding_options": {
        "threads": "1"
    },

    "master_playlist": true,
    "streams_meta": [
        {
            "index": 0,
            "duration": 605.0369873046875
        },
        {
            "index": 1,
            "duration": 1.055999994277954
        }
    ],
    "profiles": [
        {
            "name": "main",
            "container_type": "hls",
            "filename_extension": "m3u8",
            "video": {
                "stream_selection": "best",
                "codec": "libx264",
                "filter_options": {
                    "width": 720,
                    "height": 576,
                    "fps": {
```

```

        "num" : 25,
        "den" : 1
    },
    "pix_fmt": "yuv420p"

    },
    "ffmpeg_encoding_options" : {
        "preset": "ultrafast",
        "b" : "1.2M",

        "threads" : "16"
    }
},
"audio": {
    "stream_selection": "all",
    "codec": "aac",
    "filter_options": {
        "channels": 2,
        "samplerate": 48000
    },
    "ffmpeg_encoding_options": {
        "b": "128K"
    }
},
    "ffmpeg_muxing_options" : {
        "hls_playlist_type" : "vod",
        "hls_segment_type" : "fmp4",
        "hls_time" : "2",
        "hls_flags" : "independent_segments+single_file"
    },
    "cpu" : {
        "name": "main",
        "container_type": "hls",

```



```
"filename_extension": "m3u8",
"video": {
  "stream_selection": "best",
  "codec": "libx264",
  "filter_options": {
    "width": 720,
    "height": 576,
    "fps": {
      "num" : 25,
      "den" : 1
    },
    "pix_fmt": "yuv420p",
```

```
      "overlay": {
"hw_device": "",
      "source": {
        "provider": "s3",
        "bucket": "test",
        "s3_region": "ru-central1",
        "host_uri": "https://storage.test.net",
        "file_uri": "watermarks/watermark1.jpeg",
        "http_url": ""
      },
      "ffmpeg_decoding_options": {
        "threads": "1"
      },
"opacity": 0.5,
"covered_area_fraction": 0.001,
"topleft_x_offset_fraction": 0.0,
"topleft_y_offset_fraction": 0.5,
"include_paddings": false,
```

```
"hw_device": "cuda"
```

```

    },
    "ffmpeg_encoding_options" : {
        "preset": "ultrafast",
        "b" : "1.2M",
        "threads" : "16"
    }
},
"audio": {
    "stream_selection": "all",
    "codec": "aac",
    "filter_options": {
        "channels": 2,
        "samplerate": 48000
    },
    "ffmpeg_encoding_options": {
        "b": "128K"
    }
},
"ffmpeg_muxing_options" : {
    "hls_playlist_type" : "vod",
    "hls_segment_type" : "fmp4",
    "hls_time" : "2",
    "hls_flags" : "independent_segments+single_file"
}
},
"gpu" : {
    "name": "main",
    "container_type": "hls",
    "filename_extension": "m3u8",
    "video": {
        "stream_selection": "best",

```

```

"codec": "h264_nvenc",
"filter_options": {
  "width": 720,
  "height": 576,
  "fps": {
    "num" : 25,
    "den" : 1
  }
},
"ffmpeg_encoding_options" : {
  "preset": "ultrafast",
  "b" : "1.2M",
  "threads" : "16"
}
},
"audio": {
  "stream_selection": "all",
  "codec": "aac",
  "filter_options": {
    "channels": 2,
    "samplerate": 48000
  },
  "ffmpeg_encoding_options": {
    "b": "128K"
  }
},
"ffmpeg_muxing_options" : {
  "hls_playlist_type" : "vod",
  "hls_segment_type" : "fmp4",
  "hls_time" : "2",
  "hls_flags" : "independent_segments+single_file"
}

```

```

    }
    },
  {
    "name": "audio-only",
    "container_type": "hls",
    "filename_extension": "m3u8",
    "video": {
      "stream_selection": "none"
    },
    "audio": {
      "stream_selection": "best",
      "codec": "aac",
      "filter_options": {
        "channels": 2,
        "samplerate": 22050
      },
      "ffmpeg_encoding_options": {
        "b": "64K"
      }
    },
    "ffmpeg_muxing_options": {
      "hls_playlist_type": "vod",
      "hls_segment_type": "fmp4",
      "hls_time": "2",
      "hls_flags": "independent_segments+single_file"
    }
  }
],

"thumbnail":
  {
    "times": [ 10.123, 20.234, 30.345 ],

```

```

"width": 1280,
"height": 720,
    "quality" : 2,
},

"telemetry_trace_context":
{
    "trace_id": "AABBCCDDEEFF00112233445566778899",
    "parent_span_id": "ABCDEF0123456789"
},

"trickmode": {
    "frame_width": 128,
    "frame_height": 72,
    "columns": 10,
    "rows": 10,
    "frames": 100,
    "quality": 8,
}
}

```

Это сообщение имеет следующие параметры (см. таблицу ниже).

Таблица с описанием параметров

	Параметр	Тип данных	Формат	Диапазон возможных значений	Обязательно	Описание				
1	message_type	String		TRANSCODE R_TASK	Да	Тип сообщения. Может принимать следующие значения (см. таблицу ниже). <table border="1" data-bbox="1247 1812 1502 1900"> <thead> <tr> <th>Значение</th> <th>Описание</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>	Значение	Описание		
Значение	Описание									

						TRANSCODER_TASK	Задача на обработку видео
2	task_id	String	UUID			Да	Глобальный идентификатор задачи в формате UUID, например: 30f709e8-5a5a-483e-9bcf-449067e261ef.
3	source	Struct	Группа параметров			Да	Блок параметров доступа к видео.
4	source.provider	String		1. fs 2. s3		Да	Тип хранилища: <ul style="list-style-type: none"> • s3 - объектное хранилище S3 (Object storage S3) — это облачный сервис, позволяющий хранить файлы любого типа и объема; • fs - локальный диск (file system, FS).
5	source.bucket	String				Обязательно, если используется хранилище S3	Имя контейнера (bucket) в хранилище S3.
6	source.s3_region	String				Обязательно, если используется хранилище S3	Название региона хранилища S3 (если используется это хранилище). Значение, которое должно совпадать с аналогичным значением на стороне хранилища S3.

						Допустимо вводить любые значения, однако подразумевается, что здесь будет указан код страны (например, ru-central).
7	source.host_uri	String	host:port		Обязательно, если используется хранилище S3	Сетевой путь к службе S3 в формате {схема}://{хост[IP][:{порт}]}
8	source.file_uri	String	/file_path		Обязательно, если используется хранилище S3	Путь к файлу в хранилище. <ul style="list-style-type: none"> • для s3 это абсолютный путь к файлу внутри контейнера (bucket); • для fs это либо путь к файлу относительно переменной окружения BASE_PATH, либо абсолютный путь.
9	destination	Struct	Группа параметров		Да	Блок, описывающий параметры пути переноса файла.
10	destination.provider	String		1. fs 2. s3	Да	Тип хранилища: <ul style="list-style-type: none"> • s3 - объектное хранилище S3 (Object storage S3) — это облачный сервис, позволяющий хранить файлы любого типа и объема; • fs - локальный

						диск (file system, FS).
1 1	destination.bucket	String			Обязательно, если используется хранилище S3	Имя контейнера (bucket) в хранилище S3.
1 2	destination.s3_region	String			Обязательно, если используется хранилище S3	Название региона хранилища S3 (если используется это хранилище). Значение, которое должно совпадать с аналогичным значением на стороне хранилища S3. Допустимо вводить любые значения, однако подразумевается, что здесь будет указан код страны (например, ru-central).
1 3	destination.host_uri	String	host:port		Обязательно, если используется хранилище S3	Сетевой путь к службе S3 в формате {схема}://{хост[IP][:{порт}]}
1 4	destination.dir_uri	String	/file_path		Обязательно, если используется хранилище S3	Путь к каталогу итоговых файлов. Для S3 это абсолютный путь к директории внутри bucket. Для FS это либо путь к директории относительно переменной окружения BASE_PATH, либо абсолютный путь.
1 5	destination.filename_template	String	\${source_filename}_\${profile_name}_\${container_type}.\${filename_extension}		Да	Шаблон имени итогового файла. Допустимые переменные: <ul style="list-style-type: none"> • \${source_filename} - имя исходного

					<p>файла без расширения;</p> <ul style="list-style-type: none"> • <code>{profile_name}</code> - значение поля <code>name</code> профиля, который использовался для обработки; • <code>{container_type}</code> - значение поля <code>container_type</code> профиля, который использовался для обработки; • <code>{filename_extension}</code> - значение поля <code>filename_extension</code> профиля, который использовался для обработки. <p>Пример использования: Шаблон вида <code>video.{source_filename}.{filename_extension}</code> приведёт к созданию файла вида <code>video.filename.mp4</code>.</p>
16	destination.thumbnail_template	String	<code>THUMBNAIL_{source_filename}_{time}_{index}.jpg</code>	Да, если профиль обработки содержит параметры создания эскиза (<code>thumbnail</code>)	<p>Шаблон имени для генерации эскиза (<code>thumbnail</code>) исходного файла. Допустимые переменные:</p> <ul style="list-style-type: none"> • <code>{source_filename}</code> - имя исходного файла без расширения;

						<ul style="list-style-type: none"> • <code>{time}</code> - временная позиция видео кадра, который был использован для создания эскиза (<code>thumbnail</code>); • <code>{index}</code> - индекс эскиза (<code>thumbnail</code>) в соответствии с запрошенным списком в <code>thumbnail.times</code>.
17	destination.master_playlist_template	String	<code>{source_filename}_master.m3u8</code>		Нет	Шаблон имени для генерации мастер плейлиста HLS. Допустима переменная <code>{source_filename}</code> — имя исходного файла без расширения.
18	hw_device	String		cuda	Нет	Тип используемого аппаратного ускорения для обработки видео. Допустимое значение: <code>cuda</code> . Поддерживаются карты NVIDIA, драйверы которых совместимы с CUDA 11.6. Кроме того, аппаратное ускорение поддерживается только в режиме, когда декодирование и фильтрация (<code>scaling</code>) и кодирование видео выполняются на GPU. Т. е. частичная обработка на GPU в

					<p>настоящее время недоступна. Это ограничивает возможный набор кодеков:</p> <ul style="list-style-type: none"> • декодирование: <ul style="list-style-type: none"> ○ mpeg2; ○ av1; ○ h264; ○ h265/hevc; ○ mjpeg; ○ mpeg4; ○ vc1; ○ vp8; ○ vp9. • кодирование: <ul style="list-style-type: none"> ○ h264_nvc; ○ hevc_nvc. <p>Аппаратное ускорение не применяется для обработки аудио и создания эскизов видео (thumbnail).</p>
19	ffmpeg_decoding_options	Structure	Группа параметров	Нет	<p>Блок настроек FFmpeg, передаваемых напрямую в декодер в формате "строка": "строка". Полный список возможных опций можно посмотреть здесь (внешний источник на английском языке). Например: threads - число потоков, которое может использовать</p>

						декодер, если он поддерживает многопоточность. Возможные значения: auto или 0 для автоматического определения оптимального числа потоков либо явно указанное число (значение по умолчанию: auto).
20	ffmpeg_decoding_options.threads	String		<ul style="list-style-type: none"> • auto • [полужителное целое число] 	Нет	Число потоков, которое может использовать декодер, если он поддерживает многопоточность. Возможные значения: auto или 0 для автоматического определения оптимального числа потоков либо явно указанное число (значение по умолчанию: auto)
21	master_playlist	Boolean	Флаг	<ol style="list-style-type: none"> 1. true 2. false 	Нет	Включает/отключает генерацию общего мастер-плейлиста HLS для всех профилей, у которых параметр container_type имеет значение hls. Доступные значения: <ul style="list-style-type: none"> • true - настройка включена; • false - настройка выключена.
22	profiles	Struct	Группа параметров		Настройки обработки задаются в топике Kafka KAFKA_TOPIC_TASKM	Блок, содержащий профиль или профили обработки медиаданных. Профили могут быть следующих видов:

				ANAGER_SETTINGS	<ul style="list-style-type: none"> • профиль обработки аудио и видео — название профиля может быть произвольным, профиль должен содержать вложенные блоки video и /или audio — для обработки аудио и видео соответственно; Блоки video и audio можно комбинировать в одном профиле (видео с аудио) или в разные (видео без аудио или аудио без видео), тогда как блоки thumbnail и trickmode нельзя комбинировать с другими блоками в рамках одного профиля . • профиль создания эскиза (thumbnail) - всегда передаётся в
--	--	--	--	-----------------	---

					<p>блоке thumbnail;</p> <ul style="list-style-type: none"> • профиль создания холста миниатюр для ознакомительного просмотра - всегда передаётся в блоке thumbnail.
2 3	profiles.name	String		<p>Настройки обработки задаются в топике Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS</p>	<p>Название профиля обработки медиаданных. Произвольное значение</p>
2 4	profiles.container_type	String		<p>Настройки обработки задаются в топике Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS</p>	<p>Итоговый формат файла в рамках обозначений FFmpeg:</p> <ul style="list-style-type: none"> • профиль обработки видео - видеоформат, например: mpeg4; • профиль обработки (извлечения) аудио - аудиоформат : например, mp3; • профиль создания эскиза (thumbnail) - формат изображения: например, jpeg.

						Получить список поддерживаемых текущей версией FFmpeg контейнеров можно командой <code>ffmpeg -formats</code> . Параметр должен быть указан для каждого профиля.	
2 5	profiles.filename_extension	String	Расширение файла			Настройки обработки задаются в топике Kafka <code>KAFKA_TOPIC_TASKMANAGER_SETTINGS</code>	Имя расширения итогового файла. Используется при подстановке в шаблоне <code>filename_template</code> .
2 6	profiles.video	Struct	Группа параметров			Настройки обработки задаются в топике Kafka <code>KAFKA_TOPIC_TASKMANAGER_SETTINGS</code>	Блок выходных параметров обрабатываемого видео.
2 7	profiles.video.stream_selection	String			<ol style="list-style-type: none"> 1. all 2. best 3. none 	Настройки обработки задаются в топике Kafka <code>KAFKA_TOPIC_TASKMANAGER_SETTINGS</code>	Способ выбора видео- или аудиодорожки или дорожек для обработки и записи в итоговый файл: <ul style="list-style-type: none"> • для видео - параметры выбора видеодорожки; • для аудио - параметры выбора аудиодорожки. Доступны следующие значения: <ul style="list-style-type: none"> • all - все дорожки этого типа;

					<ul style="list-style-type: none"> • best - одна «лучшая» дорожка в соответствии с критериями FFmpeg; • none - дорожки этого типа не обрабатываются (например: может быть использовано для профилей генерации аудио файла).
28	profiles.video.codec	String		<p>Настройки обработки задаются в топике Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS</p>	<p>Название кодека в формате обозначений FFmpeg. Тип кодека зависит от того, к какому блоку относится параметр:</p> <ul style="list-style-type: none"> • если параметр к обработке видео - видеокодек; • если параметр относится к обработке аудио - аудиокодек. <p>Получить список поддерживаемых текущей версией FFmpeg кодеков можно командой <code>ffmpeg – encoders</code>.</p>
29	profiles.video.filter_options	String	Группа параметров	<p>Настройки обработки задаются в топике Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS</p>	<p>Блок параметров преобразования:</p> <ul style="list-style-type: none"> • для блока video - преобразование

					ANAGER_SETTINGS	<p>видеодорожки;</p> <ul style="list-style-type: none"> • для блока audio - преобразование аудиодорожки.
30	profiles.video.filter_options.width	Integer			Настройки обработки задаются в топике Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS	Ширина видео в пикселях.
31	profiles.video.filter_options.height	Integer			Настройки обработки задаются в топике Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS	Высота видео в пикселях.
32	profiles.video.filter_options.fps	Struct	Группа параметров		Настройки обработки задаются в топике Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS	Блок настроек FPS (Frames per second - кадры в секунду), определяется соотношением значений параметров num и den, входящих в блок. Итоговым значением блока после обработки будет значение соотношения num/den.
33	profiles.video.filter_options.fps.num	Integer	Целое положительное число, num/den		Настройки обработки задаются в топике Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS	Число кадров в единицу времени (количество секунд указано в параметре den).

3 4	profiles.video.filter_options.fps.den	Integer	Целое положительное число		Настройки обработки задаются в топике Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS	Количество секунд, в течение которых отображается количество кадров, определённых в параметре num.
3 5	profiles.video.filter_options.pix_fmt	String		Возможные значения можно получить из ffmpeg с помощью команды "ffmpeg -pix_fmts", например yuv420p.	Нет	Формат пикселей видео. Итоговое видео будет преобразовано в указанный пиксель формат. Значение по умолчанию - пустая строка. Пустая строка также означает автоматический подбор формата пикселей на основе формата пикселей исходного видео и возможностях кодека.
3 6	profiles.video.filter_options.padding	String	Флаг	true, false	Нет	Если true, части видео, которые образовались из-за разного соотношения сторон кадра на входе и выходе, заполняются черным цветом. Значение по умолчанию - false.
3 7	profiles.video.filter_options.rotate	String	Флаг	true, false	Нет	Если true, то при наличии в метаданных исходного видео значения rotate, отличного от нуля, делается поворот кадра. Значение по умолчанию - false.
3 8	profiles.video.filter_options.allow_swap_dimensions	String	Флаг	true, false	Нет	Если true, то разрешается сделать замену width <-> height, если это оптимальнее для

						входной картинке. Значение по умолчанию - true.
39	profiles.video.filter_options.overlay	String	Флаг	true, false	Нет	Если true, то видео, указанное в секции overlay (далее – водяной знак), будет наложено поверх основного видео профиля от левого угла. Предварительно, видео будет масштабировано по размеру, указанному для профиля, с сохранением оригинальных пропорций. Значение по умолчанию - false.
40	profiles.video.filter_options.overlay_opacity	Integer	Дробное положительное число	(0.0, 1.0]	Нет	Степень прозрачности водяного знака. Здесь 1.0 – без прозрачности; близкое, но не равное нулю – максимальная прозрачность. Значение по умолчанию – 1.0.
41	profiles.video.filter_options.overlay_covered_area_fraction	Integer	Дробное положительное число	(0.0, 1.0]	Нет	Доля от общей площади в результирующем потоке видео потока, которую должен занимать водяной знак. Фактически наложенный водяной знак может покрывать площадь несколько меньше, если это требуется для сохранения аспекта исходного контента водяного знака. Значение по умолчанию – 1.0.

4 2	profiles.video.filter_options.overlay.to_left_x_offset_fraction	Integer	Дробное положительное число	[0.0, 1.0)	Нет	Доля смещения верхнего левого угла водяного знака по горизонтали вправо в результирующем видео потоке. Этот относительный параметр будет пересчитан в пиксели (Integer) и на этот расчет может влиять наличие paddings в результирующем видео потоке и значение параметра overlay.include_paddings. Значение по умолчанию – 0.0.
4 3	profiles.video.filter_options.overlay.to_left_y_offset_fraction	Integer	Дробное положительное число	[0.0, 1.0)	Нет	Доля смещения верхнего левого угла водяного знака по вертикали вниз в результирующем видео потоке. Этот относительный параметр будет пересчитан в пиксели (Integer) и на этот расчет может влиять наличие paddings в результирующем видео потоке и значение параметра overlay.include_paddings. Значение по умолчанию – 0.0.
4 4	profiles.video.filter_options.overlay.include_paddings	String	Флаг	true, false	Нет	Флаг, определяющий будут ли учитываться paddings при определении положения водяного знака. Значение по умолчанию – false.
4 5	profiles.video.ffmpeg_encoding_options	String	Целое положительное число.		Настройки обработки задаются в топике Kafka KAFKA_	Блок настроек FFmpeg, передаваемых напрямую в кодировщик в

				<p>TOPIC_TASKMANAGER_SETTINGS</p>	<p>формате "строка":"строка". Полный список возможных опций можно посмотреть по ссылке https://ffmpeg.org/ffmpeg-codecs.html (на английском языке). Ниже представлены примеры опций.</p> <ul style="list-style-type: none"> • preset - имя пресета при кодировании видео в формате h.264. Для кодировщика libx264 список возможных значений: fast, superfast, ultrafast, и т. п. Полный список значений можно посмотреть по адресу https://trac.ffmpeg.org/wiki/Encode/H.264 (на английском языке). Для кодировщика h264_nvenc (при работе с GPU) список возможных значений отличается: slow, medium, fast, hp, hq и т. п. Полный список значений можно посмотреть
--	--	--	--	-----------------------------------	---

					<p>через командную строку: ffmpeg -g -h encoder=h264 -nvcv.</p> <ul style="list-style-type: none"> • b - значение фиксированного битрейта видео. Может быть выражено в бит/с, так и сокращением, например: 2 М, 64К. В случае кодека libx264 число означает не бит/с, а Кбит/с. • threads - число потоков, которое может использовать кодировщик, если он поддерживает многопоточность. Возможные значения auto или 0 для автоматического определения оптимального числа потоков либо конкретное число. Значение по умолчанию: auto.
--	--	--	--	--	--

4 6	profiles.video.ffmpeg_encoding_options.preset	String		<p>Для кодировщика libx264 список возможных значений: “fast“, “superfast“, "ultrafast", и т.п. Полный список значений можно посмотреть здесь: https://trac.ffmpeg.org/wiki/Encode/H.264.</p> <p>Примечание: для кодировщика h264_nvenc (при работе с GPU) список возможных значений отличается: "slow", "medium", "fast", "hp", "hq" и т.п. Полный список значений можно посмотреть через командную строку ffmpeg -h encoder=h264_nvenc.</p>	Нет	<p>Имя пресета при кодировании видео в формате h.264. Для кодировщика libx264 список возможных значений: “fast“, “superfast“, "ultrafast", и т.п. Полный список значений можно посмотреть здесь: https://trac.ffmpeg.org/wiki/Encode/H.264. Примечание: для кодировщика h264_nvenc (при работе с GPU) список возможных значений отличается: "slow", "medium", "fast", "hp", "hq" и т.п. Полный список значений можно посмотреть через командную строку ffmpeg -h encoder=h264_nvenc.</p>
4 7	profiles.video.ffmpeg_encoding_options.b	String	<p>Может быть как число, выраженное в бит/с, так и сокращения, например, “2М”, “64К“. Примечание: в случае кодека libx264 число означает не бит/с, а Кбит/с.</p>		Нет	<p>Значение фиксированного битрейта видео.</p>

48	profiles.video.ffmpeg_encoding_options.threads	String		<ul style="list-style-type: none"> • auto • [полужителное целое число] 	Нет	Число потоков, которое может использовать кодировщик, если он поддерживает многопоточность. Возможные значения auto или 0 для автоматического определения оптимального числа потоков, либо конкретное число. Значение по умолчанию - auto.
49	profiles.audio	Struct	Группа параметров		Настройки обработки задаются в топике Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS	Блок выходных параметров обрабатываемой аудиодорожки.
50	profiles.audio.stream_selection	String		<ol style="list-style-type: none"> 1. all 2. best 3. none 	Настройки обработки задаются в топике Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS	<p>Способ выбора видео- или аудиодорожки или дорожек для обработки и записи в итоговый файл:</p> <ul style="list-style-type: none"> • для видео - параметры выбора видеодорожки; • для аудио - параметры выбора аудиодорожки. <p>Доступны следующие значения:</p> <ul style="list-style-type: none"> • all - все дорожки этого типа; • best - одна «лучшая» дорожка в соответствии

					<p>с критериями FFmpeg;</p> <ul style="list-style-type: none"> • none - дорожки этого типа не обрабатываются (например: может быть использовано для профилей генерации аудио файла).
51	profiles.audio.codec	String	Группа параметров	<p>Настройки обработки задаются в топике Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS</p>	<p>Название кодека в формате обозначений FFmpeg. Тип кодека зависит от того, к какому блоку относится параметр:</p> <ul style="list-style-type: none"> • если параметр к обработке видео - видеокодек; • если параметр относится к обработке аудио - аудиокодек. <p>Получить список поддерживаемых текущей версией FFmpeg кодеков можно командой <code>ffmpeg - encoders</code>.</p>
52	profiles.audio.filter_options	String	Группа параметров	<p>Настройки обработки задаются в топике Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS</p>	<p>Блок параметров преобразования:</p> <ul style="list-style-type: none"> • для блока video - преобразование видеодорожки; • для блока audio -

						преобразован ие аудиодорожк и.
5 3	profiles.audio.filter_options.channels	Integer		1. 1 2. 2	Настройки обработки задаются в топике Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS	Количество каналов аудиодорожки. Возможны следующие значения: <ul style="list-style-type: none"> • 1 - моно; • 2 - стерео.
5 4	profiles.audio.filter_options.samplerate	Integer	В Гц		Нет	Частота аудио кадров.
5 5	profiles.audio.ffmpeg_encoding_options	String	Группа параметров		Настройки обработки задаются в топике Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS	Блок настроек FFmpeg, передаваемых напрямую в кодировщик в формате "строка":"строка". Полный список возможных опций можно посмотреть на сайте https://ffmpeg.org/ffmpeg-codecs.html (на английском языке). Например: <ul style="list-style-type: none"> • b - значение фиксированного битрейта аудио: может быть выраженным в бит/с, так и сокращением, например — 2M, 64K.
5 6	profiles.audio.ffmpeg_encoding_options.b	String	Может быть как число, выраженное в бит/с, так и сокращения, например, "2M", "64K".		Нет	Значение фиксированного битрейта аудио.
5 7	profiles.ffmpeg_muxing_options	String	Группа параметров		Настройки обработки задаются в топике	Опции FFmpeg, передаваемые напрямую в муксер в формате "строка":"строка"

				<p>Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS</p>	<p>рока". Полный список возможных опций можно посмотреть на сайте https://ffmpeg.org/ffmpeg-formats.html#Muxers (на английском языке). Например, для режима HLS (container_type равен hls). Ниже представлены возможные значения.</p> <ul style="list-style-type: none"> • hls_playlist_type - позволяет указать тип плейлиста. Возможные значения: vod и event. • hls_segment_type - тип контейнера для сегментов плейлиста. Возможные значения: mp4 и fmp4. • hls_time - желаемая длительность сегментов плейлиста. Формат возможных значений: https://ffmpeg.org/ffmpeg-utils.html#time-duration-syntax (на английском языке). • hls_flags - различные флаги, уточняющие
--	--	--	--	---	--

					<p>формат генерируемого плейлиста. Символ разделителя между флагами - «+».</p> <p>Примеры флагов:</p> <ul style="list-style-type: none">○ indep ende nt_se gmen ts - все сегм енты буду т гара нтир ован но начи нать ся с 1-го кадр а;○ singl e_file - все сегм енты плей лист а буду т сохр анен ы в одно м файл е, а сам плей лист
--	--	--	--	--	--

						будет оперировать диапазоном байтов внутри этого файла. Если destination ссылка на удалённое хранилище (например, S3), для режима HLS не могут применяться флаги и опции, которые подразумевают создание временных файлов и дальнейшее их переименование/перемещение. Например, нельзя использовать флаги temp_file и hls_playlist_type, не равные vod.
58	profiles.ffmpeg_muxing_options.hls_playlist_type	String		<ol style="list-style-type: none"> 1. vod 2. event 	Нет	Позволяет указать тип плейлиста.
59	profiles.ffmpeg_muxing_options.hls_segment_type	String		<ol style="list-style-type: none"> 1. mpegts 2. fmp4 	Нет	Тип контейнера для сегментов плейлиста.
60	profiles.ffmpeg_muxing_options.hls_time	Integer	В секундах https://ffmpeg.org/ffmpeg-utils.html#time-duration-syntax		Нет	Желаемая длительность сегментов плейлиста.
61	profiles.ffmpeg_muxing_options.hls_flags	String	Список флагов, разделенных символом "+".		Нет	Различные флаги, уточняющие формат генерируемого плейлиста.

6 2	thumbnail	St ru ct	Группа параметров		Настройки обработки задаются в топике Kafka KAFKA_ TOPIC_TASKM ANAGER_SETT INGS	Блок параметров генерации набора эскизов (thumbnail) видео.
6 3	thumbnail.times	Ar ra y			Настройки обработки задаются в топике Kafka KAFKA_ TOPIC_TASKM ANAGER_SETT INGS	Список временных позиций видеок кадров, которые будут использованы для создания эскизов (thumbnail).
6 4	thumbnail.width	Int eg er			Настройки обработки задаются в топике Kafka KAFKA_ TOPIC_TASKM ANAGER_SETT INGS	Ширина эскиза (thumbnail) в пикселях.
6 5	thumbnail.height	Int eg er			Настройки обработки задаются в топике Kafka KAFKA_ TOPIC_TASKM ANAGER_SETT INGS	Высота эскиза (thumbnail) в пикселях.
6 6	thumbnail.quality	Int eg er		От 1 до 31	Настройки обработки задаются в топике Kafka KAFKA_ TOPIC_TASKM ANAGER_SETT INGS	Уровень качества изображения. Допустимый диапазон от 1 до 31 включительно, где 1 — самое высокое качество/самый большой размер файла, а 31 — самое низкое качество/наименьший размер файла. Подробнее см. аналогичную опцию FFmpeg: https://trac.ff

					mpeg.org/wiki/Encode/MPEG-4 (на английском языке).
67	telemetry_trace_context	Structure	Группа параметров	Нет	Блок параметров, позволяющий связать в отображении диаграммы Ганта интервалы, которые генерируются разными компонентами Медиаплатформы при отображении в Zipkin.
68	telemetry_trace_context.trace_id	String	Максимальная длина HEX-числа - 16 байт.	Да, если передаётся блок <code>telemetry_trace_context</code>	Идентификатор трассировки - строка, содержащая ненулевое HEX-число. Максимальная длина HEX-числа - 16 байт.
69	telemetry_trace_context.parent_span_id	String	Ненулевое HEX-число. Максимальная длина HEX-числа - 8 байт.	Да, если передаётся блок <code>telemetry_trace_context</code>	Идентификатор родительского интервала, под который будут собраны интервалы от разных компонентов Медиаплатформы при отображении в Zipkin. Ненулевое HEX-число. Максимальная длина HEX-числа - 8 байт.
70	trickmode	Structure	Группа параметров	Настройки обработки задаются в топике Kafka <code>KAFKA_TOPIC_TASKMANAGER_SETTINGS</code>	Блок информации о сгенерированных наборах миниатюр для ознакомительного просмотра.
71	trickmode.frame_width	Integer		От 4 до 16254 Настройки обработки задаются в топике	Ширина эскиза для ознакомительного просмотра. Минимальное

					Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS	значение 4 пикселя, максимальное значение: 16254, делённое на количество столбцов холста.
7 2	trickmode.frame_height	Integer		От 4 до 16254	Настройки обработки задаются в топике Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS	Высота миниатюры. Минимальное значение - 4 пикселя, максимальное значение 16254, делённое на количество строк холста.
7 3	trickmode.columns	Integer			Настройки обработки задаются в топике Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS	Количество столбцов с эскизами на холсте для ознакомительного просмотра (т. е. количество миниатюр в одном ряду).
7 4	trickmode.rows	Integer			Настройки обработки задаются в топике Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS	Количество рядов с миниатюрами на холсте.
7 5	trickmode.frames	Integer			Настройки обработки задаются в топике Kafka KAFKA_TOPIC_TASKMANAGER_SETTINGS	Общее количество сгенерированных эскизов для ознакомительного просмотра по всем холстам (результатирующим файлам). Параметры <code>step</code> и <code>frames</code> в блоке <code>trickmode</code> взаимноисключающие.
7 6	trickmode.step	Float		Логично, если <code>step</code> равен длине сегмента	Настройки обработки задаются в топике	Шаг в секундах между двумя соседними эскизами для

				видео (чанка), т.е. 2 секунды.	Kafka KAFKA_ TOPIC_TASKM ANAGER_SETT INGS	ознакомительного просмотра. Параметры <code>step</code> и <code>frames</code> в блоке <code>trickmode</code> взаим оисключающие.
7 7	<code>trickmode.quality</code>	Integer		От 1 до 31	Настройки обработки задаются в топике Kafka KAFKA_ TOPIC_TASKM ANAGER_SETT INGS	Уровень качества изображения. Допустимый диапазон от 1 до 31 включительно, где 1 — самое высокое качество/самый большой размер файла, а 31 — самое низкое качество/наименьший размер файла. Подробнее см. аналогичную опцию FFmpeg: https://trac.ffmpeg.org/wiki/Encode/MPEG-4 (на английском языке).

6.7. Результат создания эскиза (thumbnail) медиафайла

По завершении создания эскиза компонент Transcoder отправляет в топик `KAFKA_TOPIC_TASK_MANAGER` компонента Kafka сообщение, предназначенное для компонента Task Manager.

Ниже представлено ответное сообщение на `Transcoder_Task`, содержащее информацию о сгенерированной миниатюре исходного файла. Сообщение отправляется сразу же, как только миниатюра готова, т.е. не дожидаясь полного транскодирования файла.

Формат сообщения

```
{
  "message_type": "THUMBNAIL_INFO",
  "task_id": "01GYS479JSSJJKRVC7N7PWZSKT",
  "status": "OK",
  "time": 3.1679999828338623,
  "error": "Transcoder. Thumbnail error",
  "errors": [
    {
```

```

    "error_code": 1040207,
    "error_message": "Transcoder. Thumbnail error"
  }
],
"location": {
  "provider": "s3",
  "bucket": "rupor-test2-storage",
  "s3_region": "ru-central1",
  "host_uri": "https://storage.yandexcloud.net",
  "file_uri": "01GYS479JSSJKRVC7N7PWZSKT/thumbnail.jpg"
},
"frame_size": {
  "width": 1280,
  "height": 720,
  "display_aspect_ratio": {
    "num": 16,
    "den": 9
  }
}
}}

```

Это сообщение имеет следующие параметры (см. таблицу ниже).

Таблица с описанием параметров

Параметр	Тип данных	Формат	Диапазон возможных значений	Обязательно	Описание
message_type	String		THUMBNAIL_INFO	Да	Тип сообщения. Может принимать следующие значения: <ul style="list-style-type: none"> THUMBNAIL_INFO - информация об эскизе (thumbnail).
task_id	String	UUID		Да	Глобальный идентификатор задачи в формате UUID, например: 30f709e8-5a5a-483e-9bcf-449067e261ef.

status	String		1. OK 2. FAILED	Да	Результат выполнения задачи. Может принимать следующие значения: <ul style="list-style-type: none"> • OK - успешное завершение задачи; • FAILED - перенос файла не удался.
time	Float	Положительное число		Да	Временная позиция видео кадра (в секундах и миллисекундах), который был использован для создания эскиза (thumbnail). Формат — <секунды>.<миллисекунды>, например: 10.0.
error	String	На английском языке		Да, если есть ошибка	Подробное сообщение об ошибке (на английском языке). Поле присутствует, только если значение поля status (статус) имеет значение FAILED (сбой).
errors	Array	Группа параметров		Да, если есть ошибка	Информация о найденных ошибках в виде массива.
errors.error_code	String	7 знаков	1010101 - 1999999	Да, если есть ошибка	Код ошибки.
errors.error_message	String			Да, если есть ошибка	Сообщение об ошибке.
location	Struct	Группа параметров		Да	Блок сведений о расположении файла.
location.provider	String		1. fs 2. s3	Да	Тип хранилища: <ul style="list-style-type: none"> • s3 - объектное хранилище S3 (Object storage S3) — это облачный сервис, позволяющий хранить файлы любого типа и объема; • fs - локальный диск (file system, FS).

location.bucket	String			Да, если предполагается доступ к хранилищу S3	Имя контейнера (bucket) в хранилище S3.
location.s3_region	String			Да, если предполагается доступ к хранилищу S3	Название региона хранилища S3 (если используется это хранилище). Значение, которое должно совпадать с аналогичным значением на стороне хранилища S3. Допустимо вводить любые значения, однако подразумевается, что здесь будет указан код страны (например, ru-central).
location.host_uri	String	host:port		Да, если предполагается доступ к хранилищу S3	Сетевой путь к службе S3 в формате {схема}://{хост IP}[:{порт}].
location.file_uri	String	/file_path		Да, если эскиз (thumbnail) сгенерирован успешно	Путь к файлу в хранилище. <ul style="list-style-type: none"> • для s3 это абсолютный путь к файлу внутри контейнера (bucket); • для fs это либо путь к файлу относительно переменной окружения BASE_PATH, либо абсолютный путь.
frame_size	Struct	Группа параметров		Да	Блок данных о разрешении эскиза (thumbnail).
frame_size.width	Integer	Положительное число		Да	Ширина эскиза (thumbnail) в пикселях.
frame_size.height	Integer	Положительное число		Да	Высота эскиза (thumbnail) в пикселях.

frame_size.display_aspect_ratio	Struct	Группа параметров		Да	DAR (display aspect ratio) кадров. Блок, содержащий данные соотношения сторон эскиза (thumbnail). В итоге результат равен соотношению параметров num/den блока. <ul style="list-style-type: none"> • num - число кадров в единицу времени (количество секунд указано в параметре den); • den - количество секунд, в течение которых отображается количество кадров, определённых в параметре num.
frame_size.display_aspect_ratio.num	Integer	Положительное число		Да	Числитель в соотношении сторон (числитель/знаменатель) эскиза (thumbnail).
frame_size.display_aspect_ratio.den	Integer	Положительное число		Да	Знаменатель в соотношении сторон (числитель/знаменатель) эскиза (thumbnail).
end_timestamp_ms	Integer	Положительное число		Да	Таймстемп завершения формирования миниатюры в миллисекундах с момента Epoch (01.01.1970). Источник - системные ("настенные") часы.

6.8. Результат обработки исходного медиафайла

По завершении обработки видео компонент Transcoder отправляет в Kafka (название очереди KAFKA_TOPIC_TASK_MANAGER) сообщение, предназначенное для компонента Task Manager и содержащее результаты обработки.

Ниже представлено ответное сообщение на Transcoder_Task, описывающее результат работы сервиса Transcoder.

Формат сообщения

```
{
  "message_type": "TRANSCODER_TASK_FINISHED",
```

```
"task_id": "e3338b4a-2771-11ec-9621-0242ac130002",

"status": "FAILED",
"error": "Error database connection",
"errors": [
  {
    "error_code": 1010402,
    "error_message": "Error database connection"
  }
],
"files": [
  {
    "profile_name": "main",
    "location": {
      "provider": "s3",
      "s3_region": "ru",
      "host_uri": "https://test.ru",
      "bucket": "cpp-transcoder",
      "file_uri": "e3338b4a-2771-11ec-9621-0242ac130002/playlist.m3u8"
    },
    "metadata": {
      "mime_type": "application/vnd.apple.mpegurl",
      "file_size": 571,
      "duration": 70.569705,
      "frame_size": {
        "width": 324,
        "height": 576,
        "display_aspect_ratio": {
          "num": 1,
          "den": 1
        }
      }
    }
  }
]
```

```
    },
    "number_of_video_streams": 1,
    "number_of_audio_streams": 1
  }
],

"master_playlist": {
  "location": {
    "provider": "s3",
    "s3_region": "ru",
    "host_uri": "https://test.ru",
    "bucket": "cpp-transcoder",
    "file_uri": "e3338b4a-2771-11ec-9621-0242ac130002/playlist.m3u8"
  }
},

"trickmode": {
  "mime_type": "image/jpeg",
  "canvas_width": 1280,
  "canvas_height": 720,
  "rows": 10,
  "columns": 10,
  "frames": 100,
  "step": 0.5199999809265137,
  "files": [
    {
      "provider": "S3",
      "bucket": "media-platform",
      "s3_region": "ru",
      "host_uri": "https://test.ru",
      "file_uri": "Ds09Yx11ysupcj7l0Isda/T0.jpeg"
    }
  ]
}
```

```

},
"start_timestamp_ms": 1666333264665,
"duration_us": 1192968
}

```

Таблица с описанием параметров

Параметр	Тип данных	Формат	Диапазон возможных значений	Обязательно	Описание
message_type	String		TRANSCODER_TASK_FINISHED	Да	Тип сообщения. Может принимать следующие значения: <ul style="list-style-type: none"> TRANSCODER_TASK_FINISHED - завершение обработки видео.
task_id	String	UUID		Да	Глобальный идентификатор задачи в формате UUID, например: 30f709e8-5a5a-483e-9bcf-449067e261ef.
status	String		<ol style="list-style-type: none"> OK FAILED 	Да	Результат выполнения задачи. Может принимать следующие значения: <ul style="list-style-type: none"> OK - успешное завершение задачи; FAILED - перенос файла не удался.
error	String	На английском языке		Да, если есть ошибка	Подробное сообщение об ошибке (на английском языке). Поле присутствует, только если значение поля status (статус) имеет значение FAILED (сбой).
errors	Array	Группа параметров		Да, если есть ошибка	Массив ошибок.
errors[i].error_code	Integer			Да, если есть ошибка	Код ошибки.

errors[i].error_message	String	На английском языке		Да, если есть ошибка	Текст ошибки.
files	Array	Группа параметров		Да	Массив, содержащий данные об итоговых файлах.
files[i].profile_name	String			Да	Имя профиля, в соответствии с которым был обработан файл (имя профиля берётся из сообщения Transcoder_Task).
files[i].location	Struct.	Группа параметров		Да	Блок сведений о расположении файла.
files[i].location.provider	String		<ol style="list-style-type: none"> 1. s3 2. fs 	Да	<p>Тип хранилища:</p> <ul style="list-style-type: none"> • s3 - объектное хранилище S3 (Object storage S3) — это облачный сервис, позволяющий хранить файлы любого типа и объема; • fs - локальный диск (file system, FS).
files[i].location.bucket	String			Да, если предполагается доступ к хранилищу S3	Имя контейнера (bucket) в хранилище S3.

files[i].location.s3_region	String			Да, если предполагается доступ к хранилищу S3	Название региона хранилища S3 (если используется это хранилище). Значение, которое должно совпадать с аналогичным значением на стороне хранилища S3. Допустимо вводить любые значения, однако подразумевается, что здесь будет указан код страны (например, ru-central).
files[i].location.host_uri	String	host:port		Да, если предполагается доступ к хранилищу S3	Сетевой путь к службе S3 в формате {схема}://{хост IP}[:{порт}].
files[i].location.file_uri	String	/file_path		Да, если обработка файла с использованием профиля прошла успешно	Путь к файлу в хранилище. <ul style="list-style-type: none"> • для s3 это абсолютный путь к файлу внутри контейнера (bucket); • для fs это либо путь к файлу относительно переменной окружения BASE_PATH, либо абсолютный путь.
files[i].metadata	String	Группа параметров		Да	Блок метаданных медиафайла.
files[i].metadata.mime_type	String			Да	МIME-тип файла.
files[i].metadata.file_size	Integer		1 - 1 073 741 824	Да	Размер файла в байтах. Максимальный размер загружаемого файла - 1 Гб.

files[i].metadata.duration	Float	s.ms		Да	Время видео/аудиодорожки в формате <секунды>.<миллисекунды>. Пример: 19.734.
files[i].metadata.frame_size	Struct	Группа параметров		Да	Блок данных о разрешении видеокадра.
files[i].metadata.frame_size.width	Integer	Положительное число		Да	Итоговая ширина кадра в пикселях. Может отличаться от запрошенного значения в сообщении TRANSCODER_TASK.
files[i].metadata.frame_size.height	Integer	Положительное число		Да	Итоговая высота кадра в пикселях. Может отличаться от запрошенного значения в сообщении TRANSCODER_TASK.
files[i].metadata.frame_size.display_aspect_ratio	Struct	Группа параметров		Да	Блок, содержащий данные соотношения сторон кадра видео. В итоге результат равен соотношению параметров num/den блока.
files[i].metadata.frame_size.display_aspect_ratio.num	Integer	Положительное число		Да	Числитель, в соотношении сторон видео — числитель/знаменатель.
files[i].metadata.frame_size.display_aspect_ratio.den	Integer	Положительное число		Да	Знаменатель в соотношении сторон видео — числитель/знаменатель.
files[i].number_of_video_streams	Integer	Положительное число		Да	Число видеопотоков в файле.
files[i].number_of_audio_streams	Integer	Положительное число		Да	Число аудиопотоков в файле.

master_playlist	Struct	Группа параметров		Нет	Блок сведения о мастер-плейлисте HLS. Присутствует, только если в сообщении TRANSCODER_TASK была задана опция генерации мастер-плейлиста HLS.
master_playlist.location	Struct	Группа параметров		Да, если присутствует элемент master_playlist	Блок сведений о расположении файла.
master_playlist.location.provider	String		<ol style="list-style-type: none"> 1. s3 2. fs 	Да, если присутствует элемент location	<p>Тип хранилища:</p> <ul style="list-style-type: none"> • s3 - объектное хранилище S3 (Object storage S3) — это облачный сервис, позволяющий хранить файлы любого типа и объема; • fs - локальный диск (file system, FS).
master_playlist.location.bucket	String			Да, при использовании хранилища S3	Имя контейнера (bucket) в хранилище S3.

master_playlist.location.s3_region	String			Да, при использовании хранилища S3	Название региона хранилища S3 (если используется это хранилище). Значение, которое должно совпадать с аналогичным значением на стороне хранилища S3. Допустимо вводить любые значения, однако подразумевается, что здесь будет указан код страны (например, ru-central).
master_playlist.location.host_uri	String	host:port		Да, при использовании хранилища S3	Сетевой путь к службе S3 в формате {схема}://{хост IP}[:{порт}].
master_playlist.location.file_uri	String	/file_path		Да, если плейлист успешно сгенерирован	Путь к файлу в хранилище. <ul style="list-style-type: none"> • для s3 это абсолютный путь к файлу внутри контейнера (bucket); • для fs это либо путь к файлу относительно переменной окружения BASE_PATH, либо абсолютный путь.
trickmode	Struct	Группа параметров		Да, если задача на обработку содержала блок trick mode	Блок информации о сгенерированных наборах миниатюр для ознакомительного просмотра.

trickmode.mime_type	String			Да, если задача на обработку содержала блок trick mode	МIME-тип файла.
trickmode.canvas_width	Integer	Положительное число		Да, если задача на обработку содержала блок trick mode	Ширина холста, содержащего эскизы для ознакомительного просмотра (в пикселях).
trickmode.canvas_height	Integer	Положительное число		Да, если задача на обработку содержала блок trick mode	Высота холста, содержащего эскизы для ознакомительного просмотра (в пикселях).
trickmode.rows	Integer	От 1		Да, если задача на обработку содержала блок trick mode	Количество рядов с миниатюрами на холсте.
trickmode.columns	Integer	От 1		Да, если задача на обработку содержала блок trick mode	Количество столбцов с эскизами на холсте для ознакомительного просмотра (т. е. количество миниатюр в одном ряду).

trickmode.frames	Integer	Положительное число		Да, если задача на обработку содержала блок <code>trickmode</code>	Общее количество сгенерированных эскизов для ознакомительного просмотра по всем холстам (результатирующим файлам).
trickmode.step	Float		Логично, если <code>step</code> равен длине сегмента видео (чанка), т.е. 2 секунды.	Да, если задача на обработку содержала блок <code>trickmode</code>	Шаг в секундах между двумя соседними эскизами для ознакомительного просмотра.
trickmode.files	Array	Группа параметров		Да, если задача на обработку содержала блок <code>trickmode</code>	Массив, содержащий данные об итоговых файлах.
trickmode.files[i].provider	String		<ol style="list-style-type: none"> s3 fs 	Да, если задача на обработку содержала блок <code>trickmode</code>	<p>Тип хранилища:</p> <ul style="list-style-type: none"> s3 - объектное хранилище S3 (Object storage S3) — это облачный сервис, позволяющий хранить файлы любого типа и объема; fs - локальный диск (file system, FS).
trickmode.files[i].bucket	String			Да, если задача на обработку содержала блок <code>trickmode</code>	Имя контейнера (bucket) в хранилище S3.

trickmode.files[i].s3_region	String			Да, если задача на обработку содержала блок <code>trickmode</code>	Название региона хранилища S3 (если используется это хранилище). Значение, которое должно совпадать с аналогичным значением на стороне хранилища S3. Допустимо вводить любые значения, однако подразумевается, что здесь будет указан код страны (например, <code>ru-central</code>).
trickmode.files[i].host_uri	String	host:port		Да, если задача на обработку содержала блок <code>trickmode</code>	Сетевой путь к службе S3 в формате <code>{схема}://{хост IP}[:{порт}]</code> .
trickmode.files[i].file_uri	String	/file_path		Да, если задача на обработку содержала блок <code>trickmode</code>	Путь к файлу в хранилище. <ul style="list-style-type: none"> • для <code>s3</code> это абсолютный путь к файлу внутри контейнера (<code>bucket</code>); • для <code>fs</code> это либо путь к файлу относительно переменной окружения <code>BAS E_PATH</code>, либо абсолютный путь.

start_timestamp_ms	Integer			Да	<p>Отметка времени начала обработки, выраженная в UNIX-time, то есть в миллисекундах, прошедших с 1 января 1970 года. Замеряется при помощи системных часов. Подвержена коррекциям системного времени:</p> <ul style="list-style-type: none"> • NTP, Network Time Protocol (протокол системного времени); • leap second (секунда координации) • и так далее.
duration_us	Integer			Да	<p>Продолжительность обработки, выраженная в микросекундах. Замеряется при помощи монотонных часов и не подвержена коррекциям системного времени.</p>

6.9. Сообщения о статусах обработки

Task Manager при получении сведений о начале или завершении очередного этапа обработки видео отправляет сообщение о соответствующем статусе в топик Kafka KAFKA_TOPIC_STATUS, откуда их могут прочитать внешние пользователи Медиаплатформы.

Ниже приведены примеры сообщений о статусах:

- [Начало загрузки видео \(UploaderStarted\)](#)
- [Завершение загрузки видео \(UploaderFinished\)](#)
- [Сбой загрузки видео \(UploaderFailed\)](#)
- [Завершение анализа видео \(ProbeFinished\)](#)
- [Сбой анализа видео \(ProbeFailed\)](#)
- [Создание эскиза - thumbnail видео \(ThumbnailReady\)](#)
- [Сбой создания эскиза - thumbnail \(ThumbnailFailed\)](#)

- [Завершение обработки видео \(TranscoderFinished\)](#)
- [Сбой обработки видео \(TranscoderFailed\)](#)

6.9.1.1. Начало загрузки видео

UploaderStarted

```
{  
  "message_type": "TASK_STATUS",  
  "status": "UploaderStarted",  
  "task_id": "e3338b4a-2771-11ec-9621-0242ac130002",  
  "timestamp": 1649765708,  
  "utc_time": "2022-04-12T12:15:08Z"  
}
```

6.9.1.2. Завершение загрузки видео

UploaderFinished

```
{  
  "message_type": "TASK_STATUS",  
  "status": "UploaderFinished",  
  "task_id": "e3338b4a-2771-11ec-9621-0242ac130002",  
  "timestamp": 1649765730,  
  "utc_time": "2022-04-12T12:15:30Z"  
}
```

6.9.1.3. Сбой загрузки видео

UploaderFailed

```
{  
  "message_type": "TASK_STATUS",  
  "task_id": "e3338b4a-2771-11ec-9621-0242ac130002",  
  "status": "UploaderFailed",  
  "timestamp": 1257894000,  
  "utc_time": "Tue Nov 10 2009 23:00:00 GMT+0000",  
  "error": "Error database connection",  
  "errors": [  
    {
```

```
    "error_code": 1010402,
    "error_message": "Error database connection"
  }
]
}
```

6.9.1.4. Завершение анализа видео

ProbeFinished

```
{
  "message_type": "TASK_STATUS",
  "status": "ProbeFinished",
  "task_id": "e3338b4a-2771-11ec-9621-0242ac130002",
  "timestamp": 1649765731,
  "utc_time": "2022-04-12T12:15:31Z"
}
```

6.9.1.5. Сбой анализа видео

ProbeFailed

```
{
  "message_type": "TASK_STATUS",
  "task_id": "e3338b4a-2771-11ec-9621-0242ac130002",
  "status": "ProbeFailed",
  "timestamp": 1257894000,
  "utc_time": "Tue Nov 10 2009 23:00:00 GMT+0000",
  "error": "Error database connection",
  "errors": [
    {
      "error_code": 1010402,
      "error_message": "Error database connection"
    }
  ]
}
```

6.9.1.6. Создание эскиза (thumbnail) видео

ThumbnailReady

```
{  
  "message_type": "TASK_STATUS",  
  "status": "ThumbnailReady",  
  "task_id": "e3338b4a-2771-11ec-9621-0242ac130002",  
  "timestamp": 1649765734,  
  "utc_time": "2022-04-12T12:15:34Z",  
  "thumbnail_location": "https://test.ru/e3338b4a-2771-11ec-9621-0242ac130002/thumbnail.jpg",  
  "thumbnail_width": 1280,  
  "thumbnail_height": 720  
}
```

6.9.1.7. Сбой создания эскиза (thumbnail)

ThumbnailFailed

```
{  
  "message_type": "TASK_STATUS",  
  "task_id": "e3338b4a-2771-11ec-9621-0242ac130002",  
  "status": "ThumbnailFailed",  
  "timestamp": 1257894000,  
  "utc_time": "Tue Nov 10 2009 23:00:00 GMT+0000",  
  "error": "Error database connection",  
  "errors": [  
    {  
      "error_code": 1010402,  
      "error_message": "Error database connection"  
    }  
  ]  
}
```

6.9.1.8. Завершение обработки видео

TranscoderFinished

```
{
```

```

"message_type": "TASK_STATUS",
"status": "TranscoderFinished",
"task_id": "e3338b4a-2771-11ec-9621-0242ac130002",
"timestamp": 1649765738,
"utc_time": "2022-04-12T12:15:38Z",
"hls_url": "https://test.ru/T1jvAjDEZJmK1n0HRP86I/playlist.m3u8",
"video_duration": 26.280000686645508,
"max_width": 1920,
"max_height": 1080,
  "trickmode": {
    "mime_type": "image/jpeg",
    "canvas_width": 1280,
    "canvas_height": 720,
    "rows": 10,
    "columns": 10,
    "frames": 125,
    "step": 1.23543987566473,
    "files": [
      "https://test.ru/T1jvAjDEZJmK1n0HRP86I/T0.jpeg",
      "https://test.ru/T1jvAjDEZJmK1n0HRP86I/T1.jpeg"
    ]
  }
}

```

6.9.1.9. Сбой обработки видео

TranscoderFailed

```

{
"message_type": "TASK_STATUS",
"task_id": "e3338b4a-2771-11ec-9621-0242ac130002",
"status": "TranscoderFailed",
"timestamp": 1257894000,
"utc_time": "Tue Nov 10 2009 23:00:00 GMT+0000",
"error": "Error database connection",

```

```

"errors": [
  {
    "error_code": 1010402,
    "error_message": "Error database connection"
  }
]

```

Таблица с описанием параметров

Параметр	Тип данных	Формат	Диапазон возможных значений	Обязательно	Описание
message_type	String		TASK_STATUS	Да	Тип сообщения. Может принимать следующие значения: <ul style="list-style-type: none"> TASK_STATUS - завершение обработки видео.
task_id	String	UUID		Да	Глобальный идентификатор задачи в формате UUID, например: 30f709e8-5a5a-483e-9bcf-449067e261ef.
status	String		<ol style="list-style-type: none"> OK FAILED 	Да	Результат выполнения задачи. Может принимать следующие значения: <ul style="list-style-type: none"> OK - успешное завершение задачи; FAILED - перенос файла не удался.
error	String	На английском языке		Да, если есть ошибка	Подробное сообщение об ошибке (на английском языке). Поле присутствует, только если значение поля status (статус) имеет значение FAILED (сбой).
errors.[i]	Array	Группа параметров		Да, если есть ошибка	Информация о найденных ошибках в виде массива.

errors[i].error_code	Integer			Да, если есть ошибка	Код ошибки.
errors[i].error_message	String	На английском языке		Да, если есть ошибка	Подробное сообщение об ошибке (на английском языке). Поле присутствует, только если значение поля status (статус) имеет значение FAILED (сбой).
files	Array	Группа параметров		Да	Массив, содержащий данные об итоговых файлах.
files[i].profile_name	String			Да	Имя профиля, в соответствии с которым был обработан файл (имя профиля берётся из сообщения Transcoder_Task).
files[i].location	Struct	Группа параметров		Да	Блок сведений о расположении файла.
files[i].location.provider	String		<ol style="list-style-type: none"> 1. s3 2. fs 	Да	<p>Тип хранилища:</p> <ul style="list-style-type: none"> • s3 - объектное хранилище S3 (Object storage S3) — это облачный сервис, позволяющий хранить файлы любого типа и объема; • fs - локальный диск (file system, FS).
files[i].location.bucket	String			Да, если предполагается доступ к хранилищу S3	Имя контейнера (bucket) в хранилище S3.

files[i].location.s3_region	String			Да, если предполагается доступ к хранилищу S3	Название региона хранилища S3 (если используется это хранилище). Значение, которое должно совпадать с аналогичным значением на стороне хранилища S3. Допустимо вводить любые значения, однако подразумевается, что здесь будет указан код страны (например, ru-central).
files[i].location.host_uri	String	host:port		Да, если предполагается доступ к хранилищу S3	Сетевой путь к службе S3 в формате {схема}://{хост IP}:{порт}.
files[i].location.file_uri	String	/file_path		Да, если обработка файла с использованием профиля прошла успешно	Путь к файлу в хранилище. <ul style="list-style-type: none"> • для s3 это абсолютный путь к файлу внутри контейнера (bucket); • для fs это либо путь к файлу относительно переменной окружения BASE_PATH, либо абсолютный путь.
files[i].metadata	Struct	Группа параметров		Да	Блок метаданных медиафайла.
files[i].metadata.mime_type	String			Да	МIME-тип файла.
files[i].metadata.file_size	Integer		1 - 1 073 741 824	Да	Размер файла в байтах. Максимальный размер загружаемого файла - 1 Гб.
files[i].metadata.duration	Float	s.ms		Да	Время видео/аудиодорожки в формате <секунды>.<миллисекунды>, например: 19.734.

files[i].metadata.frame_size	Struct	Группа параметров		Да	Блок данных о разрешении видеокadra.
files[i].metadata.frame_size.width	Integer	Положительное число		Да	Итоговая ширина кадра в пикселях. Может отличаться от запрошенного значения в сообщении TRANSCODER_TASK.
files[i].metadata.frame_size.height	Integer	Положительное число		Да	Итоговая высота кадра в пикселях. Может отличаться от запрошенного значения в сообщении TRANSCODER_TASK.
files[i].metadata.frame_size.display_aspect_ratio	Struct	Группа параметров		Да	Блок, содержащий данные соотношения сторон кадра видео. В итоге результат равен соотношению параметров num/den блока.
files[i].metadata.frame_size.display_aspect_ratio.num	Integer	Положительное число		Да	Числитель, в соотношении сторон видео — числитель/знаменатель.
files[i].metadata.frame_size.display_aspect_ratio.den	Integer	Положительное число		Да	Знаменатель в соотношении сторон видео — числитель/знаменатель.
files[i].number_of_video_streams	Integer	Положительное число		Да	Число видеопотоков в файле.
files[i].number_of_audio_streams	Integer	Положительное число		Да	Число аудиопотоков в файле.
master_playlist	Struct	Группа параметров		Нет	Блок сведения о мастер-плейлисте HLS. Присутствует, только если в сообщении TRANSCODER_TASK была задана опция генерации мастер-плейлиста HLS.

master_playlist.location	Struct	Группа параметров		Да, если присутствует элемент master_playlist	Блок сведений о расположении файла.
master_playlist.location.provider	String			1. s3 2. fs Да, если присутствует элемент location	Тип хранилища: <ul style="list-style-type: none"> s3 - объектное хранилище S3 (Object storage S3) — это облачный сервис, позволяющий хранить файлы любого типа и объема; fs - локальный диск (file system, FS).
master_playlist.location.bucket	String			Да, при использовании хранилища S3	Имя контейнера (bucket) в хранилище S3.
master_playlist.location.s3_region	String			Да, при использовании хранилища S3	Название региона хранилища S3 (если используется это хранилище). Значение, которое должно совпадать с аналогичным значением на стороне хранилища S3. Допустимо вводить любые значения, однако подразумевается, что здесь будет указан код страны (например, ru-central).
master_playlist.location.host_uri	String	host:port		Да, при использовании хранилища S3	Сетевой путь к службе S3 в формате {схема}://{хост IP}[:{порт}].

master_playlist.location.file_uri	String	/file_path		Да, если плейлист успешно сгенерирован	Путь к файлу в хранилище. <ul style="list-style-type: none"> • для s3 это абсолютный путь к файлу внутри контейнера (bucket); • для fs это либо путь к файлу относительно переменной окружения BASE_PATH, либо абсолютный путь.
trickmode	Struct	Группа параметров		Да, если задача на обработку содержала блок trick mode	Блок информации о сгенерированных наборах миниатюр для ознакомительного просмотра.
trickmode.mime_type	String			Да, если задача на обработку содержала блок trick mode	МИМЕ-тип файла.
trickmode.canvas_width	Integer			Да, если задача на обработку содержала блок trick mode	Ширина холста, содержащего эскизы для ознакомительного просмотра (в пикселях).
trickmode.canvas_height	Integer			Да, если задача на обработку содержала блок trick mode	Высота холста, содержащего эскизы для ознакомительного просмотра (в пикселях).
trickmode.rows	Integer	от 1		Да, если задача на обработку содержала блок trick mode	Количество рядов с миниатюрами на холсте.

trickmode.columns	Integer	от 1		Да, если задача на обработку содержала блок <code>trick mode</code>	Количество столбцов с эскизами на холсте для ознакомительного просмотра (т. е. количество миниатюр в одном ряду).
trickmode.frames	Integer			Да, если задача на обработку содержала блок <code>trick mode</code>	Общее количество сгенерированных эскизов для ознакомительного просмотра по всем холстам (результатирующим файлам).
trickmode.step	Float		Логично, если <code>step</code> равен длине сегмента видео (чанка), т.е. 2 секунды.	Да, если задача на обработку содержала блок <code>trick mode</code>	Шаг в секундах между двумя соседними эскизами для ознакомительного просмотра.
trickmode.files	Array	Группа параметров		Да, если задача на обработку содержала блок <code>trick mode</code>	Массив, содержащий данные об итоговых файлах.
trickmode.files[i].provider	String		<ol style="list-style-type: none"> 1. s3 2. fs 	Да, если задача на обработку содержала блок <code>trick mode</code>	<p>Тип хранилища:</p> <ul style="list-style-type: none"> • s3 - объектное хранилище S3 (Object storage S3) — это облачный сервис, позволяющий хранить файлы любого типа и объема; • fs - локальный диск (file system, FS).
trickmode.files[i].bucket	String			Да, если задача на обработку содержала блок <code>trick mode</code>	Имя контейнера (bucket) в хранилище S3.

trickmode.files[i].s3_region	String			Да, если задача на обработку содержала блок trick mode	Название региона хранилища S3 (если используется это хранилище). Значение, которое должно совпадать с аналогичным значением на стороне хранилища S3. Допустимо вводить любые значения, однако подразумевается, что здесь будет указан код страны (например, ru-central).
trickmode.files[i].host_uri	String	host:port		Да, если задача на обработку содержала блок trick mode	Сетевой путь к службе S3 в формате {схема}://{хост IP}:{порт}].
trickmode.files[i].file_uri	String	/file_path		Да, если задача на обработку содержала блок trick mode	Путь к файлу в хранилище. <ul style="list-style-type: none"> • для s3 это абсолютный путь к файлу внутри контейнера (bucket); • для fs это либо путь к файлу относительно переменной окружения BASE_PATH, либо абсолютный путь.

start_timestamp_ms	Integer			Да	<p>Отметка времени начала обработки, выраженная в UNIX-time, то есть в миллисекундах, прошедших с 1 января 1970 года. Замеряется при помощи системных часов. Подвержена коррекциям системного времени:</p> <ul style="list-style-type: none"> • NTP, Network Time Protocol (протокол системного времени); • leap second (секунда координации) • и так далее.
duration_us	Integer			Да	<p>Продолжительность обработки, выраженная в микросекундах. Замеряется при помощи монотонных часов и не подвержена коррекциям системного времени.</p>

6.10. Сообщения об ошибках

6.10.1. Составной ключ ошибки

1XXYYZZ, где XX - код модуля, YY - код группы ошибок, ZZ - код ошибки

1	XX	YY	ZZ
Ненулевой символ для сохранения разрядности	Код сервиса/модуля	Код группы ошибок	Код ошибки

6.10.1.1. Код модуля

- 01 - Uploader
- 02 - TaskManager
- 03 - Prober
- 04 - Transcoder

6.10.1.2. Код группы ошибок

- 01 - падение модулей

02 - внутренние ошибки модуля, зависящие от исполнения задания

03 - сообщения

04 - связь, сеть, ошибки подключения

6.10.2. Пример (ошибка 1010101)

	код модуля	код группы ошибок	код ошибки
1	01	01	01
	Uploader	Падение модулей	Ошибка работы Uploader

6.10.3. Ошибки Медиаплатформы

Составной ключ ошибки	Сообщение об ошибке	Описание ошибки	Возможные причины ошибок
1010201	Uploader. Internal error	Uploader. Внутренняя ошибка	Любая ошибка модуля Uploader, не поддающаяся описанной классификации.
1010202	Uploader. Error append record chunk database	Uploader. Ошибка добавления информации о загруженном чанке в БД	Отсутствует подключение к базе данных.
1010203	Uploader. Error upload multipart chunk	Uploader. Ошибка загрузки очередного чанка в хранилище S3	Подробную информацию необходимо смотреть в поле error.
1010204	Uploader. Error upload file to S3	Uploader. Ошибка при передаче медиаконтента в хранилище S3	

1010205	Uploader. Error GetPartByOffset	Uploader. Ошибка получения номера загружаемой части по смещению	<ul style="list-style-type: none"> • При создании задачи для загрузки параметр partsize равен 0. • Попытка загрузить данные со смещением, превышающим размер файла или меньшим нуля. <p>Смещение / offset - это метка в файле, с которой начинается загрузка чанка. При начале загрузки файла смещение равно 0.</p>
1010206	Uploader. Error upload file from front	Uploader. Ошибка загрузки файла	Ошибка загрузки файла с Frontend.
1010208	Uploader. File size exceeded: {size_file} > {size_max}	Uploader. Превышен размер файла	
1010209	Uploader. Chunk size less than minimum: {size_chunk} < {size_chunk_min}	Uploader. Указанный размер чанка меньше минимального	
1010210	Uploader. Error adding new task file upload	Uploader. Ошибка добавления новой задачи для загрузки файла	
1010211	Uploader. Error timeout upload file Uploader.	Превышено время, отведенное на загрузку файла	
1010212	Uploader. Error updating in DB status of message sent to Kafka	Uploader. Ошибка обновления статуса послыки сообщения в Kafka	Kafka недоступна во время работы механизма перепосылки сообщений
1010213	Uploader. Error execution DB Transaction	Uploader. Ошибка выполнения транзакции в базе данных	Транзакция (изменение) в БД выполнена не полностью
1010214	Uploader. Error creating upload task to S3 storage	Uploader. Ошибка создания задания на загрузку в S3 хранилище	
1010301	Uploader. Validation of received messages failed	Uploader. Не пройдена валидация полученных сообщений	
1010402	Uploader. Error database connection	Uploader. Ошибка подключения к базе данных	Не доступен сервер базы данных.

1010403	Uploader. Error S3 connect. Error code S3: {error_code_S3}	Uploader. Ошибка подключения к хранилищу S3	<ul style="list-style-type: none"> • Не доступно хранилище S3. • Неверные параметры подключения к S3.
1010404	Uploader. Error sending message to Kafka	Ошибка отправки сообщения в Kafka	Не работает Kafka
Составной ключ ошибки	Сообщение об ошибке	Описание ошибки	Возможные причины ошибок
1020201	TaskManager. Internal error	TaskManager. Внутренняя ошибка	<ul style="list-style-type: none"> • Любая ошибка модуля TaskManager, не поддающаяся описанной классификации. • Нулевой fps видеопотока. • Не найдены предпочтительные дорожки.
1020202	TaskManager. File duration exceeded: {duration_file} > {duration_max}	TaskManager. Превышена длительность файла	Проверка после обработки модулем Prober.
1020203	TaskManager. Frame resolution is greater than maximum: {resolution_file} > {resolution_max}	TaskManager. Разрешение кадра больше максимального	Проверка после обработки модулем Prober.
1020204	TaskManager. Frame resolution less than minimum: {resolution_file} < {resolution_min}	TaskManager. Разрешение кадра меньше минимального	Проверка после обработки модулем Prober.
1020205	TaskManager. The specified container cannot be processed: {container}	TaskManager. Заданный контейнер не подлежит обработке	Проверка после обработки модулем Prober.
1020206	TaskManager. The specified codec cannot be processed: {codec}	TaskManager. Заданный кодек не подлежит обработке	Проверка после обработки модулем Prober.
1020207	TaskManager. Video bitrate less than minimum: {video_bitrate_file} < {video_bitrate_min}	TaskManager. Битрейт видео меньше минимального	Проверка после обработки модулем Prober.

1020208	TaskManager. Audio bitrate less than minimum: {audio_bitrate} < {audio_bitrate_min}	TaskManager. Битрейт аудио меньше минимального	Проверка после обработки модулем Prober.
1020209	TaskManager. Video bitrate more than maximum: {video_bitrate_file} > {video_bitrate_max}	TaskManager. Битрейт видео больше максимального	Проверка после обработки модулем Prober.
1020210	TaskManager. Audio bitrate more than maximum: {audio_bitrate_file} > {audio_bitrate_max}	TaskManager. Битрейт аудио больше максимального	Проверка после обработки модулем Prober.
Составной ключ ошибки	Сообщение об ошибке	Описание ошибки	Возможные причины ошибок
1030201	Prober. Internal error	Prober. Внутренняя ошибка в процессе анализа исходного файла	Любая ошибка модуля Prober, не поддающаяся описанной классификации.
1030202	Prober. Failed to initialize	Prober. Не удалось запустить анализ исходного файла	<ul style="list-style-type: none"> • В задании указано использовать GPU, а GPU нет. • Невозможно определить лучшую дорожку для обработки и записи в результирующий файл (при указании такого способа выбора). • Другие ошибки.
1030203	Prober. Unknown codec: {codec}	Prober. Неизвестный кодек	
1030204	Prober. Error decoding. Cannot detect stream {stream_idx} duration Prober. Error decoding. Cannot open stream {stream_idx} decoder Prober. Error decoding. Cannot decode stream {stream_idx}	Prober. Ошибка декодирования	<ul style="list-style-type: none"> • Стрим не декодирован полностью. • Соответствующий декодер не найден. • Начало стрима не декодировано.

1030205	Prober. Error Demultiplexing	Prober. Ошибка демультимплексирования исходного файла	<ul style="list-style-type: none"> • Файл имеет повреждения некоторых сегментов. • Не все пакеты были прочитаны.
1030206	Prober. Reject suspicious file	Prober. Отклонение подозрительного файла	<ul style="list-style-type: none"> • Расчетная скорость обработки файла слишком низкая (<1 б/сек). • Расчетная длительность обработки слишком высокая (>30 часов).
1030301	Prober. Validation of received messages failed	Prober. Не пройдена валидация полученных сообщений	<ul style="list-style-type: none"> • Если является причиной невозможности анализа видео. • Видеофайл без видеодорожки.
1030401	Prober. S3 connect error	Prober. Ошибка подключения к хранилищу S3	<ul style="list-style-type: none"> • Не доступно хранилище S3. • Неверные параметры подключения к S3.
1030402	Prober. S3 file not found error	Prober. Отсутствие указанного файла на S3	
1030403	Prober. S3 empty file error	Prober. Файл нулевой длины на S3	
1030404	Prober. S3 read error	Prober. Ошибка чтение из S3	Потеря доступности S3 хранилища.
Составной ключ ошибки	Сообщение об ошибке	Описание ошибки	Возможные причины ошибок
1040201	Transcoder. Internal error	Transcoder. Внутренняя ошибка в процессе транскодирования файла	Любая ошибка модуля Transcoder, не поддающаяся описанной классификации.

1040202	Transcoder. Failed to initialize transcode	Transcoder. Не удалось запустить транскодирование видео	<ul style="list-style-type: none"> • В задании указано использовать GPU, а GPU нет. • Невозможно определить лучшую дорожку для обработки и записи в результирующий файл (при указании такого способа выбора). • Другие ошибки.
1040204	Transcoder. Error decoding. Stream index: {stream_idx}	Transcoder. Ошибка декодирования	<ul style="list-style-type: none"> • Файл не декодирован полностью. • Соответствующий декодер не найден. • Начало стрима не декодировано. • Файл имеет повреждения некоторых сегментов.
1040205	Transcoder. Error demultiplexing	Transcoder. Ошибка демультиплексирования файла	<ul style="list-style-type: none"> • Файл имеет повреждения некоторых сегментов; ошибка не обнаружена в модуле Prober. • Не все пакеты были прочитаны; ошибка не обнаружена в модуле Prober.
1040206	Transcoder. Error encoding. Profile {profile_name} Transcoder. Error encoding (init). Profile {profile_name} Transcoder. Error encoding (flush). Profile {profile_name} Transcoder. Error encoding (encode). Profile {profile_name}	Transcoder. Ошибка кодирования результирующего файла по заданному профилю	Невозможно применить параметры из настроек профиля обработки файла.
1040207	Transcoder. Thumbnail error	Transcoder. Ошибка в процессе генерации миниатюры	Невозможно применить параметры из настроек генерации миниатюры.

1040208	Transcoder. Invalid timestamp. Thumbnails process	Transcoder. Ошибка временной метки для создания миниатюры	Заданная временная метка для создания миниатюры больше, чем длина видео.
1040209	Transcoder. Error. Trickmode process Transcoder. Error. Trickmode process (flushing) Transcoder. Error. Trickmode process (non-monotonic)	Transcoder. Ошибка в процессе формирования холстов с миниатюрами для таймлайна	<ul style="list-style-type: none"> Невозможно применить параметры из настроек формирования холстов с миниатюрами для таймлайна. Не удалось сохранить json со списком холстов, параметры холстов с миниатюрами для таймлайна. Ошибка сохранения холстов с миниатюрами для таймлайна.
1040210	Transcoder. Error generation master playlist	Transcoder. Ошибка формирования плейлиста исходного файла	<ul style="list-style-type: none"> Неверно указан путь к хранилищу. При запросе доступа к хранилищу вернулась ошибка.
1040211	Transcoder. Unknown encoder for codec {codec}	Transcoder. При работе не получилось создать энкодер, указанный в конфигурационном сообщении.	Шаблон конфигурации для Task Manager составлен некорректно.
1040301	Transcoder. Validation of received messages failed	Transcoder. Не пройдена валидация полученных сообщений	
1040401	Transcoder. S3 connect error	Transcoder. Ошибка подключения к хранилищу S3	<ul style="list-style-type: none"> Не доступно хранилище S3. Неверные параметры подключения к S3.
1040402	Transcoder. S3 file not found error	Transcoder. Отсутствие указанного файла на S3	
1040403	Transcoder. S3 empty file error	Transcoder. Файл нулевой длины на S3	
1040404	Transcoder. S3 read error	Transcoder. Ошибка чтения из S3	Проблемы чтения из S3 или проблемы записи в S3.

Формат сообщения

```
{
  "message_type": "TASK_STATUS",
  "task_id": "e3338b4a-2771-11ec-9621-0242ac130002",
```

```

"status": "UploaderFailed",
"timestamp": 1257894000,
"utc_time": "2022-04-12T12:15:31Z",
"error": "Error database connection",
"errors": [
  {
    "error_code": 1010402,
    "error_message": "Error database connection"
  },
  {
    "error_code": 1010402,
    "error_message": "Error database connection"
  }
]
}

```

Описание параметров

Параметр	Тип	Формат	Диапазон возможных значений	Обязательность использования	Описание
message_type	String		TASK_STATUS	Да	Тип сообщения
task_id	String	UUID		Да	Глобальный идентификатор задачи в формате UUID, например: 30f709e8-5a5a-483e-9bcf-449067e261ef.

<p>status</p>	<p>String</p>		<ul style="list-style-type: none"> • UploaderStarted • UploaderFinished • UploaderFailed • ProberFinished • ProberFailed • ThumbnailReady • ThumbnailFailed • TranscoderFinished • TranscoderFailed 	<p>Да</p>	<p>Фактический статус выполнения задачи обработки видео файла.</p> <ul style="list-style-type: none"> • UploaderStarted - начата загрузка исходного видеофайла в S3 хранилище. • UploaderFinished - загрузка исходного видеофайла в S3 хранилище окончена. • UploaderFailed - ошибка при загрузке исходного видеофайла в S3 хранилище. • ProberFinished - анализ исходного видеофайла закончен. • ProberFailed - ошибка при анализе исходного видеофайла. • ThumbnailReady - готова миниатюра видео для исходного видеофайла. • ThumbnailFailed - ошибка при подготовке миниатюры видео для исходного видеофайла. • TranscoderFinished - окончено транскодирование. • TranscoderFailed - ошибка транскодирования.
----------------------	---------------	--	--	-----------	--

timestamp		unix-time		Да	Временная метка смены статуса в UNIX-time, т.е. количество секунд, прошедших с полуночи (00:00:00 UTC) 1 января 1970 года.
utc_time		RFC3339		Да	Формат временной метки timestamp для удобства отладки.
errors				Да, если есть ошибка	Сообщение об ошибке.
error_code	Integer	7 знаков	1010101 - 1999999	Да, если есть ошибка	Составной ключ ошибки.
error_message	String	Текст на англ. яз.		Нет	Сообщение об ошибке, на английском языке. Достаточно подробное, чтобы идентифицировать источник проблемы. Поле присутствует в сообщении только если значение поля status равно UploaderFailed, ProberFailed, TranscoderFailed или ThumbnailFailed.

6.11. Ограничения на загружаемое видео

Ограничения

- Длительность: не более 12 часов.
- Размер: не более 24 Гб.
- Максимальное разрешение: 4200x2160 px (горизонтально или вертикально).
- Минимальное разрешение: 128x128 px.
- Минимальное количество видеодорожек: 1 (или определение профиля, поддерживающие аудиофайлы без видео дорожек).

Белый список

- Контейнеры: MOV, MP4, MPG, AVI, FLV, 3GPP, WebM, MPEGPS, OGG, MKV, TS.
- Кодеки видео:
 - ProRes;
 - HEVC (H.265) - есть поддержка GPU декодирования (T4+);
 - MPEG-1 - есть поддержка GPU декодирования (T4+);

- MPEG-2 (H.262) - есть поддержка GPU декодирования (T4+);
- WMV;
- AVC (H.264) - есть поддержка GPU декодирования (T4+);
- Theora;
- VP-8 есть поддержка GPU декодирования (T4+);
- VP-9 есть поддержка GPU декодирования (T4+);
- VC-1 есть поддержка GPU декодирования (T4+).
- Кодеки аудио: AAC LC, HE-AAC, AC3, E-AC3, MP3, Opus, Vorbis, WMA, PCM.

Белый список - самые распространенные кодеки и форматы. Видео- и медиафайлы проходят тестирование и обработку, согласно этому списку.

Список был сформирован в результате анализа аналогичных платформ.

7. Процедуры проверки работоспособности и действия при сбоях

7.1. Процедуры проверки работоспособности

7.1.1. Проверка работоспособности сервисов Медиаплатформы после развертывания

Определение рабочего статуса Медиаплатформы, не создавая задания.

Для проверки нахождения стенда в работоспособном состоянии нужно в rancher проверить, что поды сервисов находятся в статусе running/ready.

Поды:

- delivery
- kafka
- postgres
- uploader
- prober
- transcoder
- task-manager
- zookeeper

Проверить наличие топиков в kafka:

- fhdelivery
- prober.tasks
- status
- taskmanager
- taskmanager.settings
- transcoder.tasks.cpu
- transcoder.tasks.gpu

Проверка наличия необходимых сервисов

```
kubectl get pods --namespace namespace
```

Минимально необходимый набор сервисов медиаплатформы

NAME	READY	STATUS	RESTARTS	AGE
admin-7f544fb8b7-4dvsz	1/1	Running	0	6m47s

```

delivery-6b757b497-8ts8b      1/1  Running  0      6m47s // Необязательный прикладной
компонент, позволяющий получить преобразованный медиаконтент с хранилища S3 не
используя API
fhdelivery-586b87f4d4-9pchn    1/1  Running  0      6m46s
prober-0-54b4759cbb-z46mj     1/1  Running  0      6m46s
task-manager-55675b6754-xjxbg  1/1  Running  0      6m46s
taskinfo-75c9c7b5b5-sbmcw     1/1  Running  0      6m46s
transcoder-cpu-0-8699cbfd5c-jx9ms 1/1  Running  0      6m47s
transcoder-gpu-0-8699cbfd5c-w9lnm 1/1  Running  0      6m46s
uploader-0-0                   1/1  Running  0      6m46s
webadmin-bfb5f685f-ph2c6      1/1  Running  0      6m47s

```

Необходимый набор инфраструктурных сервисов если они устанавливаются вместе с медиаплатформой в kubernetes

```

NAME                                READY STATUS  RESTARTS  AGE
infra-kafka-ui-84445b5f4c-qkcnw    1/1  Running  0         17m
kafka-0                             1/1  Running  1         17m
kafka-init-942dj                   0/1  Completed 0         6m46s // init job для создания
необходимой структуры данных в kafka
postgres-0                          1/1  Running  0         17m
postgres-init-pr9hs                 0/1  Completed 0         6m46s // init job для создания
необходимой структуры данных в postgres
zookeeper-0                         1/1  Running  0         17m

```

Проверка логов taskmanager

```

{"level":"info","time":"2023-10-17 14:42:20","message":"Starting TaskManager"}
{"level":"info","time":"2023-10-17 14:42:20","message":"Awaiting runtime settings"}
{"level":"info","time":"2023-10-17 14:42:20","message":"Runtime settings received"}
{"level":"info","time":"2023-10-17 14:42:20","message":"TranscodeManager Init"}
{"level":"info","time":"2023-10-17 14:42:20","message":"ProbeManager Init"}

```

7.1.2. Проверка работоспособности Медиаплатформы в процессе эксплуатации

Чек-лист работоспособности медиаплатформы		
#	Проверка	Ожидаемый результат

Загрузка		
1	Загрузить файл для обработки.	Файл успешно загружен. Получен id задачи. Получен ответ от Uploader со статусом finished.
2	Создана задача на загрузку файла для полученного id, сообщение в kafka в топике taskmanager - UPLOADER_TASK_CREATE	Сообщение создано.
3	Сообщение о начале загрузки файла для полученного id, сообщение в kafka в топике taskmanager - TRANSFER_TASK_START, статус ОК	Сообщение создано со статусом ОК.
4	Сообщение об окончании загрузки файла для полученного id, сообщение в kafka в топике taskmanager - TRANSFER_TASK_FINISHED, статус ОК	Сообщение создано со статусом ОК.
5	Сообщение в топике status для полученного id - TASK_STATUS, статус UploaderStarted	Сообщение создано со статусом UploaderStarted.
6	Сообщение в топике status для полученного id - TASK_STATUS, статус UploaderFinished	Сообщение создано со статусом UploaderFinished.
Обработка файла сервисом Prober		
7	Создана задача на обработку сервисом Prober в топике prober.tasks для полученного id - PROBE_TASK	Сообщение создано.
8	Обработка сервисом Prober успешно завершена, сообщение в kafka в топике taskmanager для полученного id - PROBE_TASK_FINISHED, статус ОК	Сообщение создано со статусом ОК.
9	Сообщение в топике status для полученного id - TASK_STATUS, статус ProbeFinished	Сообщение создано со статусом ProbeFinished.
Обработка файла сервисом Transcoder		
10	Создана задача на обработку сервисом Transcoder в топике transcoder.tasks.gpu или transcoder.tasks.cpu для полученного id - TRANSCODER_TASK	Сообщение создано.
11	Сообщение об успешном создании thumbnail файла, сообщение в kafka в топике taskmanager для полученного id - THUMBNAIL_INFO, статус ОК	Сообщение создано со статусом ОК.
12	Транскодирование успешно завершено, сообщение в kafka в топике taskmanager для полученного id - TRANSCODER_TASK_FINISHED, статус ОК	Сообщение создано со статусом ОК.
13	Сообщение в топике status для полученного id - TASK_STATUS, статус ThumbnailReady	Сообщение создано со статусом ThumbnailReady.
14	Сообщение в топике status для полученного id - TASK_STATUS, статус TranscoderFinished	Сообщение создано со статусом TranscoderFinished.
Проверка обработанных файлов		

15	Скачать результирующие видеофайлы через delivery endpoint.	Файлы успешно скачаны.
16	Скачать файл thumbnail.jpg через delivery endpoint.	Файлы успешно скачаны.
17	Скачать файл T0.jpeg через delivery endpoint.	Файлы успешно скачаны.
18	Проверить, что файлы .m4s проигрываются в плеере (VLC)	Файлы успешно проигрываются в плеере.
19	Проверить, что в файле thumbnail.jpg отображен кадр из видео	thumbnail.jpg является кадром из видео
20	Проверить, что в файле T0.jpeg отображено 100(10 рядов, 10 столбцов) миниатюр из видео.	T0.jpg содержит 100 миниатюр из видео

7.2. Действия при сбоях в функционировании Медиаплатформы

Медиаплатформа является сложным программным комплексом, в работе которого возможны сбои. В этом случае администратор Медиаплатформы может найти и устранить причину сбоя либо зарегистрировать инцидент. Для уменьшения количества сбоев рекомендуется выполнять действия, приведенные в разделе «Профилактические действия для предупреждения сбоев в работе Медиаплатформы».

7.2.1. Профилактические действия для предупреждения сбоев в работе Медиаплатформы

Профилактическое обслуживание Медиаплатформы заключается в следующем:

- установка обновлений безопасности 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 очистке ненужного контента;
- регулярная проверка работоспособности Медиаплатформы, включая загрузку медиаконтента в систему, обработку медиаконтента и получение обработанного контента.

7.2.2. Основные положения для исследования проблем, связанных с работой Медиаплатформы

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

1. Проведите общий осмотр сервисов Медиаплатформы средствами Kubernetes, Rancher, L
2. Проведите осмотр работоспособности стенда средствами мониторинга, изучив сопутствующие графики в Grafana.
3. Изучите лог Taskmanager.
4. Проверьте лог Kafka в случае обнаружения проблем со связью с K
5. Изучение messages в kafka - topics/taskmanager
6. При наличии проблем с обработкой медиаконтента, изучите отчёты T
7. Для выявления проблем с сетевой связью, настройкой балансера и проблем с хранилищем изучите отчёты U
8. При невозможности скачать имеющийся медиаконтент с хранилища через Delivery сервис изучите отчеты D

При обнаружении проблем в сторонних системах следуйте 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», следуйте приведенным указаниям;
- если случай не типичный, зарегистрируйте инцидент. По возможности, предоставьте описание проблемы и логи, собранные во время выполнения шагов, приведенных в этом разделе.

7.2.3. Известные проблемы в работе Медиаплатформы и методы их решения

	Проблема	Симптомы	Методы решения	Заметки
1.	Отказ NVIDIA драйверов и CUDA на узлах	<ul style="list-style-type: none"> · Экземпляры GPU-Transcoder перестаёт обрабатывать видео. · #nvidia-smi на GPU-узлах выдаёт 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 NVIDIA драйверов и CUDA на подах	<ul style="list-style-type: none"> · Экземпляр GPU-Transcoder перестаёт обрабатывать видео. · #nvidia-smi на экземплярах GPU-Transcoder выдаёт 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 Uploader видео загружается. · При пробросе порта для Delivery контент можно выгрузить. 	<ol style="list-style-type: none"> 1. Проверить валидность сертификатов балансера. 2. Проверить логи и конфигурацию балансера. 	
4.	Хранилище переполнено или недоступно	<ul style="list-style-type: none"> · Не загружается или не выгружается видео. · При пробросе порта для Uploader видео не загружается · Хранилище недоступно. · Ошибки в логах Uploader и Taskmanager при работе с хранилищем. 	<ol style="list-style-type: none"> 1. Проверить доступность хранилища. 2. Проверить наличие нужных директорий или бакетов (конфигурация в Uploader поможет понять необходимое). 3. Проверить наличие ключей доступа и их соответствие ключам в конфигурации Видеоплатформы (конфигурация Uploader). 	

	Проблема	Симптомы	Методы решения	Заметки
5.	Не работает Kafka	<ul style="list-style-type: none"> · Видеоплатформа не обрабатывает медиаконтент. · Сервисы Видеоплатформы в логах говорят о потерянной связи с Kafka. 	<ol style="list-style-type: none"> 1. Перезагрузка Kafka. 2. Перезагрузка сервисов Видеоплатформы. 	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. Пример ошибки приведен в разделе «1.3.1 Пример для кодека av1_nvenc – GPU-Transcoder log (ошибка 6)». пример для кодека av1_nvenc - gpu-transcoder log</p> <pre> 2023-07-26 08:23:38.646 [I] [VideoFilter.cpp @ open:61] Using GPU filter for source video stream #0 2023-07-26 08:23:38.647 [D] [AVUtils.cpp @ libavLogString:111] [V] [av1_nvenc @ 0x7f61fbe26d40] Loaded Nvenc version 12.0 2023-07-26 08:23:38.647 [D] [AVUtils.cpp @ libavLogString:111] [V] [av1_nvenc @ 0x7f61fbe26d40] Nvenc initialized successfully 2023-07-26 08:23:38.655 [W] [AVUtils.cpp @ libavLogString:111] [W] [av1_nvenc @ 0x7f61fbe26d40] Codec not supported 2023-07-26 08:23:38.655 [C] [AVUtils.cpp @ libavLogString:111] [F] [av1_nvenc @ 0x7f61fbe26d40] Provided device doesn't support required NVENC features </pre>	<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 или 5. повторное разворачивание стенда. 	Проводятся испытания и обновление версий драйверов для поддержки новых кодеков, зависящих от драйверов NVIDIA.

	Проблема	Симптомы	Методы решения	Заметки
7.	Неподдерживаемые кодеки согласно спецификации Видеоплатформы	Видеоплатформа не обрабатывает медиа-контент; в отчётах Taskmanager упоминается Unsupport codec.	Проверить что кодеки и формат загружаемого медиаконтента удовлетворяет требованиям к загружаемым медиафайлам.	
8.	Произведено масштабирование сервисов Transcoder или Prober средствами Kubernetes	Спорадические ошибки при обработке медиа-контента.	Проверить соответствие partitions для Prober, CPU-Transcoder, GPU-Transcoder в Kafka. При необходимости привести в соответствие количество сервисов Видеоплатформы или заново развернуть стенд с корректной конфигурацией.	
9.	Экземпляры GPU-Transcoder попали на узел без поддержки GPU	· 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 развернув стенд.	Вероятны проблемы с ребалансировкой подов и некорректной настройкой taints или другой системой распределения подов по специальным узлам

7.2.4. Действия при неустановленной проблеме

7.2.4.1. Остановка и старт Медиаплатформы

При неустановленной проблеме имеется возможность выполнить перезапуск Медиаплатформы

7.2.4.1.1. Остановка Медиаплатформы

1. Запомнить текущее количество реплик каждого сервиса – необходимо для последующего возвращения нужного количества реплик.
2. Средствами kubernetes уменьшить количество реплик сервисов Медиаплатформы до нуля, инфраструктурные сервисы останавливать нельзя.

7.2.4.1.2. Повторный запуск Медиаплатформы

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

7.2.4.2. Откат установки

Действия при неуспешном перезапуске:

1. Удалить уже установленную версию Медиаплатформы, согласно разделов «Инструкции по развертыванию Единой Медиаплатформы».
2. Вновь установить предыдущую версию Медиаплатформы, согласно разделов Инструкции по развертыванию Единой Медиаплатформы.

Для этого необходимо выбрать версию Медиаплатформы патч-версий 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.

Медиаплатформа состоит из набора stateless микросервисов.

Все обрабатываемые данные сохраняются во внешних сервисах. Отдельных процедур по возврату данных в прежнее состояние не требуется.