

Catena Streaming Engine

Manual

Кластерная система обработки видео

Максим Лапшин

© Эрливидео 2025

Table of contents

1. Продукты	3
2. Catena Streaming Engine	4
2.1 Установка	6
2.2 Каналы	10
2.3 Обслуживание	56

1. Продукты

2. Catena Streaming Engine

2.0.1 Catena SE

Catena Streaming Engine - это программа, позволяющая принять, транскодировать, записать и доставить телевизионные потоки на кластере от 1 до 200 серверов.

Catena SE может работать как с middleware Catena, так и с другими IPTV middleware.

Основные возможности Catena SE

- Захват видео
- Онлайн мониторинг качества принимаемого видео
- Отказоустойчивость приема: резервные источники и политики переключения
- Транскодирование и подготовка видео для трансляции на телевизорах и ТВ-приставках
- Запись видео
- Автоматическое распределение загрузки по всему кластеру при наличии более одного сервера
- Интеграция с middleware (в том числе Catena): контроль доступа, выдача ссылок/плейлистов, DRM
- Наблюдаемость и эксплуатация: метрики, алерты, журналы, аудит действий операторов
- Автоматизация: шаблоны, API и массовые операции для управления каналами и серверами

Быстрый старт

- [Быстро запустить комплекс для теста](#)
- [Настроить телеканал](#)
- [Проиграть ТВ канал](#)

Навигатор по документации

Эта документация поможет вам:

- Быстро запустить комплекс для теста
- Настроить телеканал
 - Принять телеканал из интернета
 - Захватить локальный мультиканал
 - Создать собственный ТВ канал из загруженных файлов
 - Добавить матрац
 - Зарезервировать источники
 - Транскодировать в MBR
 - Управлять темплейтами
 - Включить распознавание DVB субтитров в HLS WebVTT
 - Подготовить скриншоты для быстрой перемотки
 - Записать архив для nPVR услуги
 - Сделать сдвиг времени (time-zone compensation)
 - Отправить ТВ канал в мультиканал
 - Проиграть ТВ канал
 - Показать скриншоты для перемотки
 - Настроить авторизацию проигрывания
 - Настроить авторизацию в разных middleware
- Добавить в Catena
 - Интегрировать в другую middleware
 - Массовое управлять телеканалами из middleware
- Проверить статус системы
 - Прочитать логи
- Обслуживать
 - Сделать бекап и восстановить из бекапа
 - Добавить сервер
 - Вывести сервер
 - Собрать кластер

2.1 Установка

Установка Catena Streaming Engine состоит из следующих этапов:

1. Подготовка оркестратора Kubernetes на примере установки системы k3s
2. Подготовка окружения внутри запущенного kubernetes
3. Установка Catena SE

Catena SE разрабатывается с расчетом на запуск под kubernetes. Этот архитектурный выбор позволяет строить системы, переживающие выход из строя любого сервера из вашего кластера.

2.1.1 Требования для установки

- Один и более серверов для установки всего стримингового ПО под управлением Ubuntu 24.04

2.1.2 Установка kubernetes на головном сервере

Команду надо выполнить на головном сервере.

```
curl -sL https://flussonic.ru/doc/catena-se/install.sh | sh -s - kube --control-ip 192.168.0.2
```

--control-ip – внутренний IP-адрес сервера на случай, если на нём больше одного интерфейса. Это тот адрес, по которому будут подключаться другие ноды. Опция не обязательная.

--public-ip – публичный IP адрес сервера для подключения к кубернетесу снаружи. Можно не указывать, если будете обращаться только через VPN. Опция не обязательная.

Вы увидите следующий вывод:

```
# curl -sL https://flussonic.ru/doc/catena-se/install.sh | sh -s - kube --public-ip 1.2.3.4
Installing k3s server on this host (visible as 192.168.0.2)
Token will be stored in /var/lib/rancher/k3s/server/node-token (copy to agent hosts for join)
[INFO]  Finding release for channel stable
[INFO]  Using v1.34.4+k3s1 as release
...
Writing k3s.yaml to /root/k3s.yaml ...
Waiting for k3s API to be ready ...
NAME      STATUS   ROLES      AGE      VERSION
streamer3  NotReady  <none>    0s       v1.34.4+k3s1
  % Total   % Received % Xferd  Average Speed   Time   Time  Current
     0      0      0      0      0      0      0      0 --:--:-- --:--:-- --:--:-- 57628
100 11929  100 11929  0      0 57359  0 --:--:-- --:--:-- --:--:-- 57628
[WARNING] Could not find git. It is required for plugin installation.
Helm v4.1.1 is already latest
Multus manifest written to /var/lib/rancher/k3s/server/manifests/multus.yaml

Next: export KUBECONFIG=/root/k3s.yaml (or run from this dir), then sh gateway
```

Теперь необходимо установить веб-сервер и прочие сопутствующие пакеты:

```
curl -sL https://flussonic.ru/doc/catena-se/install.sh | sh -s - gateway
```

```
root@streamer3:~# curl -sL https://flussonic.ru/doc/catena-se/install.sh | sh -s - gateway
Installing cert-manager (HelmChart manifest) ...
Release "cert-manager" does not exist. Installing it now.
Pulled: quay.io/jetstack/charts/cert-manager:1.19.3
Digest: sha256:c6edad39343c88a8f0ca4df93ee8ce1d049bfffaffead3f325bbcb6df3358432f
NAME: cert-manager
LAST DEPLOYED: Sun Feb 15 13:50:01 2026
NAMESPACE: cert-manager
STATUS: deployed
...
https://cert-manager.io/docs/usage/ingress/
Waiting for cert-manager to be ready ...
...
Installing Envoy Gateway (HelmChart manifest) ...
Pulled: docker.io/envoyproxy/gateway-helm:1.7.0
Digest: sha256:fdcc01017304bc676fb7097d1fe8ceda5cc1c9673bf1c01a7576898623b3310f7
NAME: eg
LAST DEPLOYED: Sun Feb 15 13:52:25 2026
NAMESPACE: envoy-gateway-system
```

```
STATUS: deployed
...
To get more details, please visit https://gateway.envoyproxy.io and https://github.com/envoyproxy/gateway.
Waiting for Envoy Gateway to be ready ...
```

Если увидите такие сообщения, то это не страшно:

```
Waiting for cert-manager to be ready ...
error: timed out waiting for the condition on pods/cert-manager-6c46b7bb4-jlvrj
```

Просто установка может занять больше времени, чем ожидалось.

Если что-то пойдет не так, то перед повторной установкой надо удалить старую инсталляцию:

```
sudo k3s-uninstall.sh
```

2.1.3 Настройка инсталляции

В этом разделе будет описано, как запустить Catena SE на одном сервере без балансировщика, без TLS.

Для этого этапа нужно подготовить лицензионный ключ и выяснить имя ноды.

Имя ноды выясняется следующим образом на сервере:

```
# kubectl get nodes
NAME      STATUS    ROLES     AGE      VERSION
streamer3  Ready     control-plane   14m     v1.34.4+k3s1
```

Далее надо создать на целевом сервере файл `catena-se.yaml`

```
license: "my-license-key"

streamers:
  hosts:
    - node: streamer3
```

Что надо настраивать в этом файле?

- `my-license-key` заменяется на ваш лицензионный ключ, который надо взять в личном кабинете
- `streamer3` для каждого стримера в кластере надо прописать имя этой ноды, которое доступно в `kubectl get nodes`

Теперь можно установить само приложение:

```
curl -sL https://flussonic.ru/doc/catena-se/install.sh | sh -s - app --values catena-se.yaml
```

Будет следующий вывод:

```
root@streamer3:~# curl -sL https://flussonic.ru/doc/catena-se/install.sh | sh -s - app --values catena-se.yaml
Applying central-operator ...
namespace/central-operator-system created
customresourcedefinition.apirextensions.k8s.io/centrals.media.flussonic.com created
serviceaccount/central-operator-controller-manager created
role.rbac.authorization.k8s.io/central-operator-leader-election-role created
clusterrole.rbac.authorization.k8s.io/central-operator-manager-role created
clusterrole.rbac.authorization.k8s.io/central-operator-metrics-reader created
...
service/media-server-operator-controller-manager-metrics-service created
deployment.apps/media-server-operator-controller-manager created
pod/media-server-operator-controller-manager-7d566d4b7c-8w1nd condition met
Installing Catena SE helm chart (-values catena-se.yaml) ...
Release "tv" does not exist. Installing it now.
Pulled: docker.io/flussonic/catena-se:26.2.3
Digest: sha256:62d58a19ccce18171e702921821d91fa476b5916701748fb73d9db618d416c5
NAME: tv
LAST DEPLOYED: Sun Feb 15 14:05:26 2026
NAMESPACE: catena-se
STATUS: deployed
REVISION: 1
DESCRIPTION: Install complete
NOTES:
Your login is `root`

Your current Catena SE password can be extracted with following command:
kubectl -n catena-se get secret license -o jsonpath='{.data.edit_auth}' | base64 --decode | awk -F : '{print $2}'
```

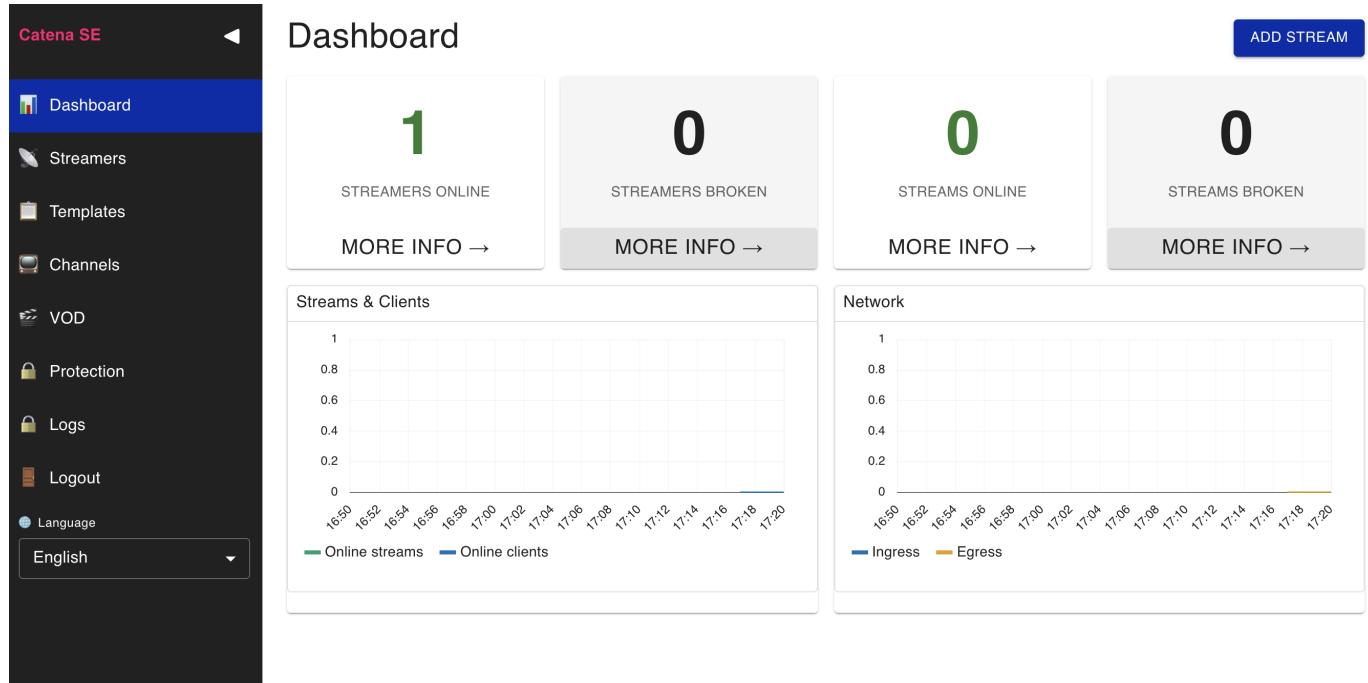
Теперь выполните последнюю команду и получите пароль.

```
# kubectl -n catena-se get secret license -o jsonpath='{.data.edit_auth}' | base64 --decode | awk -F : '{print $2}'
6CytVA00aFYpvQ
```

2.1.4 Проверка инсталляции

Надо зайти на адрес <http://your-installation>

Введите логин и пароль с предыдущего шага



2.1.5 Обновление настроек приложения

Если вы поменяли файл с настройками инсталляции, можно обновить инсталляцию

```
curl -sL https://flussonic.ru/doc/catena-se/install.sh | sh -s - app --values ./catena-se.yaml
```

Т.е. повторный запуск скрипта с режимом прогона app позволит обновить приложение и заново применить настройки.

2.1.6 Включение HTTPS

Надо отредактировать `catena-se.yaml` и добавить секцию `ingress`:

```
license: ...
ingress:
  host: my-server-hostname.com
streamers:
  hosts:
    - node: streamer3
      host: my-server-hostname.com
...
```

Т.е. один и тот же хостнейм надо указать и в настройках `ingress` и у первого стримера.

После этого обновите конфигурацию по инструкции выше.

2.1.7 Добавление серверов

Добавить второй сервер можно по инструкции [по добавлению сервера](#)

2.2 Каналы

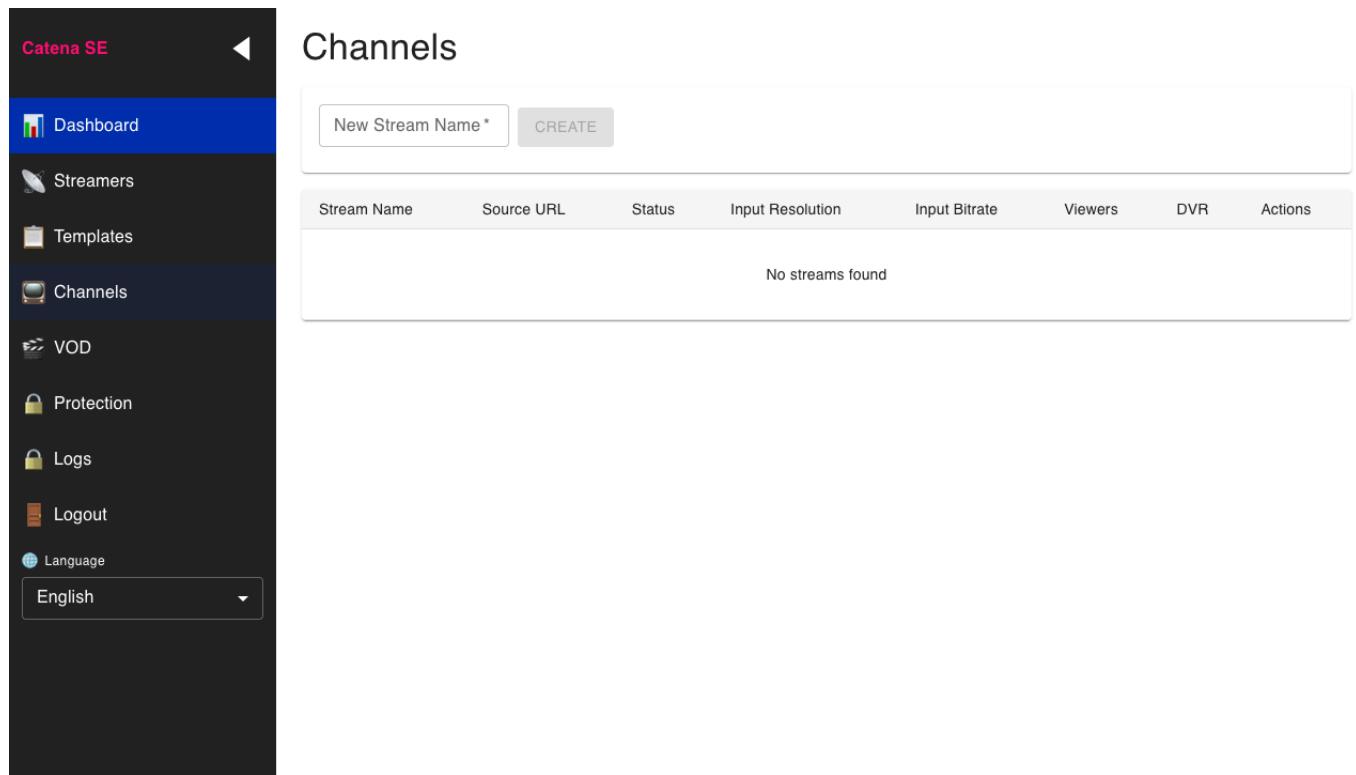
2.2.1 Настройка ТВ канала с демо-источником

В этом мануале будет описано, как настроить в Catena SE демонстрационный канал, для которого вам не нужны никакие внешние источники.

Последовательность шагов следующая:

1. Создаём новый ТВ канал

Перейдите в раздел **Channels**. Введите имя канала (например, `demo`) и нажмите кнопку создания канала.



Stream Name	Source URL	Status	Input Resolution	Input Bitrate	Viewers	DVR	Actions
No streams found							

После создания откроется страница нового канала.

Catena SE

Channels <- demo

BASIC SETTINGS SOURCES PROCESS EGRESS DVR PROTECTION

Name: demo Title: Template:

Disabled Static Status: streamer3

Player Not found

SAVE **CANCEL** **DELETE**

2. Добавляем источник

Откройте вкладку **Inputs** (Источники). Изначально список источников пуст.

Нажмите **Add source**, укажите URL источника `fake://` и сохраните настройки (**Save**).

3. Проверяем проигрывание

Перейдите на вкладку **Basic** и убедитесь, что встроенный плеер показывает поток.

Channels <- demo

BASIC SETTINGS SOURCES PROCESS EGRESS DVR PROTECTION

Name: demo Title: Select a template to apply defaults

Status: streamer3

Player



demo
2026-02-09T12:17:40Z
2026-02-09T12:17:40Z
1770639460.551301
streamer@server.1

SAVE CANCEL DELETE

2.2.2 Настройка ТВ канала с источником из интернета

В этом документе будет описано, как настроить в Catena SE ТВ канал с источником, доступным из интернета.

Как правило подобные источники характеризуются своей непредсказуемостью, высокой степенью изменчивости, пониженной надежностью.

Доступны следующие протоколы источников:

Протокол	Пример урла	Особенности
HTTP MPEG-TS	tshttp://source-ip/some-path.ts	Важно указать протокол <code>tshttp://</code> а не просто <code>http://</code>
HLS	hls://source-ip/some-path.m3u8	Аналогично: надо указывать <code>hls://</code> вместо <code>http://</code>
RTMP	rtmp://source-ip/app/some-path	
SRT	srt://source-ip:9051?passphrase=123	Предпочтительнее, если вам предоставлят источник с раздельными портами для каждого канала
M4F	m4f://source-ip/channel	Получение видео от Flussonic streaming server

Последовательность шагов следующая:

1. Создаем новый ТВ канал
2. Добавляем в него источник с URL в нужном протоколе (например, `hls://...` или `tshttp://...`)
3. Проверяем как проигрывается ТВ канал встроенным плеером

1. Создаём новый ТВ канал

Перейдите в раздел **Channels**. Введите имя канала (например, `internet`) и нажмите кнопку создания канала.

После создания откроется страница нового канала.

2. Добавляем источник

Откройте вкладку **Inputs** (Источники). Изначально список источников пуст.

Нажмите **Add source**, укажите URL источника в нужном протоколе (например, для HLS:

`hls://devstreaming-cdn.apple.com/videos/streaming/examples/bipbop_16x9/bipbop_16x9_variant.m3u8`) и сохраните настройки (**Save**).

3. Проверяем проигрывание

Перейдите на вкладку **Basic** и убедитесь, что встроенный плеер показывает поток. Для интернет-источника запуск может занять несколько секунд.

Catena SE

Channels <- internet

BASIC SETTINGS SOURCES PROCESS EGRESS DVR PROTECTION

Name: **internet** Title: **Template** Select a template to apply defaults

Disabled Static Status: **streamer4**

Player

ENLACE TELEVISIÓN

ANÁLISIS DE FENALCO BARRANCABERMEJA SOBRE ECONOMÍA LOCAL

7:24 AM

Dashboard Streamers Templates Channels VOD Protection Logs Logout Language English

2.2.3 Создание ТВ канала с источником из мультикаста

В этом документе будет описано, как настроить в Catena SE ТВ канал с источником, доступным в локальной сети в виде мультикаста.

На практике чаще всего встречается **SPTS**: один UDP multicast поток содержит один телеканал.

Иногда встречается **MPTS**: один UDP multicast поток содержит много телеканалов (программ). В этом случае нужно выбрать нужную программу внутри MPTS.

Что понадобится

- Мультиicast-группа и порт источника: 239.0.0.1:1234 (пример).
- Для MPTS: идентификатор программы (часто называют **program/service id**, **PNR** или **service_id**).
- Доступ Catena SE до источника в сети (VLAN/маршрутизация/IGMP – в зависимости от вашей сети).

SPTS: один канал в одном мультикасте

Последовательность шагов следующая:

1. Создаем новый ТВ канал
2. Добавляем в него источник мультикаста (см. ниже)
3. Проверяем как проигрывается ТВ канал встроенным плеером

1. СОЗДАЁМ НОВЫЙ ТВ КАНАЛ

Перейдите в раздел **Каналы**, введите имя канала (например, `multicast`) и создайте канал.

После создания откроется страница нового канала.

Channels <- multicast

BASIC SETTINGS SOURCES PROCESS EGRESS DVR PROTECTION

Name: multicast

Title

Template: Select a template to apply defaults

Status: streamer3

Player: Not found

Language: English

2. ДОБАВЛЯЕМ ИСТОЧНИК МУЛЬТИКАСТА

Откройте вкладку **Источники**. Изначально список источников пуст.

Нажмите **Добавить источник**, укажите URL мультикаста (для SPTS – адрес и порт) и сохраните.

КАК УКАЗАТЬ SPTS ИСТОЧНИК

Если ваш источник – SPTS, достаточно указать только адрес и порт:

udp://239.0.0.1:1234

Пример: udp://239.0.10.1:1234 – группа 239.0.10.1, порт 1234.

3. ПРОВЕРЯЕМ ПРОИГРЫВАНИЕ

Перейдите на вкладку **Основное** и убедитесь, что встроенный плеер показывает поток. Для мультикаста запуск может занять несколько секунд.

The screenshot shows the 'Channels <- multicast' configuration screen in the Catena SE software. The channel is named 'multicast' and is set to 'Static' status, connected to 'streamer3'. A preview window shows a woman with a headband, and the ORICA TELEVISION logo is visible in the top right corner of the video frame.

MPTS: несколько каналов в одном мультикасте

Если в мультикасте приходит **MPTS**, Catena SE должна “вырезать” из MPTS нужную программу и дальше обрабатывать её как обычный канал.

Как указать источник MPTS

Если в мультикасте приходит **MPTS**, обязательно укажите, какую программу из него нужно выбрать.

Для этого используется параметр `programs=<id>` в URL источника.

Пример:

```
udp://239.0.0.1:1234?programs=101
```

Где 101 – это идентификатор программы (PNR/service_id) внутри MPTS.

Если не указать `programs=...`, система может попытаться обработать лишние программы из MPTS, что приведёт к лишней нагрузке и непредсказуемому результату.

Как узнать `programs=<id>`

`programs=<id>` должен совпадать с тем, как провайдер/головная станция маркирует канал внутри MPTS.

Обычно этот номер можно:

- получить у вашего DVB/IPTV оборудования или у провайдера сигнала;
- посмотреть в анализаторе MPEG-TS (по таблицам PAT/PMT), где будет указан program number / service id.

Типовые проблемы мультикаста

При захвате мультикаста самые частые причины "сыпется/не стартует" – не Catena SE, а сеть:

- IGMP/IGMP snooping настроены неправильно, и трафик не доходит до ноды.
- Мультиктаст приходит не на ту ноду/интерфейс (особенно в multi-homing сетях).
- Потери UDP пакетов из-за перегруза линка или очередей.

Если источник нестабилен, используйте сценарии отказоустойчивости из [Резервирование источников](#).

2.2.4 Телеканал из плейаута (файлы и стримы)

В этом документе описано, как создать в Catena SE плейаут – расписание воспроизведения из файлов (VOD) и при необходимости из стримов – и вывести его как телеканал.

Последовательность действий:

1. Создать плейаут и заполнить плейлист файлами (и при необходимости стримами).
2. Создать телеканал и указать ему источник вида `playlist:///vod/имя-плейаута.txt`.

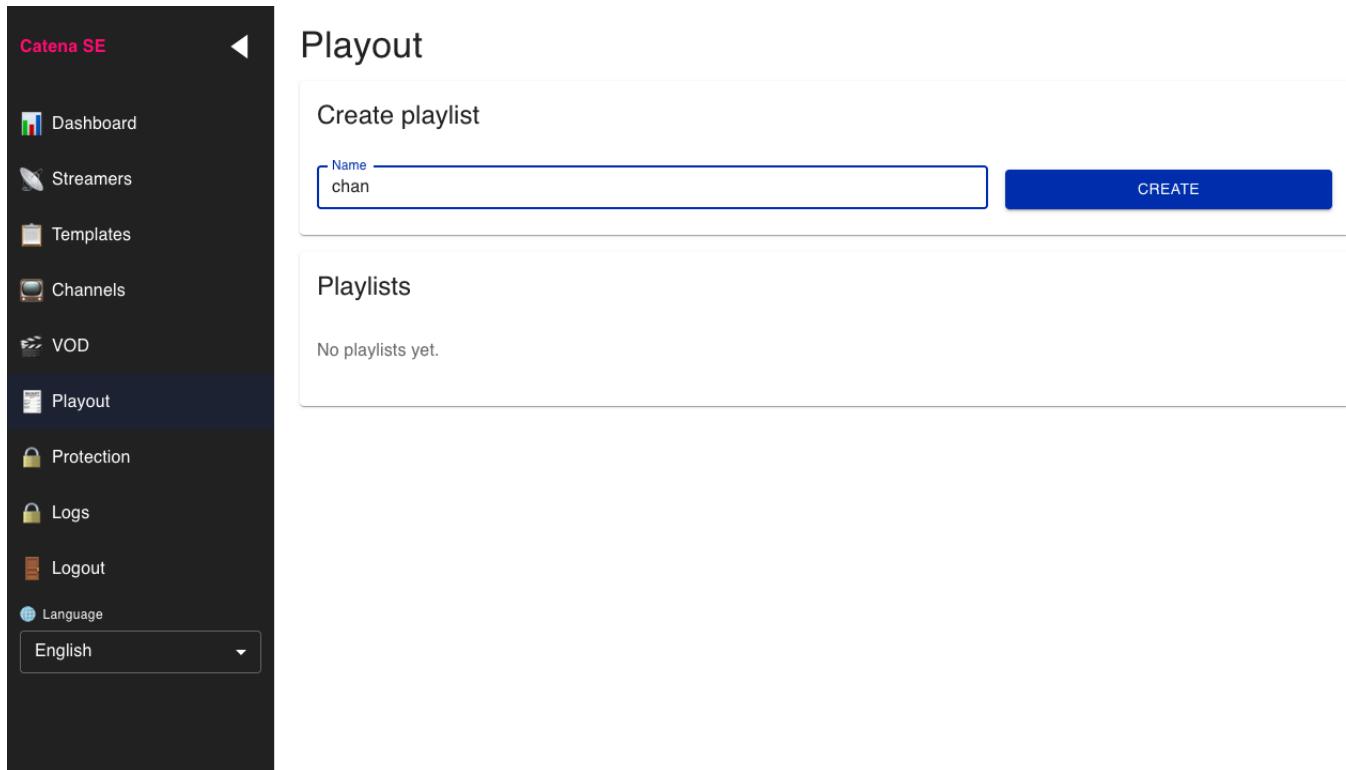
Что понадобится

- Файлы, загруженные в раздел **VOD** Catena SE (или доступные по URL).
- Имя для плейаута (например, `chan`). По нему будет создан плейлист `имя.txt` в VOD.

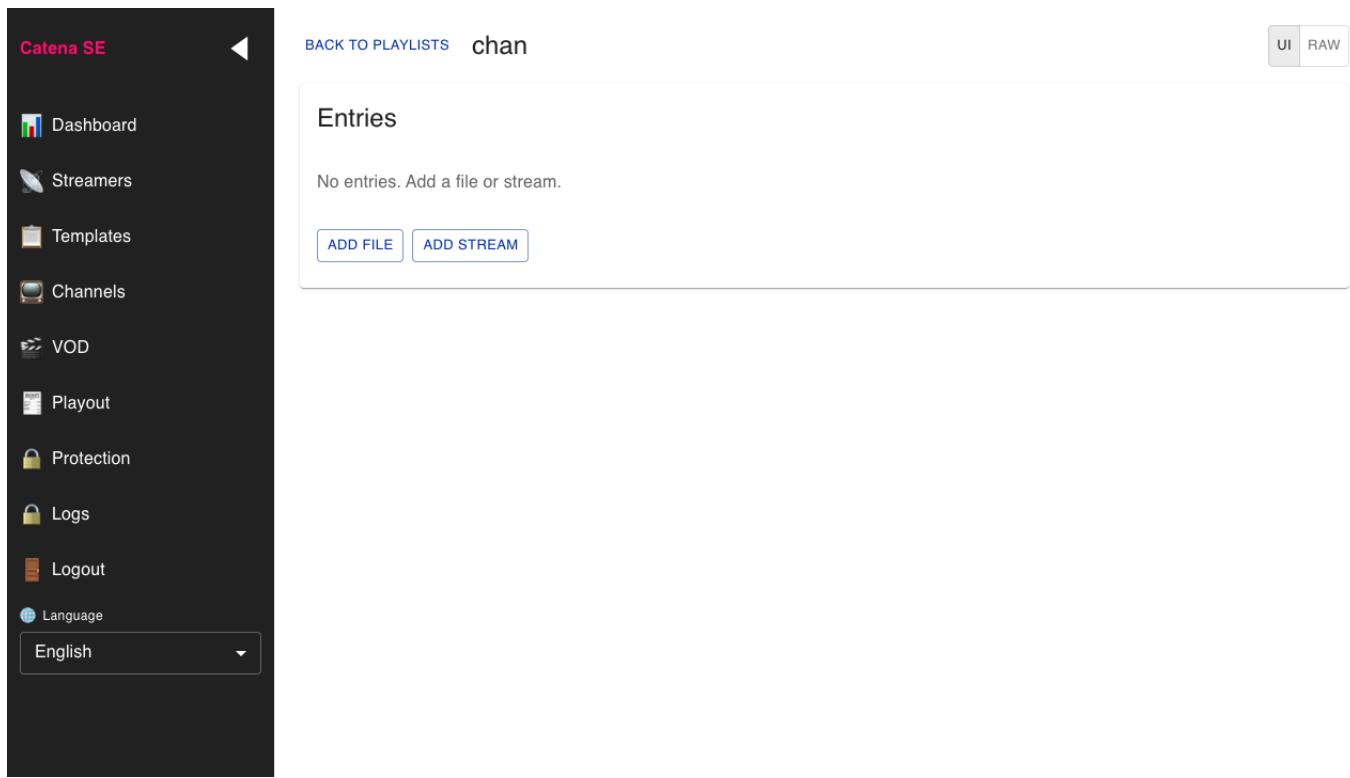
Шаг 1. Создание плейаута и плейлиста

1.1. СОЗДАЁМ НОВЫЙ ПЛЕЙАУТ

Перейдите в раздел **Playout** (Плейаут). В блоке **Create playlist** введите имя плейаута (например, `chan`) и нажмите **CREATE**.



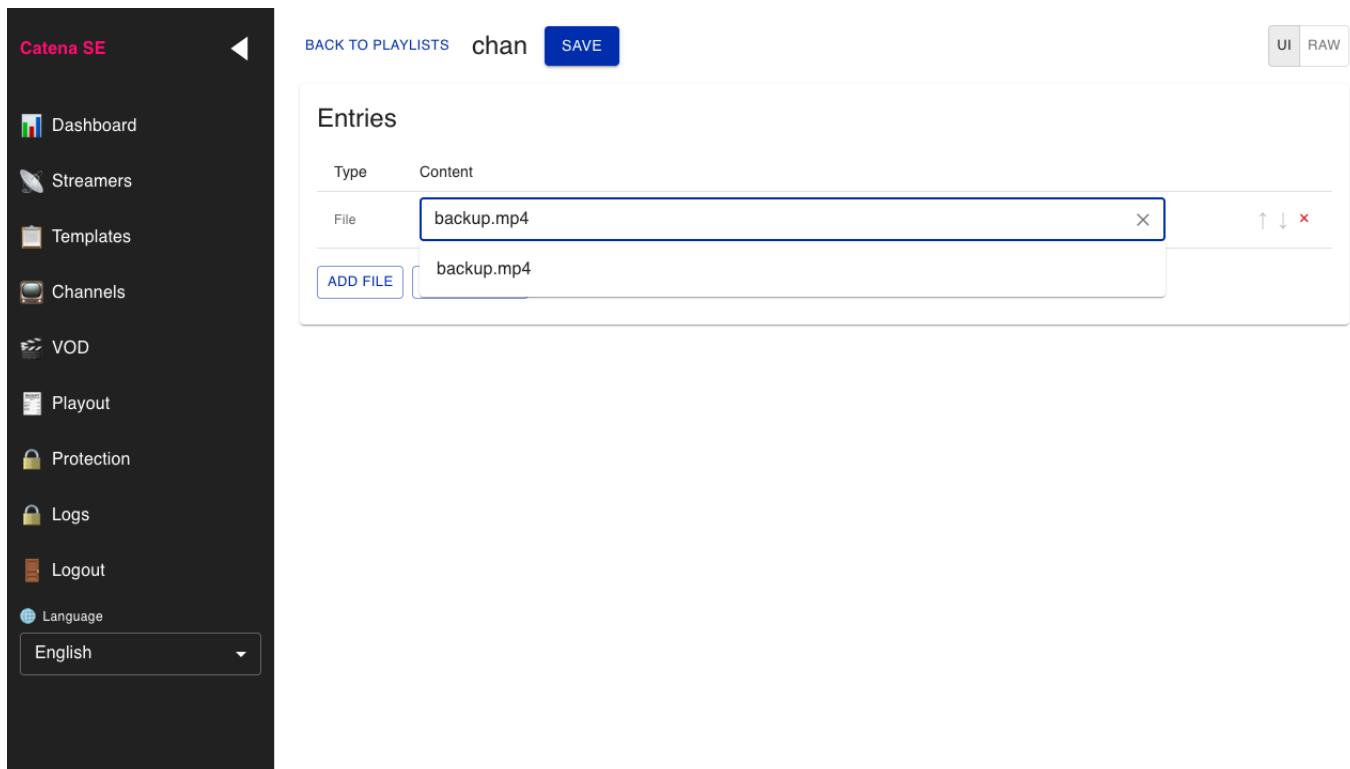
После создания откроется редактор плейаута с пустым плейлистом.



1.2. ДОБАВЛЯЕМ ФАЙЛЫ В ПЛЕЙЛИСТ

Нажмите **Добавить файл** (или **Add file**). В списке появится строка с полем имени файла. Укажите путь к файлу в VOD (например, `backup.mp4`). При необходимости добавьте ещё строки и укажите следующие файлы (например, `agro-drone.mp4`).

Порядок строк задаёт порядок воспроизведения в плейяуте.



1.3. СОХРАНЯЕМ ПЛЕЙЛИСТ

Нажмите **Save** (Сохранить). После сохранения плейлист начнёт использоваться при обращении к `playlist:///vod/chan.txt`.

Шаг 2. Создание канала с источником-плейаутом

Перейдите в раздел **Channels** (Каналы). Создайте новый канал (например, `playout-demo`). На странице канала откройте вкладку **Источники (Inputs)**, нажмите **Добавить источник (Add source)** и укажите URL плейаута:

```
playlist:///vod/chan.txt
```

Здесь `chan` — имя вашего плейаута; в VOD ему соответствует файл `chan.txt`. Сохраните канал.

Channels <- playout-demo

BASIC SETTINGS SOURCES PROCESS EGRESS DVR PROTECTION

Source timeout (s)

Source 1 **playlist**

URL *

Common Settings

Comment

Timeout (s)

Max Retry Timeout (s)

User Agent

Headers

+ ADD HEADER

+ ADD SOURCE

Backup

File

Timeout (s)

Record backup to DVR

После этого канал будет вещать контент по расписанию плейаута.

Стримы в плейауте: обязательно указывайте длительность

Если в плейаут добавляются не только файлы, но и **стримы** (живые потоки), для каждого стрима обязательно укажите **длительность** (duration). Без длительности системе неизвестно, когда переключаться с этого стрима обратно на следующий элемент плейлиста, и воспроизведение может «застрять» на стриме.

Для файлов (VOD) длительность обычно определяется автоматически; для стримов её нужно задать вручную при добавлении в плейлист.

2.2.5 Проигрывание файла при проблемах источника

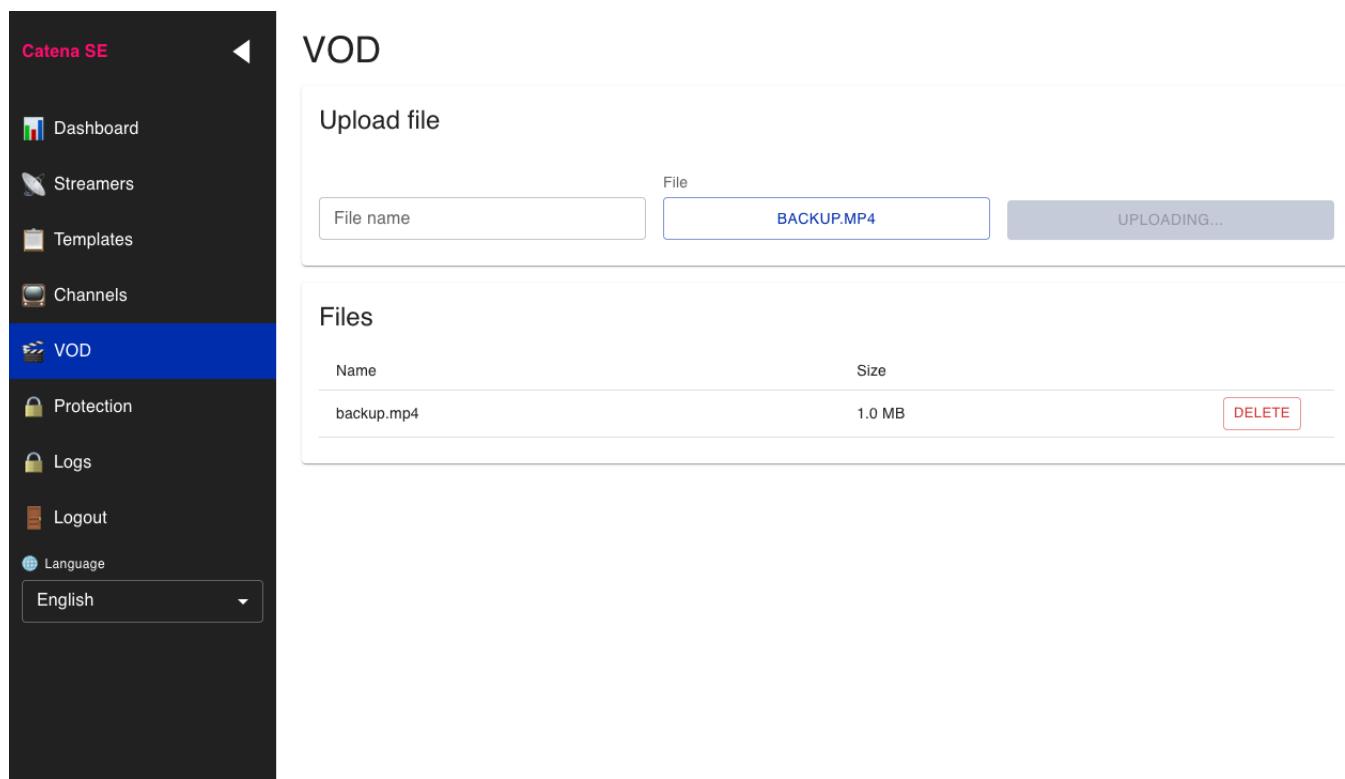
При потере связи с источником или некачественном сигнале можно автоматически переключаться на заставку (файл из VOD). Ниже – пошаговая настройка.

Последовательность шагов:

1. Загрузить файл заставки в раздел VOD
2. Создать канал и указать основной источник
3. Указать файл в качестве бекапа (заставки) на канале
4. Сохранить и проверить воспроизведение
5. (Опционально) симулировать отказ источника и убедиться в переключении на заставку

1. Загрузка файла в VOD

Перейдите в раздел **VOD**, загрузите файл с заставкой (например, видео в формате, подходящем под ваш шаблон). Дождитесь появления файла в списке.



Имя файла (например, `backup.mp4`) понадобится при настройке бекапа на канале.

2. Канал и основной источник

Перейдите в **Каналы**, создайте новый канал (например, `backup-demo`). Откройте вкладку **Источники**, добавьте основной источник (URL вашего потока – multicast, HLS и т.п.). При необходимости задайте таймаут переключения на бекап (секунды).

Канал с настроенным основным источником

3. Настройка бекапа (заставки)

В той же вкладке **Источники** укажите бекап: в поле заставки введите имя загруженного в VOD файла (например, `backup.mp4`). Задайте таймаут включения бекапа при необходимости. Нажмите **Сохранить**.

При недоступности основного источника Catena SE переключит воспроизведение на заставку.

Таймаут бекапа отличается от таймаута источников тем, что таймаут источника отсчитывается от старта источника, а таймаут бекапа отсчитывается от последнего полученного кадра, т.е. бекап может включаться, пока ещё перебираются источники.

4. Проверка воспроизведения

Перейдите на вкладку **Основное** и убедитесь, что поток идёт (основной источник или, если он недоступен, заставка).

5. Проверка переключения на заставку (оpционально)

Чтобы убедиться в работе бекапа:

- временно измените URL основного источника на заведомо невалидный и сохраните;
- в плеере должно произойти переключение на заставку;
- верните корректный URL источника и сохраните – воспроизведение должно переключиться обратно на основной поток.

2.2.6 Резервирование источников

Нормальная практика иметь более одного источника для телеканала на случай отказа или поломки.

Catena SE позволяет указать более одного источника для телеканала.

Просто добавьте второй input к каналу и в случае, если с первого не приходят кадры указанное время (`source_timeout` секунд), то будет переключение на второй и дальше.

2.2.7 Внешний источник конфигурации стримов (config_external)

Catena SE может получать список телеканалов (стримов) из внешнего сервиса провижининга вместо ручной настройки. Такой сервис вы реализуете сами: Catena SE обращается к нему по протоколу `config_external` и синхронизирует конфигурацию со своим состоянием.

Ключевой метод, который обязан предоставлять ваш сервис – `streams_list` (GET `/streams`). По этому методу Catena SE выгружает полный список стримов и принудительно приводит свою конфигурацию к этому списку (добавляет, обновляет и при необходимости удаляет каналы).

Что должен реализовать ваш сервис

Внешний сервис должен реализовать HTTP API, совместимый с описанием в приложенном файле `catena-se.yaml` (OpenAPI 3.1). В первую очередь нужен метод:

- `GET /streams` (operationId: `streams_list`) – возвращает список стримов в формате `streams_list`.

Формат ответа:

- Тело ответа – JSON-объект с полем `streams`: массив элементов типа `stream_config`.
- Опционально поддерживаются поля постраничной навигации из `collection_response`: `next`, `prev`, `estimated_count`, `timing`. Catena SE может передавать query-параметры `limit` и `cursor` для постраничной выборки.

Каждый элемент в `streams` – конфигурация одного стрима: `name`, `inputs`, `template`, `dvr`, `labels`, таймауты, `backup`, `on_play` и прочие поля, описанные в схеме `stream_config` в `catena-se.yaml`. Имя стрима (`name`) уникально и после создания не меняется.

Как Catena SE использует список стримов

Алгоритм синхронизации со стороны Catena SE:

- Catena SE запрашивает у внешнего сервиса список стримов (GET `/streams`, при необходимости с `cursor` / `limit`).
- Все стримы из ответа добавляются в Catena SE или обновляются: если канал уже есть, его настройки полностью перезаписываются данными из ответа. Существующие лейблы канала заменяются теми, что пришли от внешнего сервиса.
- Каналы, которые есть в Catena SE, но отсутствуют в списке от внешнего сервиса, удаляются – **кроме** каналов с лейблом `manual`.
- Каналы с лейблом `manual` не удаляются и не перезаписываются этим механизмом.

Чтобы какой-то канал не управлялся внешним провижинингом, ему нужно выдать лейбл `manual` (например, в `labels.manual`). Если во внешнем сервисе позже появится стрим с тем же именем, при следующей синхронизации лейбл будет перезаписан и канал перейдёт под автоматическое управление.

Синхронизация списка передач

Middleware может отдавать абонентам информацию о статусе записи каждой телепередачи для корректного проигрывания.

Статус записи телепередачи можно получить с Catena SE, запросив передачи за указанное время методом `episodes_get`. Catena SE при необходимости запрашивает список передач с сервера по протоколу `config_external` для недостающих диапазонов времени, дообогащает его и отдаёт.

Механизм рассчитан на то, что границы передач не меняются, т.е. кеширование выполняется один раз.

Пример: один файл и HTTP-сервер

Минимальный вариант – один статический файл `streams` с JSON в формате `central_streams_list`. В этой папке лежит пример такого файла с одним стримом (демо-источник `fake://fake`).

Запуск отдачи файла через встроенный HTTP-сервер Python:

```
python3 -m http.server 8000
```

После запуска список стримов будет доступен по адресу `http://localhost:8000/streams`. В настройках Catena SE укажите URL вашего сервера (например, `http://HOST:8000`), чтобы метод `config_external` запрашивал `http://HOST:8000/streams`.

Содержимое примера файла `streams`:

```
{
  "streams": [
    {
      "name": "channel1",
      "title": "Тестовый канал",
      "comment": "Демо-стрим из внешнего конфига",
      "inputs": [
        {
          "url": "fake://fake"
        }
      ]
    }
  ]
}
```

Для продакшена нужен сервис, который отдаёт ответ с `Content-Type: application/json` и при необходимости обрабатывает параметр `cursor`.

Как включить config external

Для этого надо в файле настройки инсталляции добавить секцию `config_external`:

```
config_external:
  url: "https://auth@middleware.myservice.com"
```

Технические детали протокола config_external

Полное описание API приведено в формате OpenAPI 3.1 в файле `catena-se.yaml` в этой же папке. Там заданы схемы `streams_list`, `stream_config`, типы входов стримов (`stream_input_*`), параметры DVR, лейблы и остальные поля, которые ваш сервис может возвращать в `streams_list` для провижининга стримов в Catena SE.

2.2.8 Управление темплейтами

Темплейты обработки телеканалов в Catena SE - главный и рекомендованный способ единообразной настройки системы.

Использование темплейтов позволит унифицировать все настройки и избежать ошибок по невнимательности при изменении правил транскодирования, авторизации и т.п.

Темплейт всегда присутствует у телеканала, даже если в нём нет никаких настроек.

Шаги:

1. Создание нового темплейта
2. Редактирование его полей
3. Создание нового канала с этим темплейтом
4. Проверка, что все настройки применились.

2.2.9 Транскодирование в MBR

Для проигрывания на Smart TV, STB и мобильных телефонах телеканал нужно транскодировать в несколько качеств, т.е. MBR.

Чтобы не ошибиться с подбором железа, мы рекомендуем купить у нас уже собранный транскодер. При этом вы всегда можете поэкспериментировать и подобрать железо на своё усмотрение.

Последовательность шагов:

- Создаем новый поток
- Открываем раздел транскодера
- Добавляем три качества: 1080, 720 и 360
- Сохраняем
- Дожидаемся плеера
- Проверяем, что плеер может переключаться между ними

1. Создаём новый поток

Перейдите в раздел **Каналы**, введите имя канала (например, `mbr-demo`) и создайте канал.

После создания откроется страница канала. Добавьте источник (например, `fake://`) и сохраните.

2. Открываем раздел транскодера

Откройте вкладку **Транскодер** на странице канала.

The screenshot shows the 'Catena SE' software interface. On the left is a dark sidebar with white icons and text: 'Dashboard', 'Streamers', 'Templates', 'Channels' (which is the active tab), 'VOD', 'Protection', 'Logs', 'Logout', and 'Language' set to 'English'. The main right area is titled 'Channels <- mbr-demo'. It has tabs for 'BASIC SETTINGS', 'SOURCES', 'PROCESS' (which is underlined in blue), 'EGRESS', 'DVR', and 'PROTECTION'. Below these tabs is a large blue button labeled 'ENABLE TRANSCODER'. At the top right are three buttons: 'SAVE' (grey), 'CANCEL' (grey), and 'DELETE' (red).

3. Добавляем качества 1080, 720 и 360

В настройках транскодера добавьте три профиля (например, 1080p, 720p и 360p).

Channels <- mbr-demo

PROCESS

Segment Duration (ms) Deinterlace DISABLE TRANSCODER

Tracks

Audio Codec AAC Bitrate 64000

Video Track 2 x Delete

Codec H.264 Bitrate (kbit/s) Width Height B-frames

Resize Strategy (optional)

Strategy Background Used only with 'fit' strategy

+ ADD VIDEO TRACK

Burn-in Configuration▶

4. Сохраняем

Нажмите **Сохранить** и дождитесь применения настроек.

Channels <- mbr-demo

PROCESS

Segment Duration (ms) Deinterlace **DISABLE TRANSCODER**

Tracks

Audio Codec: AAC Bitrate: 64000

Video Track 2 **x Delete**

Codec: H.264 Bitrate (kbit/s): 3000 Width: 1080 B-frames

Resize Strategy (optional)

Strategy Background Used only with 'fit' strategy

Video Track 3 **x Delete**

Codec: H.264 Bitrate (kbit/s): 1000 Width: 720 B-frames

Resize Strategy (optional)

Strategy Background Used only with 'fit' strategy

Video Track 4 **x Delete**

Codec: H.264 Bitrate (kbit/s): 400 Width: 360 B-frames

Resize Strategy (optional)

Strategy Background Used only with 'fit' strategy

+ ADD VIDEO TRACK

Burn-in Configuration▶

5. Дожидаемся плеера и проверяем переключение качества

Перейдите на вкладку **Основное** и убедитесь, что встроенный плеер показывает поток. Убедитесь, что можно переключаться между качествами (1080, 720, 360).

Channels <- mbr-demo

BASIC SETTINGS SOURCES PROCESS EGRESS DVR PROTECTION

Name: mbr-demo Title: Select a template to apply defaults

Status: streamer3

Player



mbr-demo
2026-02-09T19:08:55Z
2026-02-09T19:08:55Z
1770664135.171988
streamer@server.1

SAVE CANCEL DELETE

2.2.10 Запись телепередач

Catena SE позволяет записывать телеканал на диск для последующего проигрывания как уже завершенных телепередач, так и текущих, записывающихся прямо сейчас.

В темплейтах, идущих в поставке Catena SE запись архива уже включена по умолчанию и для этого будут использоваться ресурсы серверов, которые вы предоставили при [добавлении сервера](#)

Запись идет всех видеокачеств и вместе с ними так же и скриншоты.

По API можно запросить список телепередач на нужный канал и если middleware отвечает на нужный запрос в протоколе `config-external`, то Catena SE вернет ответ с передачами в которых будет заполнено поле статуса записи.

Архитектура записи архива

АРХИВ ПИШЕТСЯ ПОСЛЕ ТРАНСКОДЕРА

Архив в Catena SE пишется **после транскодера**: на диск сохраняется уже подготовленное для доставки видео (все качества/дорожки, которые вы настроили в темплейте).

Это важно по двум причинам:

- архив сразу пригоден к проигрыванию на клиентских устройствах (Smart TV / STB / mobile) без повторной обработки;
- нагрузка и требования к хранилищу считаются по фактическому выходному набору профилей (MBR), а не по "сырым" входным потокам.

ХРАНИЛИЩЕ: ТОЛЬКО ЛОКАЛЬНЫЕ ДИСКИ + РЕПЛИКАЦИЯ МЕЖДУ СЕРВЕРАМИ

Catena SE пишет архив **строго на локальные диски серверов**.

Мы **не используем** NFS, дисковые полки/SAN и другие общие сетевые хранилища: они добавляют сложность и часто ухудшают предсказуемость задержек и надёжность.

Рекомендуемая модель хранения:

- много недорогих отдельных дисков (обычно HDD) под большой объём архива;
- репликация архива **между серверами кластера**, чтобы переживать отказ отдельного диска/сервера;
- конфигурация дисков и точек монтирования задаётся при добавлении сервера (см. [Добавление нового сервера](#)).

SSD cache для массового проигрывания архива

Чтобы обслуживать большое количество одновременных зрителей, Catena SE использует подход "много архива на HDD, самое горячее – на SSD".

Типовой сценарий:

- основной объём архива хранится на локальных HDD;
- наиболее часто запрашиваемые фрагменты (например, "последние часы" или "прайм-тайм") попадают в SSD cache;
- это позволяет резко увеличить количество параллельных проигрываний без раздувания стоимости хранения всего архива на SSD.

Как это обычно используется в IPTV (catchup TV)

Задачи, которые чаще всего хотят реализовать поверх архива:

- поставить прямой эфир на паузу и продолжить с того же места
- посмотреть текущую передачу с начала (startover)
- посмотреть вчерашнюю передачу по EPG
- сохранить "прогресс" и продолжить просмотр позже

Технически эти сценарии проще всего решаются в RTSP, но в IPTV/OTT в основном используются сегментные HTTP-протоколы (HLS, DASH, и иногда MSS), поэтому для паузы/перемотки и "startover" используются специальные URL и плейлисты.

Время и идентификаторы передач

В этой документации время передачи рассматривается **только** как Unix timestamp в **секундах UTC (Epoch)**.

Ни при каких обстоятельствах не используйте локальное время в логике проигрывания архива: это почти гарантированно приводит к ошибкам на смене часовых поясов, DST и при работе с пользователями из разных регионов.

Рекомендуется при просмотре передачи сохранять текущее положение (UTC timestamp или offset) в базе middleware, чтобы при повторном открытии предлагать "продолжить" или "начать сначала".

Проигрывание передачи из архива по EPG (VOD-плейлист)

Если middleware хранит расписание EPG, то у каждой передачи есть:

- `from` — время начала в UTC (epoch)
- `to` — время конца в UTC (epoch)

Длительность передачи вычисляется как `(duration = to - from)`.

Для проигрывания передачи из архива сформируйте URL по правилам из статьи [Проиграть ТВ канал](#).

Пример для HLS:

```
https://<hostname>/lb/-/<channel>/archive-<from>-<duration>.m3u8
```

Пример для DASH:

```
https://<hostname>/lb/-/<channel>/Manifest-<from>-<duration>.mpd
```

В плеере при таком доступе обычно работают:

- перемотка внутри передачи
- пауза
- ускоренный просмотр

При завершении проигрывания рекомендуется автоматически переключить пользователя на **следующую** передачу по EPG, чтобы сохранить непрерывность просмотра.

Просмотр текущей передачи: event-плейлисты и "пауза лайва"

Для "startover" и паузы текущей передачи удобно использовать тот же URL, что и для архива, но так, чтобы конец окна попадал в "настоящее" или в будущее.

Хитрость в том, что когда указанный конец периода находится в будущем, плейлист отдаётся не как VOD, а как EVENT, и клиент может его переопрашивывать и двигаться дальше.

Если ваша клиентская платформа позволяет, используйте тот же URL, что и для архива, добавив параметр `event=true`:

```
https://<hostname>/lb/-/<channel>/archive-<from>-<duration>.m3u8?event=true
```

Где `from` и `duration` соответствуют текущей передаче по EPG.

ВАЖНОЕ ЗАМЕЧАНИЕ ПРО КНОПКУ "LIVE"

Кнопку "live" почти всегда нужно программировать отдельно.

Пока пользователь стоит на паузе, ситуация может измениться:

- плейлист EVENT может со временем превратиться в VOD (окно "закроется");
- плеер может перестать обновлять плейлист;
- при переходе передачи плейлист для "текущей" передачи закончится.

В этом случае нужно:

- по EPG определить текущую передачу,
- пересобрать URL на актуальный период,
- либо перепрыгнуть на следующий элемент EPG.

ОГРАНИЧЕНИЯ ПЛАТФОРМ

Event-плейлисты и перемотка "внутри текущей передачи" могут иметь ограничения на отдельных платформах (например, в нативном Safari).

Если для вас критичны такие платформы, закладывайте отдельную реализацию таймлайна/перемотки в приложении (обычно на JavaScript/в нативном SDK) поверх тех же EPG данных и URL.

2.2.11 Сдвиг времени (time-zone compensation)

Эта страница отвечает на частую задачу локального провайдера (edge/headend), когда ТВ-каналы приходят от апстрайм-провайдера, а абоненты находятся в другом часовом поясе.

Пример задачи клиента:

- принять каналы один раз у провайдера;
- раздавать локально (1-to-many), экономя WAN bandwidth;
- для выбранных "премиум" каналов сделать **фиксированный сдвиг**, например (6h), и дать **catchup**.

Catena SE решает эту задачу через архив (DVR/nPVR): сначала записываем канал, затем читаем архив со сдвигом.

Важное про время

- Все времена в API/ссылках рассматривайте **в UTC epoch (секунды)**.
- Сдвиг (6h) – это 21600 секунд.

Требования

- Канал должен писаться в архив.
- Глубина архива должна быть **больше**, чем желаемый сдвиг. Для (6h) практично держать запас (например, 8–24 часа), чтобы переживать "дыры" и инциденты источника.

Способ 1 (рекомендуемый для популярных каналов): отложенный канал

Это лучший вариант для задачи "1-to-many": архив читается **один раз**, независимо от количества зрителей.

Идея:

- есть исходный канал `channel`, который пишется в архив;
- создаётся второй канал, который читает архив с постоянным сдвигом.

Пример концепции:

- исходный канал пишет архив;
- отложенный канал читает его с задержкой 21600 секунд (6 часов).

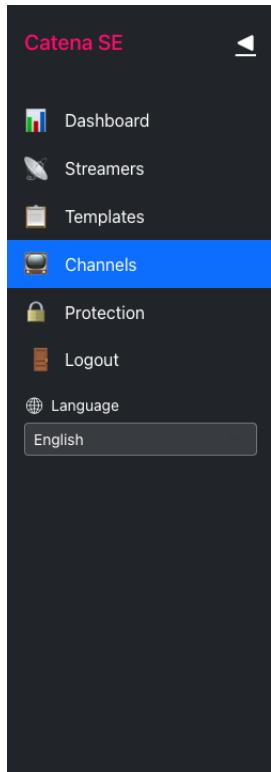
Такой "отложенный" канал дальше раздаётся как обычный live-канал и попадает в общий балансировщик /lb/.

Если вы делаете "отложенный канал" как отдельный объект в Catena SE, то (по аналогии с Flussonic Media Server) его источник можно задать как чтение архива исходного канала с фиксированным сдвигом, например:

```
timeshift://channel/21600
```

Пример настройки в интерфейсе (исходный канал `recorded`, сдвиг на 1 час – 3600 секунд):

1. В разделе **Каналы** уже есть канал с архивом (например, `recorded`). Создайте новый канал для отложенного просмотра.



Channels

New Stream Name

Enter stream name

Create

Stream Name	Source URL	Status	Input Resolution	Input Bitrate	Transcoder Outputs	Output Bitrate	Viewers	DVR	Actions
recorded	fake://	●	SD	N/A	-	N/A	0	2d	▶

1. Откроется страница нового канала. Перейдите во вкладку **Источники**.



Channels <- delayed

Save

Cancel

Delete

Basic Settings

Sources

Process

Egress

DVR

Protection

Name

delayed

Title

Template

No template

▼

Select a template to apply defaults

Disabled

Static



Status



Player

Pretty print □

```
{"errors": [{"code": "streaming_service get_online_streamer err: no live streamer"}]}
```

1. Добавьте источник `timeshift://recorded/3600` (или `timeshift://recorded/21600` для сдвига 6 часов) и сохраните.

1. На вкладке **Основное** встроенный плеер показывает поток со сдвигом.

Как это сочетается с catchup (EPG)

Time-zone compensation и catchup решают разные задачи и обычно используются вместе:

- **time-zone compensation**: “смотреть эфир как будто он идёт в моём времени” (постоянный сдвиг, например 6 часов);
- **catchup**: “посмотреть конкретную передачу по EPG” (архив по `from / duration`).

Рекомендация для middleware:

- если вы показываете абоненту “отложенный канал” как отдельный live-канал, то и EPG на UI обычно нужно отображать со сдвигом (на стороне middleware);
- для catchup URL по EPG используйте те же правила, что в [nPVR](#) и [playback](#) (UTC epoch, `archive-<from>-<duration>`).

Практическая привязка к фиксированному сдвигу:

- если вы показываете EPG для “отложенного канала” со сдвигом (+21600), то при формировании catchup-ссылки на архив исходного канала используйте `from = from_displayed - 21600`;
- пример: пользователь выбрал передачу, которая в “сдвинутом” EPG начинается в `from_displayed`. Для проигрывания по HLS сформируйте ссылку:

```
https://<hostname>/lb/-/<name>/archive-<from_displayed-21600>-<duration>.m3u8
```

Практические замечания для edge/headend

- Для таких “отложенных каналов” нагрузка возрастает несущественно и вещание происходит в основном из оперативной памяти, клиенты не обращаются к диску.
- Планируйте место на диске от битрейта: (channels \times bitrate \times hours).

2.2.12 Отправка ТВ канала в мультикаст

В гостиничных и операторских IPTV-сетях часто нужно отдавать телеканалы в локальную сеть **по UDP multicast** (обычно в контейнере MPEG-TS).

Catena SE умеет “пушить” (рассылать) телеканал в указанную мультикаст-группу – как правило, это делается для совместимости с телевизорами/приставками, которые ожидают IPTV по UDP.

Предварительные требования

- Сеть должна корректно поддерживать multicast (IGMP/IGMP snooping, маршрутизация между VLAN при необходимости).
- Узлы Catena SE должны отправлять multicast в тот интерфейс, который смотрит в локальную сеть (а не в интернет).

Базовая настройка

Настройка отправки в multicast выполняется **на самом канале**, в разделе **Push**.

Для темплейта эта настройка **недоступна**, потому что не имеет смысла: push – это “куда и как отдавать именно этот канал” (конкретная мультикаст-группа/порт), а не правило обработки контента.

Откройте вкладку **Push**.

Чтобы отправлять канал в UDP multicast, добавьте push на нужный адрес.

Типовой формат:

udp://239.0.0.1:5000

Где:

- 239.0.0.1 – IP multicast-группы;
- 5000 – UDP порт.

Стандартная схема в локальных сетях – один и тот же порт для всех каналов (например, 5000), а различаются только IP адреса групп:

- udp://239.0.10.1:5000 – Channel 1
- udp://239.0.10.2:5000 – Channel 2
- udp://239.0.10.3:5000 – Channel 3

Так проще сопровождать сеть (ACL/файрволы/плееры), и проще объяснять адресацию эксплуатации.

Расширенные опции (по необходимости)

Ниже – несколько опций, которые часто полезны в IPTV-сетях.

ВЫБОР ДОРОЖЕК

Если в канале есть несколько дорожек, можно явно выбрать, что отправлять:

```
udp://239.0.0.1:5000?tracks=v1a1
```

Где:

- v1 – первая видеодорожка;
- a1 – первая аудиодорожка.

НАСТРОЙКА PID-ОВ

При необходимости можно настраивать PID-ы выходного MPEG-TS потока (PMT/PNR и PID-ы дорожек).

Это используется, когда ваша ТВ/приставки/головная станция ожидают определённую сигнализацию или фиксированные PID-ы.

БОНУС: ПОДГОТОВКА CBR UDP ПОТОКА

При отправке в UDP Catena SE может дополнительно подготовить CBR-поток (constant bitrate), что полезно для некоторых IPTV сетей и оборудования.

Ограничения авторизации

Важно: **protection (авторизация по токену)** не работает для UDP multicast.

UDP multicast не является “клиент-серверным” протоколом: сервер не видит отдельных клиентов и не может проверять токены на просмотр.

Если вам нужна защита просмотра в сети, обычно используют:

- изоляцию сети/VLAN/ACL;
- CAS/шифрование на уровне IPTV контура (в зависимости от вашей схемы).

MPTS и расширенные настройки мультиплексоров

Сборка нескольких каналов в **MPTS** и расширенные настройки мультиплексоров (SI/NIT/SDT, LCN, “multiplexer”) доступны в отдельном продукте **MCaster**.

Проверка

Проверить, что multicast пошёл, можно на любом хосте в той же сети (или на ТВ/приставке).

Примеры:

- в VLC: открыть сетевой поток `udp://@239.0.0.1:5000`
- в ffplay: `ffplay udp://@239.0.0.1:5000`

Если проигрывание не стартует, чаще всего проблема в сети (IGMP/маршрутизация/ACL), а не в Catena SE.

2.2.13 Проиграть ТВ канал

Catena SE поддерживает следующие протоколы проигрывания:

- HLS
- DASH
- MSS
- LL-HLS
- CMAF

Эти протоколы являются основными для современной ТВ индустрии и позволяют проигрывать видео на:

- SmartTV
- Android (phone, STB, TV)
- iPhone
- STB

Урлы проигрывания

Урлы для проигрывания телеканалов формируются из хостнейма самой инсталляции и имени канала.

Назначение	Протокол	Урл
Проигрывать live	HLS	https://lb/-//index.m3u8
	DASH	https://lb/-//Manifest.mpd
	MSS	https://lb/-/.ism/Manifest
	LL-HLS	https://lb/-//index.ll.m3u8
Проиграть архивную передачу	HLS	https://lb/-//archive--.m3u8
	DASH	https://lb/-//Manifest--.mpd
	MSS	https://lb/-/(archive=-).ism/Manifest
	LL-HLS	https://lb/-//archive--.ll.m3u8
Проиграть архив в режиме псевдо-live	HLS	https://lb/-//timeshift_abs-.m3u8

Важно отметить, что для архива LL-HLS бесполезен, потому что для архива понятие низкой задержки не имеет смысла, но этот способ доступа существует для упрощения работы и единообразия.

Режим fMP4 является режимом по-умолчанию для HLS.

2.2.14 Скриншоты для перемотки

Для удобной перемотки стриминговая подсистема должна уметь делать скриншоты.

В Catena SE используются **JPEG-скриншоты**.

JPEG-скриншоты можно использовать для:

- предпросмотра (preview) при перемотке;
- быстрого “пролистывания” архива;
- оценки качества картинки.

Нагрузка и транскодер

JPEG-скриншоты требуют дополнительных ресурсов CPU, потому что кадр нужно декодировать и закодировать в JPEG.

В Catena SE скриншоты **делаются автоматически при запущенном транскодере** (рекомендуемый режим).

Если транскодер выключен, включение JPEG-скриншотов приведёт к **росту нагрузки** на сервер, поэтому планируйте CPU с запасом.

Включение скриншотов делается в настройках темплейта или отдельно для стрима.

Можно задать отдельный размер скриншотов.

Сами скриншоты хранятся в архиве вместе с видео и доступны для просмотра по апи.

Как получить live-скриншот

Последний live-скриншот потока доступен по URL:

```
https://<hostname>/lb/-/<channel>/preview.jpg
```

Как получить скриншот из архива

Скриншоты в архиве адресуются временем в **UTC epoch (секунды)**.

Скриншот для момента времени `utc`:

```
https://<hostname>/lb/-/<channel>/<utc>.jpg
```

Также поддерживается человекочитаемый формат времени GMT:

```
https://<hostname>/lb/-/<channel>/YYYY/MM/DD/HH/MM/SS.jpg
```

Скриншоты, встроенные в плейлисти

Catena SE может добавлять ссылки на скриншоты прямо в HLS/DASH плейлист (или манифест) – это удобно для UI перемотки, потому что плеер получает таймлайн превью “в комплекте” с плейлистом.

Для этого добавьте параметр `thumbnails=<N>` к URL проигрывания.

`N` – это количество ссылок на скриншоты, которое будет добавлено в плейлист, чтобы покрыть длительность окна (live/DVR/VOD). Чем больше `N`, тем плотнее сетка превью, но тем больше нагрузка на клиент и сеть.

Примеры:

- Live HLS:

<https://<hostname>/lb/-/<channel>/index.m3u8?thumbnails=50>

- Архив HLS (catchup):

<https://<hostname>/lb/-/<channel>/archive-<from>-<duration>.m3u8?thumbnails=50>

- Архив DASH:

<https://<hostname>/lb/-/<channel>/Manifest-<from>-<duration>.mpd?thumbnails=50>

2.2.15 Настройка авторизации проигрывания

Без авторизации проигрывания инсталляция Catena SE доступна всем желающим в интернете и довольно быстро превратится в бесплатный источник для чужих сервисов.

Чтобы смотреть могли только ваши абоненты, необходимо настроить авторизацию и связать её с вашей middleware.

В Catena SE базовый подход такой:

- клиент получает URL проигрывания, в котором есть **токен**;
- Catena SE проверяет этот токен локально (securelink) или через ваш backend (middleware).

Правила проверки токена описываются в авторизационных бекендах.

Авторизационный бекенд - это набор правил, позволяющих:

1. сразу разрешить клиенту просмотр на основании его IP адреса или известного токена
2. сразу запретить просмотр на основании геолокации или других признаков
3. проверить токен по определенным правилам локальной
4. отправить токен на проверку в стороннюю систему, например middleware

Токены

Токен – это строка, которая добавляется к URL проигрывания (в query string) и позволяет:

- отличить “своего” клиента от “чужого”;
- ограничить время жизни ссылки (TTL);
- (опционально) привязать просмотр к IP/устройству/сессии;
- (опционально) собирать статистику и ограничивать мультискрин.

Практическое правило: токен должен быть **короткоживущим** и генерироваться вашей системой выдачи ссылок (портал/ middleware).

Securelink без backend middleware

Если вы не хотите (или пока не можете) делать отдельный backend для проверки прав, можно использовать “securelink” – проверку токена по общей формуле с секретным ключом.

Схема работы:

- ваш портал генерирует токен по формуле и хеширует его секретным ключом;
- клиент открывает поток с этим токеном;
- Catena SE пересчитывает ожидаемый токен тем же способом и сравнивает;
- если совпало – доступ разрешён, если нет – запрещён.

Обычно в токен включают:

- имя канала (или другой идентификатор контента),
- время начала/окончания действия токена (TTL),
- (опционально) IP клиента,
- (опционально) `user_id` (для ограничения количества одновременных сессий).

Плюсы securelink:

- не нужен отдельный сервис проверки на каждый просмотр;
- высокая производительность и простая эксплуатация.

Минусы securelink:

- сложнее сделать тонкие политики (пакеты каналов, гео, устройства, аудит) без усложнения токена;
- при компрометации ключа придётся оперативно его менять.

Для включения securelink, надо включить возможность этой проверки в настройках авторизационного бекенда.

Как работает backend middleware

В “классической” схеме Catena SE проверяет доступ через backend (middleware) с помощью механизма `on_play`.

Схема работы:

- клиент запрашивает URL у middleware;
- middleware возвращает URL с уникальным токеном;
- при открытии новой сессии Catena SE обращается к backend и спрашивает, можно ли проигрывать;
- middleware отвечает “разрешить/запретить” и (опционально) возвращает дополнительные параметры сессии.

Важно: Catena SE **не ходит в middleware на каждый сегмент**. Проверка делается при открытии сессии и затем периодически перепроверяется (это снижает нагрузку на backend).

Этот вариант лучше всего подходит, когда вам нужно:

- включать/отключать доступ по подписке в реальном времени;
- ограничивать мультискрин и/или собирать статистику;
- применять сложные правила (гео, устройства, пакеты, родительский контроль и т.п.).

Session keys: что это и зачем

Catena SE ведёт понятие “сессии” проигрывания. Сессия идентифицируется на основе набора ключей (например: протокол, имя канала, IP, токен).

По умолчанию (идеологически) это выглядит так:

- `session_id = hash(name + ip + proto + token)`

Если меняется любой из ключей, с точки зрения сервера это **новая** сессия, и он может:

- заново проверить доступ у backend,
- заново применить лимиты (например, “сколько одновременных просмотров разрешено”),
- пересоздать состояние (что иногда видно пользователю как “переподключение”).

Зачем управлять session keys:

- **мобильные сети**: у клиента может меняться IP при переходе между сотами;
- **NAT/прокси**: реальный IP может быть нестабильным или общим для группы пользователей;
- **устойчивость сессий**: иногда важнее “не рвать просмотр” при смене IP, чем жёстко привязывать токен к IP.

Компромисс здесь простой:

- чем меньше параметров участвует в идентификаторе сессии, тем стабильнее просмотр при сетевых изменениях;
- чем больше параметров (например, IP), тем сильнее защита от передачи ссылки “другому человеку”.

Если вы используете middleware backend, настройку session keys имеет смысл фиксировать в вашей архитектуре авторизации как отдельное решение “безопасность vs UX”.

Для настройки ключей сессии надо редактировать их список в авторизационном бекенде.

Multiauth

Если один стриминговый кластер работает с несколькими middleware, есть возможность проверять каждый токен на разных серверах.

Для этого достаточно в конфигураторе авторизационного бекенда указать несколько серверов.

2.2.16 Авторизация на нескольких middleware

С точки зрения маркетинга и развития бизнеса нормальная практика - поддерживать одну инсталляцию стриминговой инфраструктуры на несколько разных клиентских порталов.

В такой схеме надо проверять токен проигрывания в разных порталах, для этого поддерживается концепция различных апстримов авторизации.

В настройках авторизационного бекенда достаточно указать несколько разных апстримов.

Проверка будет происходить на всех урлах одновременно и параллельно. Первый ответивший положительно прерывает остальные запросы.

2.3 Обслуживание

2.3.1 Дашборд Catena SE

Главная страница Catena SE помогает ответить на вопрос: надо ли что-то чинить или всё ок.

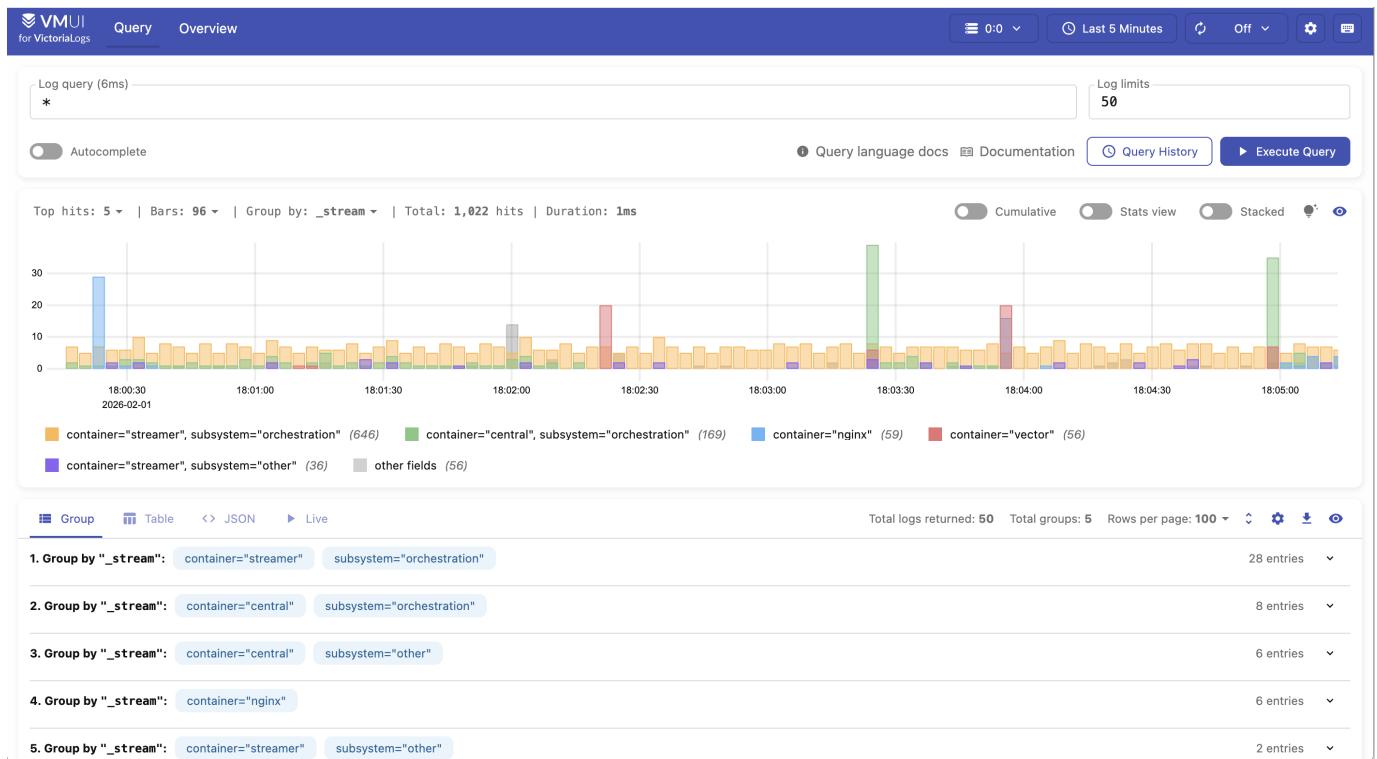
Для этого на ней выводятся следующие панели:

- График сетевого трафика на вход и выход
- График нагрузки транскодеров
- Алерты об потоках, которые стали непригодны
- Алерт о недоступности или проблемах сервера авторизации
- Алерт об аппаратных проблемах на серверах
- Предупреждение о недоступности резервных источников

С помощью этого дашборда можно спрогнозировать возможности вашего кластера и своевременно [добавить новые сервера](#)

2.3.2 Доступ к логам

Логи Catena SE помогают диагностировать проблемы приёма, транскодирования и проигрывания. Доступ к ним осуществляется через веб-интерфейс: пункт меню **Логи** на главной странице.



Откуда берутся логи

В логах объединяются сообщения от компонентов Catena SE:

- **Central** – управление каналами, конфигурация, API, балансировка;
- **стримеры (Media Server)** – приём источников, транскодер, запись архива, отдача потоков клиентам.

При поиске по имени канала вы видите события, связанные с этим каналом на всех узлах кластера.

Поиск по логам (LogQL)

Вы можете искать по логам для диагностики сбоев и аномалий с помощью запросов в стиле LogQL.

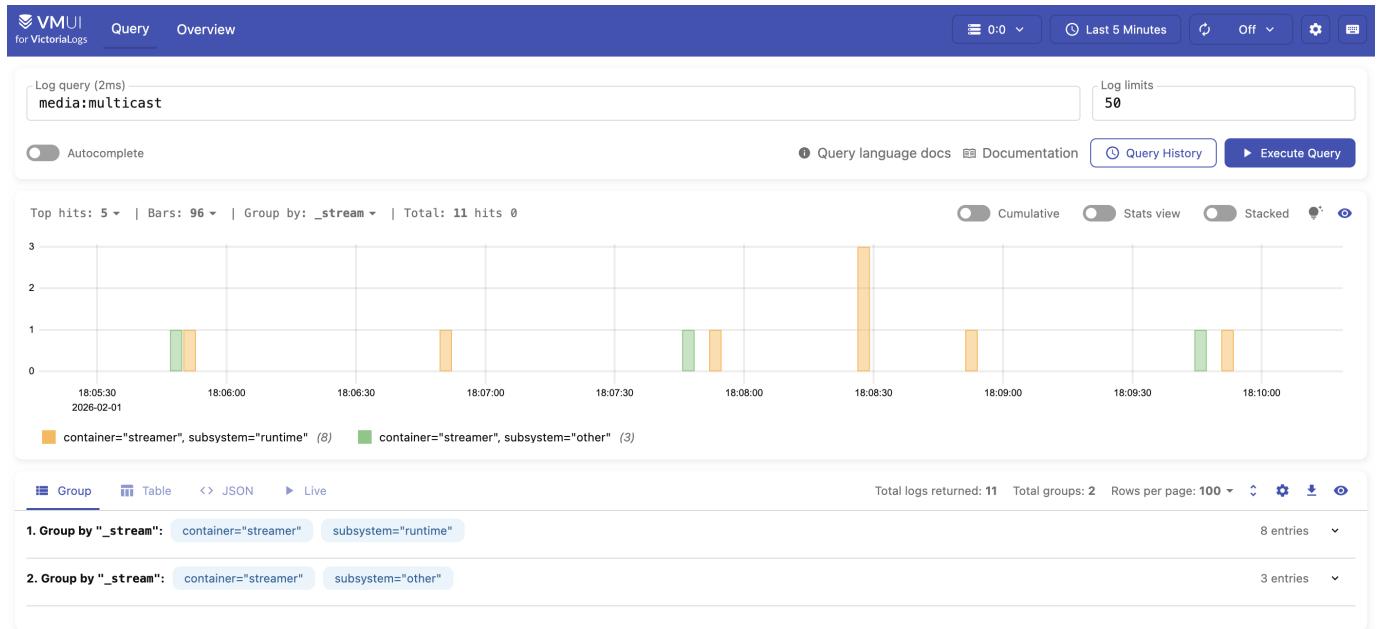
Типичные сценарии:

- события по конкретному каналу – укажите имя потока, например `media:multicast`;
- ошибки приёма или публикации – фильтр по уровню и тексту сообщения;
- таймлайн инцидента – выберите временной диапазон и отфильтруйте по потоку или узлу.

Пример запроса по имени канала:

```
media:multicast
```

Так вы увидите все записи, относящиеся к потоку `multicast`.



Когда смотреть логи

Логи полезны, когда:

- канал не стартует или обрывается (проверьте ошибки источника и транскодера);
- при воспроизведении возникают сбои (ошибки отдачи, таймауты);
- нужно понять причину переключения на резервный источник или перезапуска канала.

Для систематической диагностики используйте также [дашборд](#) и [алерты](#).

2.3.3 Бекап базы данных

Для надёжной работы Catena SE необходимо регулярно сохранять бекап базы данных и уметь его восстанавливать.

Доступа к PostgreSQL извне кластера нет. Бекап и восстановление выполняются через под `db-0` в namespace `catena-se`.

Создание бекапа вручную

Выполните на хосте с настроенным `kubectl` и доступом к кластеру:

```
kubectl -n catena-se exec db-0 -- backup
```

Бекап создаётся внутри пода. Имя файла формируется по времени (например, `dump-2026-02-19-12-40-40.sql.gz`).

Список доступных бекапов

Чтобы посмотреть список сохранённых бекапов:

```
kubectl -n catena-se exec db-0 -- list
```

Пример вывода:

```
$ kubectl -n catena-se exec db-0 -- list
2026-02-19-12-40-40
2026-02-19-12-40-47
```

Это метки времени, которые нужно подставлять в команду восстановления.

Восстановление из бекапа

Для восстановления укажите нужную метку времени:

```
kubectl -n catena-se exec db-0 -- restore 2026-02-19-12-40-47
```

Пример сеанса восстановления:

```
$ kubectl -n catena-se exec db-0 -- restore 2026-02-19-12-40-47
Restoring from dump-2026-02-19-12-40-47.sql.gz
DROP DATABASE
CREATE DATABASE
SET
SET
SET
SET
SET
set_config
-----
(1 row)

SET
SET
SET
...
CREATE INDEX
CREATE INDEX
ALTER TABLE
Restore completed
```

Во время восстановления Central лучше остановить или не выполнять в нём изменений.

Регулярные бекапы и отправка в облако

Система настроена на ежедневные бекапы в 3 ночи.

При желании вы можете самостоятельно настроить отправку бекапов в S3 или другой сторадж, забирая файлы из каталога `/backup` на головном сервере.

2.3.4 Добавление нового сервера

Последовательность действий:

1. Устанавливаем ОС
2. Размечаем и монтируем диски
3. Добавляем в кластер кубернетеса
4. Добавляем описание нового стримера

Установка ОС

На сервере необходимо установить Ubuntu Server 24.04.

Обязательно надо правильно настроить hostname на этом сервере, после добавления в кластер поменять имя сервера уже будет невозможно.

Разметка дисков

На сервере для вещания ТВ могут находиться выделенные диски для архива. Этот шаг нужно делать только в случае, когда вы собираете много дисков для хранения архива.

Диски монтируются в подкаталоги одного каталога:

```
/storage/disk1
/storage/disk2
/storage/disk3
```

Диски надо добавить в описание стримера в файле конфигурации `catena-se.yaml`:

```
streamers:
  hosts:
    - name: s2
      node: streamer2
      disks:
        - path: disk1
        - path: disk2
        - path: disk3
```

Добавление в кластер

Для подключения понадобится секретный токен, который хранится на первом сервере в `/var/lib/rancher/k3s/server/node-token`:

```
sudo cat /var/lib/rancher/k3s/server/node-token
```

`private_ip` - это IP адрес под которым первый сервер кластера виден из нового. Это вполне может быть серый IP адрес.

```
curl -sL https://get.k3s.io | sh -s - agent -t ${token} --server https://${private_ip}:6443
```

Добавляем новый стример

Далее надо добавить описание этого стримера в `catena-se.yaml` на первом сервере

```
streamers:
  hosts:
    ...
    - node: streamer2
      host: s2.mystreamingservice.com
```

Возможно вы уже начали добавлять эту запись выше, в разделе с дисками.

и добавляем описание в приложение:

```
curl -sL https://flussonic.ru/doc/catena-se/install.sh | sh -s - app --values ./catena-se.yaml
```

2.3.5 Вывод сервера из работы

Последовательность шагов:

1. Снимаем нагрузку со стримера в Catena SE
2. Опционально реплицируем с него весь контент, если время позволяет
3. Если на нём находится база данных, переносим её
4. Снимаем с него все поды: `kubectl drain <node> --ignore-daemonsets --force --delete-emptydir-data`
5. Удаляем ноду из кубернетеса: `kubectl delete node <node>`
6. Удаляем кубернетес с этого сервера: `k3s-agent-uninstall.sh`

2.3.6 Работа в кластере

Catena SE спроектирована как кластерное ПО со следующими возможностями:

- Балансировка нагрузки на транскодер
- Реализация расширяемого резервируемого хранилища видео на дисках
- Балансировка проигрывания
- Автоматическое реагирование на поломки оборудования

Распределение транскодирования

Catena SE имеет встроенный механизм, позволяющий распределить обработку телеканалов по доступным серверам и видеокартам.

Механизм не требует настройки, однако вариативность железа, подобранныго клиентами самостоятельно может превышать даже самые смелые ожидания. В случае возникающих вопросов к качеству работы интеллектуального балансировщика транскодирования, обращайтесь в поддержку.

Дисковое хранилище enterprise уровня

Два десятилетия опыта по хранению видео упакованы в Catena SE. Мы предоставляем ПО, позволяющее превратить дешевые сервера без дорогостоящего оборудования в полноценное приватное облачное хранилище прямых эфиров.

Мы категорически не рекомендуем использование сетевых СХД, за которыми вы не следите самостоятельно и не меряете 95% перцентиль времени доступа. Если вы просто поставили какую-то СХД и рассчитываете на то, что это подойдет для оказания качественного сервиса, то вам стоит воспользоваться нашим мониторингом проблем с DVR, в котором хорошо видны все проблемы чтения.

Как правило, на графике проблем хорошо видны сетевые хранилища.

Балансировка проигрывания

Каждый сервер, включенный в кластер Catena может работать балансировщиком проигрывания.

Для этого достаточно обратиться к телеканалу с префиксом: `/1b/-/<channel-name>` и произойдет автоматическое перенаправление на тот стример, на котором сейчас есть нужный телеканал.

Автоматическое реагирование на поломки оборудования

Catena SE может автоматически реагировать на выход из строя серверов.

При выключении сервера каналы с него автоматически перенесутся на другие работающие сервера.

Для того, чтобы это не привело к потере сервиса, стоит оставлять нагрузку ниже предельной и планировать выбор серверов так, чтобы оставалась свободная мощность. Один мощный сервер хуже трех средних, вы фактически сделаете сервис непригодным к обслуживанию и поддержке.

Резервирование балансировщика сложнее, потому что у вас есть следующий выбор:

- Добавить несколько разных IP адресов к DNS имени в надежде на то, что клиент будет сам перебирать адреса в поисках рабочего. На практике на такое стоит рассчитывать только с веб-браузерами. Помните, что кеширование DNS составляет часы и даже дни, т.е. ваши изменения будут долго распространяться по клиентам
- Добавить в клиентское приложение несколько хостнеймов, по которым приложение будет обращаться за проигрыванием. Это самый простой и практичный вариант.
- Реализовать самую сложную конструкцию с плавающим IP адресом. Для этого надо самостоятельно настроить ваше сетевое оборудование так, чтобы роутер отслеживал, какой из серверов сейчас жив и обслуживал выделенный IP адрес балансировщика именно на живом сервере. Это потребует от вас настроек протоколов BGP, OSPF или аналогов CARP. Такое может потребоваться при работе со старыми и необслуживаемыми клиентами.