

# Руководство по Watcher

22 мая 2019

Введение в Flussonic Watcher

Аппаратные требования

Установка Watcher

База данных Watcher

Настройки

Сброс пароля

Фейловер захвата

Прием событий о движении с камеры

События распознавания автомобильных номеров

Auto-login

Пользовательский интерфейс

Мобильные приложения

Кастомизация интерфейса

Брендирование email

Watcher API

Импорт пользователей по API

Импорт камер по API

Интеграция в существующую систему

Бэкэнд для авторизации пользователей

RADIUS

API для мобильных приложений

Управление камерами

Создание клиентской мозаики

Управление пользователями

Группы

Предоставление доступа к камерам

Серийные номера

Импорт и Поиск камер

# Введение в Flussonic Watcher

## Flussonic Watcher

— программный комплекс, устанавливаемый на сервера и камеры клиента, для трансляции видео, записи и предоставления персонализированного удаленного доступа к IP-камерам и архиву.

С его помощью вы можете за несколько часов запустить свою собственную облачную систему видеонаблюдения и получить удаленный и безопасный доступ к вашим видеокамерам из любой точки мира через обычный веб-браузер или мобильное приложение.

Также, вы можете установить Flussonic Watcher и внутри локальной сети, сделав закрытую (Private Cloud) систему видеонаблюдения.

## Презентация Flussonic Watcher

Операторам — для запуска собственного облачного сервиса абонентского видеонаблюдения.

Производствам и бизнесам — для аудио-видео фиксации и удаленного контроля за производственными процессами.

Управляющим компаниям и застройщикам — для трансляции видео с социально-значимых объектов, а также для контроля за строительством.

Муниципальным и федеральным властям — для предоставления населению доступа к публичным камерам, для закрытых проектов спец. служб, для программ типа «Безопасный город», «Безопасный регион».

Серверное ПО — база данных пользователей и камер, медиа-сервер (осуществляет захват, транскодирование, запись и мультипротокольную доставку видео на разные устройства), API для интеграции с внешними системами;

Веб-интерфейс — интерфейс для пользователей и админов, работающий в любом браузере. В интерфейсе вы найдете: dashboard с камерами и архивом, избранное, группы, карту, инструменты управления пользователями и камерами, настройки, инструменты брендирования интерфейса и многое другое;

Мобильные приложения для iOS и Android — удобный удаленный доступ к камерам, архиву и другим инструментам;

Прошивка для камер (агент) — устанавливается на камеры, для обеспечения видимости из-за NAT и стабильной зашифрованной доставки видео напрямую от камеры на ваш сервер.

Single — базовая версия. Подходит тем, кому не нужно брендирование интерфейса (свое лого, цвета), для проектов до 500 камер и не более 1 сервера.

Cluster — продвинутая версия. Включает в себя инструменты брендирования (смена лого, цвета, текста), возможность расширения до десятков тысяч камер, включая возможность собрать кластер с резервированием потоков, прошивка для камер. Подходит тем, кто хочет запустить свой собственный абонентский сервис (VSaaS) или большую систему видеонаблюдения.

# Аппаратные требования

Flussonic Watcher работает в двух вариантах:

Single — Подходит тем, кому не нужно брендирование интерфейса (свое лого, цвета), для проектов до 500 камер, не более 1 сервера.

Cluster — Продвинутая версия. Включает в себя инструменты брендирования (смена лого, цвета, текста), возможность расширения до десятков тысяч камер, включая возможность собрать кластер из десятка серверов с резервированием потоков. Для данной конфигурации требуется минимум 2 сервера. Подходит тем, кто делает абонентский сервис или большую систему видеонаблюдения.

Управляющий сервер и он же транслирующий:

Операционная система: Ubuntu 14.04 и выше, Debian 7 и выше;

Железо: Не ниже CPU Xeon E-3 1230v5 3.4 GHz, 32Gb RAM;

База Данных: PostgreSQL;

Диски: Расчет дискового пространства следует произвести из того, что 1 мбит/с камера за сутки пишет 10Гб видео, 2 мбит/с — 20Гб. Например, 1 мбит/с камера с 7-дневным архивом займет 70Гб.

Управляющий сервер (где стоит БД и веб-интерфейс Flussonic Watcher):

Система: Ubuntu 14.04 и выше, Debian 7 и выше;

Железо: виртуальный или «железные» сервер, CPU с 2-я ядрами, 8Gb RAM;

База Данных: PostgreSQL;

Диски: SSD 64Гб.

Транслирующий сервер (серверы, куда заведены потоки с камер и пишется архив):

Система: Ubuntu 14.04 и выше, Debian 7 и выше;

Железо: Не ниже CPU Xeon E-3 1230v5 3.4 GHz, 32Gb RAM;

Диски: Расчет дискового пространства следует произвести из того, что 1 мбит/с камера за сутки пишет 10Гб видео, 2 мбит/с — 20Гб. Например, 1 мбит/с камера с 7-дневным архивом займет 70Гб.

Данные конфигурации серверов приведены как пример, пригодный для 500 камер, с 1 мбит/с потоком, 500 пользователей, с выключенным фейловером захвата и без собранных мозаик.

При увеличении битрейта до 2 мбит/с, кол-

во камер на 1 сервер следует сократить на 2 (до 250). Рекомендации актуальны только с учетом того, что на серверах не будет другого запущенного ПО.

Важно! Для корректной работы Flussonic Watcher требует открытые порты 80, 443, 1935, 554 на всех хостах, а на управляющем сервере должен быть прописан реальный hostname, который резолвится из Интернета.

# Установка Watcher

Flussonic Watcher может работать как в кластерном (многосерверном) режиме, так и в одиночном (односерверном). Процесс установки почти одинаков для обоих режимов.

Watcher поддерживает проекты от малых до крупных, где требуется поддержка неограниченного количества камер. Количество камер ограничивается только возможностями аппаратной части системы.

При работе в кластере из серверов Watcher обеспечивает отказоустойчивость захвата потоков (failover).

Watcher использует СУБД PostgreSQL.

Внимание! Watcher теперь устанавливается из пакета flussonic-watcher и сразу вместе с ним будут установлены Flussonic Media Server и база данных PostgreSQL.

## Отличия режима Watcher cluster от Watcher single

С точки зрения организации видеонаблюдения, Watcher в режиме кластера отличается от Watcher в односерверном режиме следующим:

Watcher Cluster поддерживает инструменты брендирования веб-интерфейса (возможность поставить свой логотип, выбрать свои цвета и т.д.).

Watcher Cluster умеет работать с прошивкой для камер — Flussonic Agent, или просто Агент. С помощью Агента становится возможной видимость камер из-за NAT и значительно упрощается настройка всей инфраструктуры. Агент делает возможным plug-and-play режим при добавлении камеры, а также повышает стабильность доставки видео и реализует шифрование данных напрямую с камер на стримеры (транслирующие серверы). Для подготовки прошивки обратитесь к нашему менеджеру, который ведет ваш проект.

Watcher Single подходит для маленьких и средних проектов, где максимальное количество IP камер не превышает 500.

## Конфигурация серверов

Если ваш проект небольшой и вам не нужен кластер и транслирующие серверы, просто установите Watcher. В односерверном режиме все камеры работают с одним сервером, где установлены Flussonic Watcher, Flussonic Media Server, база данных, работают веб-интерфейсы Flussonic и Watcher, проходят видеопотоки и пишется архив.

Для кластера необходимо минимум два сервера:

Один сервер — управляющий. На нём работает веб-интерфейс Watcher, Flussonic Media Server, бизнес-логика и сервер баз данных PostgreSQL. Watcher работает лишь на одном из серверов — на управляющем сервере.

От 1 до 100 серверов — транслирующие сервера (стримеры), которые контролируются управляющим сервером. На них установлен Flussonic Media Server. На транслирующих серверах хранятся DVR архивы, и через эти серверы проходят потоки с камер.

Все серверы должны иметь публичные IP адреса и одинаковый кластерный ключ (указывается в настройках Flussonic). Кроме того, имя хоста управляющего сервера должно резолвиться в IP адрес.

Если ваш проект небольшой и вам не нужны стримеры, просто установите Watcher на один сервер.

На схеме показано, из чего состоит кластер и как проходят видеопотоки:

## Порядок установки

Чтобы установить Watcher, нужно:

Установить Flussonic Watcher на управляющий сервер. Watcher устанавливается отдельным пакетом, и с ним будут установлены необходимые для его работы PostgreSQL и Flussonic Media Server.

В веб-интерфейсе Flussonic Media Server указать путь до PostgreSQL.

Открыть веб-интерфейс Watcher и завести администратора Watcher.

Этого достаточно для установки в односерверном режиме (single).

Чтобы создать кластер, после выполнения шагов 1-3 нужно продолжить установку:

Установить Flussonic Media Server на все транслирующие серверы (стримеры).

В администраторском интерфейсе Watcher завести стримеры.

Все шаги описаны ниже.

Про обновление Watcher рассказано в разделе Обновление Flussonic Watcher.

## Установка Flussonic Watcher

На сервере, где вы планируете установить Watcher, выполните команду:

```
curl -sSf https://flussonic.com/raw/install_watcher.sh | sh
```

После успешной установки PostgreSQL система предложит вам запустить PostgreSQL и выведет команду для запуска. Не нужно это выполнять, а нужно перейти к следующему шагу - создание пользователя и базы.

Создайте пользователя и базу данных. Для этого выполните по порядку эти команды в консоли. Создайте пользователя `vsaas` в PostgreSQL:

```
sudo -u postgres -i createuser -P vsaas
```

Система предложит ввести пароль, который будет у пользователя `vsaas`:

Enter password for new role: (придумайте и введите пароль супер-админа базы данных)

После того, как вы введете пароль, нужно сделать это ещё раз:

Enter it again: (повторите пароль супер-админа)

Создать базу данных `vsaas_production` с владельцем-созданным пользователем `vsaas`:

```
sudo -u postgres -i createdb -O vsaas -e -E UTF8 -T template0 vsaas_production
```

Ответ системы в случае успешного создания базы данных:

```
CREATE DATABASE vsaas_production OWNER vsaas ENCODING 'UTF8' TEMPLATE template0;
```

Затем в административном интерфейсе Flussonic (<http://flussonic:8080/admin>) перейдите в раздел IP cameras и в поле Database path укажите путь к базе данных.

Важно. Замените `VSAAS_PASSWORD` на пароль, который вы указали при создании пользователя `vsaas` в командной строке.

Только для кластера: В конфигурационном файле `/etc/flussonic/flussonic.conf` автоматически будет добавлена директива `vsaas`. Добавьте к ней опцию `mode cluster`, открыв файл на редактирование:

```
vsaas {  
  database postgresql://vsaas:vsaas_password@localhost/vsaas_production;  
  endpoint enabled;  
  mode cluster;  
}
```

После внесенных в файл изменений перезапустите Flussonic Media Server командой в командной строке:

```
service flussonic restart
```

Теперь в браузере обновите страницу с веб-интерфейсом Flussonic.

После настройки Watcher по адресу `http://FLUSSONIC:8080` будет открываться веб-интерфейс Watcher вместо веб-интерфейса Flussonic Media Server.

Для того, чтобы вернуться в веб-интерфейс Flussonic Media Server, перейдите по адресу `http://FLUSSONIC:8080/admin`.

Далее необходимо создать главного администратора в веб-интерфейсе Watcher.

## Создание администратора Watcher

При первом запуске Flussonic Watcher (`http://FLUSSONIC:8080`) в веб-интерфейсе появится диалог создания первого администратора. На этом экране необходимо придумать и ввести логин и пароль будущего администратора системы видеонаблюдения.

На этом установка для односерверного режима работы завершена.

Чтобы создать кластер, необходимо подготовить стримеры и настроить Watcher для работы в кластере (см. следующие шаги).

## (Только для кластера) Установка Flussonic Media Server на стримеры

В случае кластера необходимо установить Flussonic Media Server на всех транслирующих серверах.

Выполните команду:

```
curl -sSf https://flussonic.com/raw/install.sh | sh
```

Затем запустите Flussonic Media Server:

```
/etc/init.d/flussonic start
```

Подробно установка Flussonic Media Server описана в документации Flussonic:

Быстрый старт с Flussonic Media Server — кратко о том, как установить Flussonic и начать работу.

Установка Flussonic Media Server — подробное описание установки и системные требования.  
Далее на каждом стримере нужно настроить HTTPS и добавить кластерный ключ в настройки Flussonic — см. Создание кластера ниже.

## (Только для кластера) Создание кластера (многосерверного режима)

Создание кластера заключается в добавлении стримеров (т.е. транслирующих серверов) в настройках Flussonic Watcher. Нужно добавить хотя бы один стример, на котором заведены IP камеры. Это позволит начать принимать видео с камер в кластерном режиме.

Страница Watcher UI Настройки > Стримеры отвечает за создание кластера.

Предварительные действия

Для каждого стримера необходимо установить Flussonic Media Server на отдельный сервер, который станет стримером. То есть помимо сервера, на котором вы установили Flussonic Watcher, должен быть "поднят" еще один сервер с внешним (публичным) IP-адресом.

После установки Flussonic на стример немедленно смените логин и пароль администратора на каждом стримере.

Настройте HTTPS на каждом стримере. Достаточно указать порт для протокола HTTPS, и Flussonic станет использовать самоподписанные SSL сертификаты.

Установите одинаковое время на управляющем сервере и на каждом стримере.

В настройках каждого стримера на управляющем сервере пропишите `cluster_key` (он должен совпадать с `cluster_key` Flussonic Watcher). Подробнее о `cluster_key`.

Настройте DNS зону для управляющего сервера.

Для корректной работы Watcher в кластере необходимо внести в настройки DNS зоны для вашего домена A-запись, где указать имя хоста. Это имя хоста также нужно прописать и в операционной системе на сервере с Flussonic Watcher. Это необходимо, чтобы стримеры могли обращаться к управляющему серверу.

Для проверки выполните на управляющем сервере команду `hostname` — она должна вернуть правильное имя хоста, указанное в настройках DNS, например, `example.com`.

Когда сервер, выделенный под передачу видео с камер, готов к работе, нужно добавить его в настройках Flussonic Watcher.

Добавление стримеров в Watcher

Зайдите в Flussonic Watcher под администратором.

Перейдите в Настройки > Стримеры и кликните по значку "+" для того, чтобы ввести данные стримера:

Хост – доменное имя стримера. Пример: `streamer2.example.com`

Ключ кластера – кластерный ключ (`cluster_key` в файле настроек). Если кластерный ключ совпадает у Flussonic Watcher и стримера, то заполнять данное поле необязательно.

DVR путь – путь к архиву, например: `/dvr`, или множество путей (через пробел): `/dvr1 /dvr2`.

Обязательно заполните это поле, иначе архив не будет работать.

Хост – доменное имя стримера. Пример: streamer2.example.com

Ключ кластера – кластерный ключ (cluster\_key в файле настроек). Если кластерный ключ совпадает у Flussonic Watcher и стримера, то заполнять данное поле необязательно.

DVR путь – путь к архиву, например: /dvr, или множество путей (через пробел): /dvr1 /dvr2. Обязательно заполните это поле, иначе архив не будет работать.

Если вы добавили несколько стримеров, вам необходимо выбрать основной, нажав на Default в карточке созданного стримера. Все новые камеры будут автоматически добавляться на основной стример.

Важно!

В разделе Стримеры не нужно добавлять хост, на котором развернут сам Flussonic Watcher. На всех серверах должно стоять одинаковое время.

Для каждого стримера вы можете включить автоматическое использование запасных серверов для захвата потоков в случае отказа стримера (см. Фейловер захвата).

## Обновление Flussonic Watcher

Чтобы обновить Watcher:

```
apt-get update  
apt-get -y install flussonic-watcher  
/etc/init.d/flussonic restart
```

При своём обновлении Watcher автоматически производит миграцию базы данных для работы с новой версией. В редких случаях может потребоваться ручная миграция, тогда Watcher сообщит об этом в веб-интерфейсе.

Мы настоятельно рекомендуем делать резервную копию базы данных ежедневно и перед обновлениями.

# База данных Watcher

В этом разделе приведены инструкции по решению отдельных задач поддержки базы данных, используемой в Watcher.

Миграция на PostgreSQL (необходима начиная с версии 19.03)

Обновление структуры базы данных вручную (может потребоваться в некоторых случаях, Watcher сообщит об этом)

## Миграция с SQLite на PostgreSQL

Сделайте бэкапы этих файлов:

```
/etc/flussonic/flussonic.conf
```

```
/opt/flussonic/priv/vsaas.db
```

Установите последнюю актуальную версию Flussonic Watcher (для этого достаточно обновить Flussonic Media Server). Выполните в командной строке две команды:

```
apt update
```

и затем

```
apt install flussonic
```

Подробнее о процессе обновления

Сделайте бэкап всех данных с помощью нашей встроенной утилиты для миграции:

```
cd /opt/flussonic/apps/vsaas
```

```
/opt/flussonic/bin/python -m manage backup create
```

В результате выполнения получится примерно такой файл:

```
/opt/flussonic/apps/vsaas/backups/20190215201434-b62d21842ab7-WatcherBackup.gz
```

Установите PostgreSQL

Замените строку database в файле конфигурации flussonic.conf:

```
database postgresql://vsaas:vsaas@localhost/vsaas_production;
```

Для редактирования файла можно воспользоваться редактором nano.

Перезагрузите сервис Flussonic:

```
/etc/init.d/flussonic restart
```

Восстановите данные из файла, созданного утилитой для миграции:

```
cd /opt/flussonic/apps/vsaas
```

```
/opt/flussonic/bin/python -m manage backup restore -d 20190215201434
```

## Установка и настройка PostgreSQL

Для установки PostgreSQL необходимо в консоли под пользователем root выполнить команду:  
apt install postgresql

Создайте пользователя и базу данных. Для этого выполните по порядку эти команды в консоли. Сначала создадим пользователя с именем vsaas:

```
sudo -u postgres createuser -P vsaas
```

Система предложит ввести пароль, который будет у пользователя vsaas:

Enter password for new role: (придумайте и введите пароль супер-админа)

После того, как вы введете пароль, нужно сделать это ещё раз:

Enter it again: (повторите пароль супер-админа)

Создайте базу данных vsaas\_production с владельцем-созданным пользователем vsaas:

```
sudo -u postgres createdb -O vsaas -e -E UTF8 -T template0 vsaas_production
```

Ответ системы в случае успешного создания базы данных:

```
CREATE DATABASE vsaas_production OWNER vsaas ENCODING 'UTF8' TEMPLATE template0
```

Затем в административном интерфейсе Flussonic Media Server (см. Watcher UI и Flussonic UI ниже) перейдите в раздел IP cameras и в поле Database path укажите путь к базе данных.

Важно. Замените VSAAS\_PASSWORD на пароль, который вы указали при создании пользователя vsaas.

## Обновление структуры БД

Иногда новая версия Flussonic Watcher требует для работы изменений в структуре своей базы данных. В этом случае web-интерфейс покажет сообщение:

Изменения в структуру базы вносит скрипт watcher\_db\_migrate.sh:

```
/opt/flussonic/contrib/watcher_db_migrate.sh
```

Его необходимо запустить на сервере вручную.

Пример успешного выполнения скрипта:

```
# /opt/flussonic/contrib/watcher_db_migrate.sh
```

```
INFO [alembic.runtime.migration] Context impl PostgresqlImpl.
```

```
INFO [alembic.runtime.migration] Will assume non-transactional DDL.
```

```
INFO [alembic.runtime.migration] Running upgrade dfd74e510414 -
```

```
> 1822b8f25e20, agent:model, agent:camera
```

```
INFO [alembic.runtime.migration] Running upgrade 1822b8f25e20 -> 1a71a9477bbb, streamer: cluster_key
```

```
INFO [alembic.runtime.migration] Running upgrade 1a71a9477bbb -
```

```
> 7a3ab2550cab, streamer_fkey cascade
```

## Restarting Watcher

После этого интерфейс Flussonic Watcher будет снова доступен для работы.

Если в ходе выполнения `watcher_db_migrate.sh` появится ошибка

```
alembic.util.exc.CommandError: Can't locate revision identified by 'ebdce5515b6d'
```

то необходимо установить предыдущую используемую вами версию (до даунгрейда либо обновления Watcher) и выполнить:

```
cd /opt/flussonic/apps/vsaas  
/opt/flussonic/bin/python -m manage db downgrade
```

Важно! Перед запуском скрипта `watcher_db_migrate.sh` необходимо сделать резервную копию базы.

Место расположения базы данных отражено в конфигурации Flussonic Media Server:  
`/etc/flussonic/flussonic.conf`.

```
vsaas {  
    database postgresql://vsaas:VSAAS_PASSWORD@localhost/vsaas_production;  
}
```

Создайте папку, куда будет помещена резервная копия базы (`/PATH/TO/BACKUP/` в примере ниже).

Выполните резервное копирование с помощью утилиты `pg_dump`:

```
pg_dump -h localhost -U vsaas -d vsaas_production > /PATH/TO/BACKUP/vsaas.sql
```

Ответ системы: Password:

Введите пароль `VSAAS_PASSWORD`, который вы указывали при создании пользователя `vsaas`.

# Настройки

Для администратора доступны следующие настройки Flussonic Watcher.

Режим работы — режим работы: многосерверный кластер или запуск плагином на одном сервере.

Стримеры — становится активным в кластерном режиме.

Archive path — путь к хранилищу DVR. Может быть Swift-хранилищем, доступно только в standalone режиме с запуском на одном сервере. В кластерном режиме путь для DVR задается в настройках стримера на управляющем сервере.

API ключ — токен использующийся для мобильного доступа. Необходимо использовать в личном кабинете на сайте Flussonic для активации доступа к мобильным приложениям.

Доступ для гостей — разрешать ли доступ незалогиненным пользователям.

Самостоятельная регистрация пользователей — включить возможность самостоятельной регистрации пользователей.

Главная страницы — какую страницу показывать автоизованным пользователям: карту или дашборд.

Главная страница для гостей — какую страницу показывать гостям: карту или дашборд.

Тип панели по умолчанию — какую страницу показывать на дашборде: камеры или группы.

External Authentication — внешняя авторизация. Указывается http адрес или адрес RADIUS сервера.

Показывать карту в гл. меню — можно добавить или убрать карту из бокового меню.

Центр карты — по каким координатам будет отцентрирована карта.

Язык карты — язык карты.

Движок карт — выбор поставщика карт: Google Maps, OpenStreetMaps или OpenStreetMaps Offline.

Управление серийными номерами — управление серийными номерами камер в Flussonic Watcher. Нужно в случае, когда вы используете агента.

Дополнительный текст в заголовке — подробнее в разделе «Кастомизация интерфейса»

Свой заголовок страницы — подробнее в разделе «Кастомизация интерфейса»

Свой логотип — подробнее в разделе «Кастомизация интерфейса»

Свой логотип для страницы входа — подробнее в разделе «Кастомизация интерфейса»

Своя цветовая схема — подробнее в разделе «Кастомизация интерфейса»

# Сброс пароля

Восстановление забытого пароля возможно с помощью отправки специального письма на почтовый адрес пользователя. Письмо содержит ссылку для восстановления пароля.

Чтобы этот механизм заработал, в Flussonic Watcher должны быть настроен SMTP сервер для отправки писем.

Также пароль можно поменять с помощью специальной утилиты, которая входит в поставку Flussonic Media Server:

```
/opt/flussonic/contrib/reset_password email.address@example.com pwd123xyz
```

Первым аргументом укажите e-mail, вторым — новый пароль.

В ответ программа должна вывести на экран:

```
Changing password for email.address@example.com
```

# Фейловер захвата

Функция фейловер захвата разработана, чтобы при падении одного из стримеров камеры (стримы) равномерно распределялись по работающим серверам кластера.

Доступ к архиву на упавшем сервере пропадает, начинается запись нового архива (на донорском сервере).

При восстановлении работы основного сервера камеры автоматически возвращаются обратно, возобновляя доступ к старому архиву (если он не был поврежден), но теряя архив, записанный на резервном сервере.

Для включения фейловера, необходимо нажать на соответствующую кнопку в карточке каждого стримера, в разделе «Стримеры» и поменять режим работы в файле `/etc/flussonic/flussonic.conf` на сервере Watcher:

```
vsaas {  
    mode cluster+failover=30;  
    ...  
}
```

Где 30 — частота проверки стримеров в секундах. После изменения настройки нужно перезапустить Watcher:

```
/etc/init.d/flussonic restart
```

# Прием событий о движении с камеры

## Прием событий о движении с камеры

Flussonic Watcher умеет принимать события по email, для этого у него есть специальный SMTP сервер. С камеры приходят email сообщения о событиях движения, и система добавляет пометки в архивное видео в тех местах, когда было обнаружено движение.

Настройка состоит из двух этапов:

Настройка приема писем в Flussonic Watcher

Настройка камеры на отправку писем через SMTP сервер Flussonic Watcher.

Как работает запись по движению:

Flussonic Watcher ведет постоянную запись с указанной глубиной записи как обычно, но когда прилетает email с движением, он отмечает этот интервал у себя в базе данных, чтобы отразить метки в архивном плеере и, что самое важное, защищает эту запись от удаления. Длительность защищенной записи определяется двумя временными метками, первая из которых рассчитывается как текущее время минус 10 сек, а вторая метка - текущее время плюс 30 сек.

Это значит, что вам достаточно установить глубину архива: например, 6 часов. И затем включить прием событий. В итоге вы получите 6 часов непрерывного архива + записи по датчику будут храниться до тех пор, пока на диске есть свободное место. Запись новых событий будет удалять старые.

Рассчитав необходимый объем диска исходя из битрейта камер и частоты движений, можно сэкономить до 50-90% дискового пространства по сравнению с обычной записью без событий.

## Включаем прием эвентов в Watcher

Для включения захвата событий о движении с камеры необходимо открыть конфигурационный файл `/etc/flussonic/flussonic.conf` и прописать `camera_alarm` плагин:

```
plugin camera_alarm {
  catch motion;
  listen smtp://0.0.0.0:1025;
}
```

Параметр `catch` определяет слово, которое Flussonic Watcher будет искать в теме письма.

Большинство камер по умолчанию присылают письма примерно с таким заголовком: "Camera 123 Motion Detected at 14:21 27-10-2017".

Если ваша камера отправляет письмо с другой темой или имеет возможность задать свой, то настройте `catch` по своему усмотрению.

Параметр `listen` определяет интерфейс и порт для встроенного SMTP сервера. Тут же можно указать логин и пароль: `listen smtp://username:password@0.0.0.0:1025;`

Перезапустите Flussonic Media Server.

На странице управления камерами появится ссылка на список событий.

## Настройка камеры

Настройка камеры сводится к указанию адреса SMTP сервера, имени отправителя и получателя.

Важно. Укажите имя отправителя и получателя в виде полного имени камеры (имя камеры и ее ID), например: cam1-abcdefg@example.com. Если вы назвали камеру cam1, то Flussonic Watcher присвоит ей ID и получится полное имя, например cam1-abcdefg.

Полное имя камеры можно найти в интерфейсе Flussonic Watcher.

И в интерфейсе архива камеры будут появляться отметки о движении.

# События распознавания автомобильных номеров

Flussonic умеет находить и распознавать номера российских автомобилей, попавших в кадр IP камеры, включая спецтранспорт (пожарные и скорые). Эта функциональность известна как ANPR (automatic number plate recognition).

Flussonic делает следующее:

- Создает события детекции номеров.

- Видео приходит с IP камер на транслирующий сервер (в кластерной установке) или на управляющий сервер (в обычной установке), на котором и происходит распознавание номеров.

- Предоставляет Watcher UI для просмотра событий распознавания номеров.

- В новом веб-интерфейсе к Watcher, начиная с версии 18.11, можно посмотреть зарегистрированные события и видео с места событий.

- Предоставляет API для выгрузки событий во внешние системы.

Чтобы начать распознавать автомобильные номера:

- Подготовьте аппаратную и программную часть на сервере Flussonic, который будет распознавать номера.

- Включите и настройте распознавание номеров. Настроить можно как через файл, так и в веб-интерфейсе Watcher, но некоторые параметры сейчас можно указать только в файле.

Содержание:

- Установка модуля распознавания номеров

- Настройка в конфигурационном файле

- Настройка в веб-интерфейсе

- Просмотр событий распознавания номеров

- API модуля распознавания номеров

## Установка модуля распознавания номеров

Чтобы распознавать номера, нужно установить Flussonic Media Server и Watcher (если вы ещё не работали с ними). Функция распознавания работает как в одиночной установке Flussonic вместе с Watcher, так и в кластере.

Распознавание номеров происходит на транслирующем сервере (в кластерной установке) или на управляющем сервере (в обычной установке). К этому серверу нужно подключить камеру, с которой вы хотите распознавать номера. Необходимо, чтобы сервер имел минимум одну производительную видеокарту NVIDIA GPU, у которой не менее 6 Гб видеопамяти.

## Требования к системе, на которой будет работать ANPR

ОС: Ubuntu 16.04 x64

GPU: Nvidia (Pascal) от 6 Гб VRAM (более точные рекомендации мы можем дать только для конкретного проекта).

Процессор: не менее 4-х ядер.

Память: не менее 8 Гб ОЗУ.

Flussonic Media Server (для стримера в кластерной установке)

Flussonic Media Server + Watcher (в одиночной установке)

Важно. В кластерной установке система распознавания работает на транслирующем сервере (стримере), где уже должен быть установлен Flussonic Media Server. В одиночной установке система распознавания требует установленного Flussonic Watcher вместе с Flussonic Media Server.

## Установка

Для установки модуля распознавания номеров подключите официальный репозиторий Nvidia и установите пакет flussonic-vision из нашего репозитория:

```
wget http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/cuda-repo-ubuntu1604_9.2.148-1_amd64.deb
dpkg -i cuda-repo-ubuntu1604_9.2.148-1_amd64.deb
apt-key adv --fetch-keys http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/7fa2af80.pub
apt update
apt install flussonic-vision
```

## Настройка распознавания номеров через конфигурационный файл

Замечание. Всё то же самое можно сделать через UI — настройки сами запишутся в файл. Но вам потребуется проверить номер GPU в конфигурационном файле и при необходимости отредактировать его вручную.

Откройте файл /etc/flussonic/flussonic.conf.

Для включения системы распознавания добавьте строку с плагином plugin vision;

Добавьте директиву vision в конфигурацию потока, указав номер GPU:

```
stream cam1 {
  url rtsp://192.168.0.11:554/h264;
  vision gpu=0;
}
```

```
stream cam2 {
  url rtsp://192.168.0.12:554/h264;
  vision gpu=1;
}
```

plugin vision;

gpu (обязательная опция) – номер видеокарты, его можно получить с помощью утилиты nvidia-smi.

По умолчанию система распознавания ищет автомобильные номера по всей области кадра.

## Настройка распознавания номеров через UI

Прежде чем настраивать камеры в UI, убедитесь, что система распознавания включена. Для этого в файл `/etc/flussonic/flussonic.conf` добавьте строку:

```
plugin vision;
```

Теперь включите детекцию и распознавание номеров для камеры:

В Watcher UI перейдите в раздел `Cameras`. Найдите камеру в списке камер и откройте ее настройки, щелкнув в правом верхнем углу плеера значок настроек.

Выберите опцию ANPR и нажмите `Save`.

Теперь система распознавания Flussonic будет определять номера в кадре этой камеры и фиксировать время, когда номер появился в кадре и когда он покинул кадр.

Flussonic добавит опции потока в конфигурационный файл `/etc/flussonic/flussonic.conf`.

Вам потребуется вручную отредактировать номер GPU (см. предыдущий раздел о настройке через файл).

## Просмотр событий распознавания номеров в UI

Flussonic создает события двух видов:

`enter` – номер появился в кадре (в поле видимости камеры)

`leave` – номер покинул кадр.

Чтобы посмотреть, какие номера появлялись в кадре камеры:

В Watcher UI перейдите в раздел `Notifications`. Откроется список всех событий.

Используйте фильтры и поиск справа, чтобы найти камеру:

В поле `Source` выберите `Plate detector`.

В полях `From` и `To` выберите дату и время начала и окончания интервала времени, за который хотите посмотреть событие.

В поле `Search` введите номер машины.

Чтобы очистить введенные параметры поиска, нажмите `Clear Filter`.

В поле `Source` выберите `Plate detector`.

В полях `From` и `To` выберите дату и время начала и окончания интервала времени, за который хотите посмотреть событие.

В поле `Search` введите номер машины.

Чтобы очистить введенные параметры поиска, нажмите `Clear Filter`.

Список событий фильтруется сам по мере ввода вами критериев поиска.

Чтобы просмотреть запись с камеры, щелкните в строке, содержащей данное событие. Внизу экрана откроется плеер и проиграт запись. Чтобы увеличить изображение, нужно сделать двойной щелчок по изображению в плеере.

## API модуля распознавания номеров

Если система правильно настроена, то вы сможете получить данные о событиях в формате JSON через Watcher API. Подробная документация по API доступна по ссылке.

Ниже пример запроса и ответа:

```
root@ubuntu:~# curl localhost/vsaas/api/v2/events?type=activity -H 'x-vsaas-api-key: dfb21d1f-3e00-44a2-a706-36d99f9e9d73'
```

```
{
  "start_at": 1538645882872,
  "type": "activity",
  "id": "7ecb0a13-414c-462f-a206-3c5d047baad4",
  "ext_data": null,
  "object_id": "A123AA 77",
  "end_at": null,
  "source": "plate_detector",
  "camera_id": "cam0-00",
  "source_id": "0",
  "object_class": "leave",
  "event_data": null
}
```

start\_at - время события

id - уникальный ID записи

object\_id - номер автомобиля

camera\_id - имя камеры в Watcher

object\_class - событие, может быть enter и leave (машина появилась или покинула наблюдаемую зону).

# Auto-login

У Flussonic Watcher есть возможность входа пользователя по специальной ссылке без ввода пароля. Это может быть необходимо, если вы хотите упростить вход в Flussonic Watcher для своих пользователей или предотвратить передачу паролей третьим лицам. Предполагается, что ссылка для автологина выдается уже авторизованному клиенту.

Для того чтобы сгенерировать URL адрес для автологина, нужно предварительно запросить у Flussonic Watcher токен по ссылке `/vsaas/api/generate-autologin-token`. Затем пользователя можно авторизировать через POST на `/vsaas/autologin`.

Шаги:

1) Предварительно сгенерируем админский токен. Для этого нам понадобится API KEY со страницы настроек административного интерфейса Flussonic Watcher.

```
sign = md5(salt + ":" + utc + ":" + api_key)
admin_token = salt + ":" + utc + ":" + sign
```

где:

`salt` — любая случайная строка;

`utc` — текущее UTC время в секундах;

`api_key` — ключ API со страницы настроек админки Flussonic Watcher.

Сгенерированный токен имеет конечное время жизни и привязывается к указанному времени UTC.

Например, если

```
salt = "20a666"
utc = "1487258700"
api_key = "HELLO"
```

тогда итоговый `admin_token` будет: `"20a666:1487258700:4b60f36de708e5b3472155db2fea990a"`

2) Итак, админский токен готов, теперь нужно получить токен для автологина конкретного пользователя. Для этого нужно сделать POST запрос в JSON-формате:

```
curl --header "X-Vsaas-Api-Key: ADMIN_TOKEN" --header "Content-Type: application/json" --request POST --data '{"login": LOGIN, "valid_till": VALID_TILL, "lifetime": LIFETIME}' "http://watcher.com/vsaas/api/generate-autologin-token"
```

`ADMIN_TOKEN` — токен, сгенерированный в шаге 1. Этот токен передается в HTTP-заголовке `X-Vsaas-Api-Key`.

`LOGIN` — login (он же email) пользователя, которому нужно дать доступ. Строка. Обязательный параметр.

`VALID_TILL` — время UTC в секундах, до которого этот токен будет действителен для автологина. Целое число. Необязательный параметр.

`LIFETIME` — длительность сессии, открываемой через автологин в секундах. Целое число. Необязательный параметр.

Ответ будет в JSON формате:

```
{
```

```
"autologin_token": "demo:1487258314:f8b1:b4bdaac58cbe94638e5b14a3728b8e6d633f3c6e",  
"success": true  
}
```

Нам нужен этот autologin\_token.

3) Токен autologin\_token, полученный на шаге 2, используется для POST-запроса в Flussonic Watcher. Например, таким образом:

При нажатии на кнопку подтверждения пользователь автоматически попадает в интерфейс Flussonic Watcher и оказывается залогинен.

# Пользовательский интерфейс

После добавления пользователя в систему, он может зайти в интерфейс Flussonic Watcher под своим e-mail и паролем.

После этого он сможет посмотреть доступные камеры, которые ему разрешил смотреть администратор: публичные, только для авторизованных и приватные.

Пользователю доступны следующие меню:

Дашбоард «Мои камеры».

Карта.

Избранное.

Камеры.

Пользователи и группы. (Для Администратора)

Настройки. (Для Администратора)

## Дашбоард «Мои камеры»

Дашбоард «Мои камеры» — основная страница доступа к камерам, которые доступны пользователю.

Доступны фильтры по доступу:

Любой.

Публичный.

Для клиентов.

Частный.

Режимы отображения камер:

Средняя сетка.

Крупная сетка.

Список.

Также можно включить Компактный режим и Скрыть отключенные камеры.

В этом списке показываются скриншоты с камер, которые периодически обновляются. Чтобы просмотреть какую-либо камеру или посмотреть архив, достаточно кликнуть на неё или на кнопку Play.

Нажав на название камеры, можно открыть ее на просмотр в большом окне или перейти на отдельную страницу просмотра архива:

Мозаика — это возможность собрать на отдельной странице до 8 любых камер по выбору пользователя. Чтобы пользователю были доступны мозаики, их необходимо создать в разделе Камеры.

## Карта

На карте показаны камеры, для которых в настройках указаны координаты.

## Избранное

В «Избранное» добавляются камеры, которые пользователь отметил «звёздочкой».

## Камеры

В меню «Камеры» производятся все операции с камерами:

- Добавление камеры.

- Поиск камеры по протоколу ONVIF.

- Импорт из Flussonic.

Также есть возможности:

- Экспорт списка файлов в формате CSV.

- Просмотр списка серийных номеров камер.

- Создание мозаики.

Подробнее про добавление камеры в разделе «Управление камерами».

## Пользователи и группы

В этом меню производится добавление и изменение пользователей Flussonic Watcher.

Подробнее в разделе «Управление пользователями»

С помощью групп вы сможете упростить навигацию по камерам и пользователям, делая логическую группировку (например, по этажам, отделам, регионам, микрорайонам и т.д.), а также облегчить процесс распределения прав видимости камер. Подробнее в разделе «Группы»

## Настройки

В этом меню производится настройка Flussonic Watcher. Подробнее в разделе «Настройки»

# Мобильные приложения

Watcher предлагает мобильные приложения для доступа к системе видеонаблюдения в реальном времени.

Главные возможности Watcher:

- Просмотр видео в реальном времени с IP-камер со сверхнизкой задержкой

- Просмотр архива без ограничений по его глубине

- Контроль доступа на основе отпечатка пальца или PIN-кода

- TLS-шифрование видео потоков

- Push-уведомления о событиях

- Загрузка видео скриншотов.

Для того, чтобы мобильные приложения знали с какого сервера забирать видео, необходимо получить Оператор ID, куда будет зашит адрес вашего Flussonic Watcher.

По умолчанию, приложение обращается к серверу Flussonic, поэтому без создания личного Оператор ID авторизация в мобильном приложении работать не будет.

Для получения Оператор ID необходимо залогиниться в личном кабинете, кликнуть на ключ инсталляции на главном экране ЛК и в поле Watcher URL указать публичный URL вашего Flussonic Watcher в виде: `http://APIKEY@watcher-host`.

Адрес Flussonic Watcher должен быть виден из интернета и, крайне желательно, быть в виде доменного имени, а не IP-адреса.

Для использования мобильного приложения вам необходимо скачать его в Apple Store или Google Play, а далее авторизоваться под созданным в вашем Watcher логином/паролем и полученным в `manage.erlyvideo.org` Оператор ID.

APIKEY — идентификационный ключ, указан в интерфейсе настроек Flussonic Watcher.

Watcher-Hostname — публично видимый путь к Flussonic Watcher.

Важно! Для работы мобильного приложения необходимо, чтобы ваш сервер был виден из сети интернет и обладал фиксированным белым IP-адресом.

Если ваш сервер стоит за NAT или защищен Firewall, наши специалисты помогут выполнить необходимые настройки в рамках расширенной технической поддержки.

# Кастомизация интерфейса

В

Watcher Cluster

есть инструментарий для кастомизации интерфейса. Для этого найдите в разделе

Настройки

:

Дополнительный текст в заголовке. Добавление произвольного текста в заголовок интерфейса.

Свой заголовок страницы. Изменение заголовка.

Свой логотип. Установка логотипа в формате PNG. Для корректного отображения логотип рекомендуется загружать вариант в белом/сером цвете на прозрачном фоне, высота — не более 50 пикселей.

Свой логотип для страницы входа.

Своя цветовая схема. Изменение цветовой схемы. Есть четыре готовый темы и возможность самостоятельного редактирования CSS. Свои CSS-правила следует сохранить в файле:

`/vsaas/static/css/user.css`. Файл `user.css.example` может быть использован в качестве примера.

Также вы можете брендировать email-шаблон восстановления пароля. Подробнее в статье [Брендирование email](#).

# Брендинг email

## Настройка SMTP сервера

Для настройки почты и отправки писем восстановления паролей в конфигурации необходимо указать параметры сервера исходящих сообщений.

Пример. Используем параметры:

Почтовый адрес: email.address@example.com;

Пароль: xyz123;

Адрес почтового сервера: smtp.example.com;

Порт: 465;

Защита соединения: SSL

Пример конфигурации:

```
vsaas {  
  database postgresql://vsaas:PASSWORD@localhost/vsaas_production;  
  smtp_server smtp.example.com:465;  
  smtp_login email.address:xyz123;  
  email_from "Flussonic Watcher ";  
  smtp_opts ssl;  
}
```

## Смена шаблона

В рамках опции брендинга Flussonic Watcher есть возможность поменять email-шаблон восстановления пароля. Для того чтобы использовать свой собственный шаблон письма сброса пароля, либо HTML-шаблон необходимо сделать следующее:

Зайти в директорию /opt/flussonic/apps/vsaas/watcher/templates.

Создать свои собственные шаблоны писем. Можно использовать файлы password\_reset\_request.email и password\_changed.email, как пример.

Сохранить свои шаблоны используя приставку custom\_. Таким образом на сервере должны появиться два файла custom\_password\_reset\_request.email и custom\_password\_changed.email.

Для того, чтобы использовать ещё и HTML-шаблоны, нужно в той же директории создать ещё два файла по такому же принципу: custom\_password\_reset\_request.email.html и custom\_password\_changed.email.html.

Шаблон делится на две части:

Заголовок и непосредственно тело шаблона.

В заголовке можно указывать тему письма.

Кроме того, в теле можно использовать две переменных:

```
{{data.base_url}}
```

```
{{data.token}}
```

Примеры шаблонов:

custom\_password\_reset\_request.email:

---  
subject: "Это тема письма"

---  
А это тело письма, вот ссылка: `{{data.base_url}}/vsaas/forgot-password/{{data.token}}`

erlang  
custom\_password\_reset\_request.email.html:

---  
subject: "Это тема письма"

---  
Тело письма, ссылка

# Watcher API

С помощью Watcher API вы сможете интегрировать сервис видеонаблюдения в вашу экосистему или сделать свои мобильные приложения.

Для Flussonic Watcher разработан API, с помощью которого можно импортировать и экспортировать пользователей и камеры, включая массовый импорт-экспорт. API имеет развитый инструментарий интеграции с биллингом, позволяющий управлять доступностью камер, частных архивов и отдельных пользователей из биллинга, в том числе менять пароли, собирать статусы камер и решать прочие задачи, доступные в интерфейсе.

Также, API позволяет настроить авторизацию через RADIUS-сервер или использовать авторизационный бэкэнд.

Импорт пользователей по API    Импорт камер по API    Интеграция с биллингом    Смена пароля    Бэкэнд для авторизации пользователей    RADIUS

# Импорт пользователей по API

В Flussonic Watcher есть API, которое позволяет разом заливать большое количество уже созданных пользователей и их настройки из внешней системы, базы данных или таблицы. Запросы к API должны быть авторизованы с помощью API-ключа. Для этого вы должны отправить API-ключ в http заголовке x-vsaaas-api-key.

Вы можете найти свой APIKEY в разделе Настройки Watcher-a.

Для импорта списка пользователей надо сформировать CSV с полями и отправить на `http(s)://URL_ВАШЕГО_ВОТЧЕРА/vsaas/api/import/users`:

email — E-mail используется для идентификации пользователей и восстановления пароля

password — пароль открытым текстом

is\_active 1: активный пользователь, 0: пользователь заблокирован

is\_admin 1: пользователь является администратором, 0 - обычный пользователь

groups — список групп, к которым принадлежит пользователь. Разделитель - ";"

note — комментарий к пользователю

Импорт через команду в консоли:

```
curl --data-binary @mydata.csv -H 'Content-type:text/csv' -H 'x-vsaaas-api-key: API_KEY_ВАШЕГО_ВОТЧЕРА' http://WATCHER-HOSTNAME/vsaas/api/import/users
```

Пример файла mydata.csv:

```
email,password,is_active,is_admin,note,groups
ivanov@domain.tld,CergitMig,1,0,user1,grp1;grp2
petrov@domain.tld,LajQuolOy,0,1,user2,grp2
```

Пример одной строкой:

```
echo -
e "email,password,is_active,is_admin,note,groups\nivanov@domain.tld,CergitMig,1,0,user1,grp1;grp2\npetrov@domain.tld,LajQuolOy,0,1,user2,grp2" | curl http://127.0.0.1:8080/vsaas/api/import/users --data-binary @- -H 'Content-type:text/csv' -H 'x-vsaaas-api-key: 3a7d9386-6c3a-440d-a75d-e6b3fdc3368e'
```

Ответ:

```
{"success": true}
```

# Импорт камер по API

```
dd { font-style: italic; /* Курсивное начертание текста <em>/ } dt { margin-top: 1em; /</em>  
Отступ сверху */ }
```

Импорт камер выполняется через POST запрос на URL

```
http(s)://URL_ВАШЕГО_БОТЧЕРА/vsaas/api/interchange/usercameras
```

Импорт через команду в консоли:

```
curl http://127.0.0.1:8080/vsaas/api/interchange/usercameras --data-binary @mydata.csv -H 'Content-  
type:text/csv' -H 'x-vsaas-api-key: '
```

Пример файла mydata.csv:

```
stream_url,substream_url,thumbnails,onvif_url,onvif_profile,ptz,dvr_depth,dvr_path,enabled,access,title  
rtsp://127.0.0.1:554,,,http://127.0.0.1:8899,000,0,3,storage,1,private,office_cam1  
rtsp://127.0.0.2:554,,,http://127.0.0.2:8899,000,0,3,storage,1,private,office_cam2
```

Пример одной строкой:

```
echo -  
e "stream_url,substream_url,thumbnails,onvif_url,onvif_profile,ptz,dvr_depth,dvr_path,enabled,access,title\  
rtsp://127.0.0.1:554,,,http://127.0.0.1:8899,000,0,3,storage,1,private,office_cam1\br/>rtsp://127.0.0.2:554,,,http://127.0.0.2:8899,000,0,3,storage,1,private,office_cam2" | curl http://127.0.0.1:8080/vsaas/api/interchange/us  
ercameras --data-binary @- -H 'Content-type:text/csv' -H 'x-vsaas-api-key: 3a7d9386-6c3a-440d-a75d-  
e6b3fdc3368e'
```

Ответ:

```
{"zu": 0, "cameras": [{"name": "office_cam2-689f1b1548", "created": true}, {"name": "office_cam1-  
c0ce3faa10", "created": true}], "users": [], "success": true, "zc": 2}}
```

Надо передать CSV или JSON со следующими полями:

title: Название камеры.

name: Название потока. По умолчанию — title плюс случайный суффикс.

static: 1 — постоянно работающий поток (static), 0 — поток по запросу (ondemand).

stream\_url: RTSP URL основного потока.

substream\_url: RTSP URL вторичного видеопотока (для мультибитрейта). По умолчанию нет.

thumbnails (строка): URL снэпшотов с камеры. Отправив 0 вместо URL вы можете сбросить адрес thumbnails. Если вы не знаете URL снэпшотов с камеры, передайте в строке значение 1 и их включит Watcher (в данном варианте будет 10% увеличение нагрузки на сервер, рекомендуем использовать URL, для общения с камерой напрямую).

onvif\_url: URL по которому камера будет отвечать по onvif протоколу. По умолчанию нет.

onvif\_profile: ONVIF profile

ptz: (0 или 1) — вкл/выкл PTZ (если камера поддерживает)

access: Тип доступа к камере: (private/public/authorized) — видимость камеры, публичная она будет (для всех абонентов) или приватная (только для owner'a камеры). По умолчанию — private.

owner: (логин привязанного абонента / владельца камеры)

enabled: (0 или 1) — вкл/выкл камеры.

dvr\_path: Путь для сохранения архива. По умолчанию — архива нет.

dvr\_depth: (целое число в днях) — количество дней записи архива для камеры. 0 — отключает архив.

coordinates: Координаты.

postal\_address: Адрес.

comment: Комментарий.

agent\_model (строка): модель камеры.

agent\_serial (строка): серийный номер камеры.

agent\_id (строка): уникальный номер агента на камере.

agent\_key (строка): специальное поле, используемое для авторизации камеры в Flussonic Watcher.

agent\_pin (строка): специальное поле, используемое для авторизации камеры в Flussonic Watcher.

# Интеграция в существующую систему

В этой статье описан типовой сценарий внедрения Flussonic Watcher с контролируемой продажей камер по подписке и учетом абонентов и их услуг в сторонней системе. Далее будут использоваться термины:

клиент — клиент компании Эрливидео, владелец сервиса

абонент — абонент у клиента, пользователь сервиса

биллинг — система внешняя к Watcher, в ней ведется тарификация услуг клиента абонентам и взимание денег

Сценарий следующий:

На партию камер заливается модифицированная прошивка с Flussonic Agent

В этой прошивке зашита информация о том, к какому Flussonic Watcher надо привязать эту камеру

Камеру с серийником клиент заносит в систему инвентаризации биллинга, пока она ещё лежит на складе

При продаже абонентусотрудник клиента связывает в биллинге серийный номер камеры с идентификатором абонента

При первом включении камера от нашего сервера активации получает данные для авторизации в Flussonic Watcher.

Эти данные никак не связаны с идентификатором абонента, это авторизация камеры

Активированная камера немедленно начинает попытки соединения с Flussonic Watcher

Сервер активации посылает данные о камере напрямую в Watcher или в биллинг

Биллинг получает информацию о свежесозданной камере, добавляет к ней идентификатор абонента, занесенный ранее в систему инвентаризации

Биллинг отправляет информацию о свежесозданной камере в Watcher и камера там заводится. Этот и предыдущий пункты необходимо реализовать в рамках интеграции на стороне клиента

Теперь камера может подключиться к Watcher и начать отдавать видео на Flussonic

Важно то, что при такой организации процесса не требуется никакой настройки роутеров, камер и прочих сетевых устройств у абонента. После включения камеры в сеть она автоматически появится в личном кабинете.

Агент и модификация прошивки не являются обязательным требованием, всё может работать и со стандартными, неизменными камерами, ниже будут описаны детали.

Подробнее об агенте в подробной статье.

## API биллинга

Со стороны клиента надо реализовать в биллинге API для приема данных о свежесозданных камерах и отправку этих данных в Watcher. Такая схема с проксированием данных нужна для добавления информации о владельце камеры и услугах, которые доступны на этой камере.

Вся концепция подобного использования биллинга подразумевает, что именно он является центральным местом хранения данных в системе, а не Flussonic Watcher. Такая практика

является стандартной и позволяет централизованно управлять услугами в разных системах, связывая, например, умный дом и видеонаблюдение в едином проекте.

Сервер активации, обслуживаемый Эрливидео, присылает запрос на сконфигурированный url (по вопросу конфигурации необходимо обратиться к техподдержке) с CSV или списком JSON объектов.

Все данные, которые присылает сервер активации надо переслать в Flussonic Watcher без изменений, если только нет задачи по какой-либо причине их поменять. Так, например, может прийти флаг ptz=1, его можно выставить в 0, если не хочется вообще управлять этой камерой в Watcher.

Поля, передаваемые от сервера активации (Эрливидео) в биллинг:

agent\_model (строка) — модель камеры

agent\_serial (строка) — серийный номер камеры

agent\_id (строка) — уникальный номер агента на камере

agent\_key (строка) — специальное поле, используемое для авторизации камеры Watcher'ом

stream\_url (строка) — основной RTSP-URL потока

substream\_url (строка) — вторичный RTSP-URL потока

thumbnails (строка) — URL снейшотов с камеры

onvif\_url (строка) — URL по которому камера будет отвечать по onvif протоколу

onvif\_profile (строка) — служебное поле

ptz (0 или 1) — вкл/выкл PTZ

Пришедшая от сервера активации информация должна создавать в биллинге новые записи для камер, либо «склеивать» эту информацию с уже существующими в биллинге или другой системе учета камер, камеры по параметру agent\_serial (серийный номер камеры). Важно понимать, что agent\_id может поменяться в случае, если камеру сбросили или передали другому абоненту. Серийный номер у камеры меняться не должен.

Т.е. если в биллинге существует система инвентаризации в которой камера привязывается к абоненту до первого включения, то новая запись появляться не будет, вместо этого надо заполнить пропущенные поля в существующей строчке в БД.

На стороне биллинга можно быть реализована возможность дозаполнять атрибуты к камерам, например привязка к владельцу или детали по управлению услугами для формирования тарифных планов.

## Управление камерами в Watcher

Биллинг может управлять камерами, которые существуют в Watcher. В частности при получении данных от сервера активации он может дополнить данные своими полями и отправить информацию о камерах в Watcher.

Атрибуты заполняемые на стороне биллинга:

owner — (логин привязанного абонента).

dvr\_depth (целое число в днях) - кол-во дней записи архива для камеры. 0 — отключает архив.

enabled (0 или 1) - вкл/выкл камеры.

access (private/public) - видимость камеры, публичная она будет (для всех абонентов) или приватная (только для owner'a камеры).

Разработка механизма создания тарифных планов и логики, которая будет влиять на доступ,

глубину архива и прочие настройки, находится на стороне клиента.

После добавления полей индивидуальной настройки (owner, dvr\_depth, enabled, access) необходимо отправить расширенный список камер в формате CSV в Watcher, по ссылке: [http\(s\)://URL\\_ВАШЕГО\\_БОТЧЕРА/vsaas/api/v2/cameras/import](http(s)://URL_ВАШЕГО_БОТЧЕРА/vsaas/api/v2/cameras/import).

```
curl -v http://localhost:8080/vsaas/api/v2/cameras/import \  
-d \  
'{\  
  "name":"cam1","stream_url":"fake://clock","access":"private",\  
  "enabled":true,"dvr_depth":3,"agent_id":"123098456","agent_serial":"mJ00DnktZFc",\  
  "agent_key":"salt:secretkey","owner":"alex@smith.com","dvr_path":"movies"  
}' \  
-H "X-Vsaas-Api-Key: 7ab056b1-5bb1-4501-b528-d69538392842" \  
-H "Content-Type: application/json"  
...  
{  
  "deleted": 0,  
  "updated": 0,  
  "errors": {},  
  "created": 1  
}
```

Здесь X-Vsaas-Api-Key: 7ab056b1-5bb1-4501-b528-d69538392842 это API key из настроек Watcher.

В случае если биллинг отправляет в Watcher камеру с идентификатором несуществующего пользователя (поле owner), Watcher создает нового пользователя. Его пароль при этом не передается. Мы подразумеваем, что клиент не хранит пароли абонентов в открытом виде и никак не может передать. Для того, что бы абонент мог залогиниться в Watcher, необходимо настроить внешний авторизационный бекенд в Watcher.

## Управление пользователями (абонентами)

Параллельно с информацией по камерам можно отправлять в Watcher информацию об абонентах.

Вызов [http\(s\)://URL\\_ВАШЕГО\\_БОТЧЕРА/vsaas/api/v1/users](http(s)://URL_ВАШЕГО_БОТЧЕРА/vsaas/api/v1/users)

Пример:

```
curl -v http://localhost:8080/vsaas/api/v2/users \  
-d \  
'{\  
  "login":"user@domain.com","dvr_allowed":true,"external_id":"12346780"  
}' \  
-H "X-Vsaas-Api-Key: 7ab056b1-5bb1-4501-b528-d69538392842" \  
-H "Content-Type: application/json"  
...  
[  
  {  
    "authorized_ip": null,  
    "enabled": null,  
    "id": null,  
    "note": null,
```

```
"is_admin": null,  
"dvr_allowed": true,  
"notification_email": null,  
"external_id": "12346780",  
"login": "user@domain.com"  
}  
]
```

Полный список полей, которые можно отправлять по ссылке на полную документацию

# Бэкенд для авторизации пользователей

## Схема работы

Оператор реализует обработчик http-запросов, в котором делает необходимую ему логику по аутентификации абонентов (пример в комментарии ниже).

Оператор указывает в настройках Flussonic Watcher http url к аутентификационному бекенду (settings/authentication backend).

Абонент логинится в Flussonic Watcher (отправляет логин/пароль со страницы).

Flussonic Watcher передает их бекенду оператора в параметрах запроса.

Бекенд на основе собственных данных принимает решение о том пускать абонента или нет.

В случае, когда абоненту разрешен вход, бекенд возвращает http код 200.

В противном случае, бекенд возвращает http код 403.

Если абонент не был найден, ожидается код 404.

Вместе с разрешением на вход, бекенд может вернуть в теле ответа json с указанием актуальных групп этого абонента.

Получив ответ, Flussonic Watcher актуализирует информацию о пользователе в собственной базе.

При полученном разрешении обновляется хеш пароля абонента и список групп, в которые этот абонент входит.

При отказе пароль абонента в Flussonic Watcher аннулируется.

В случае, если аутентификационный бекенд был недоступен или не успел ответить за 2 секунды, проверка абонента происходит по внутренней базе Flussonic Watcher.

```
import falcon, json
```

```
class AuthResource:
```

```
    def on_get(self, req, resp):
```

```
        print "GET %r\n%r" % (req.uri, req.params)
```

```
        login = req.params.get('login', None)
```

```
        password = req.params.get('password', None)
```

```
        if not login or not password:
```

```
            print 'incorrect request login: %r, pass: %r' % (login, password)
```

```
            resp.status = falcon.HTTP_400
```

```
            return
```

```
        if login == 'user0':
```

```
            if password == 'letmein':
```

```
                return
```

```
            resp.status = falcon.HTTP_403
```

```
            return
```

```
        if login == 'user1':
```

```
            if password == 'letmein':
```

```
                resp.body = json.dumps(dict(groups=['a', 'b']))
```

```
return
resp.status = falcon.HTTP_403
return
```

```
resp.status = falcon.HTTP_404
```

```
app = falcon.API()
ad = AuthResource()
```

```
app.add_route('/auth', ad)
```

## Примеры запросов

Абонента пускаем, ему доступны группы a и b:

```
curl -vvv http://localhost:8001/auth?login=user1&password=letmein
```

```
* Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8001 (#0)
> GET /auth?login=user1&password=letmein HTTP/1.1
> Host: localhost:8001
> User-Agent: curl/7.47.0
> Accept: */*
>
```

Абонента пускаем, групп нет:

```
curl -vvv http://localhost:8001/auth?login=user0&password=letmein
```

```
* Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8001 (#0)
> GET /auth?login=user0&password=letmein HTTP/1.1
> Host: localhost:8001
> User-Agent: curl/7.47.0
> Accept: */*
>
```

Абонента не пускаем:

```
curl -vvv http://localhost:8001/auth?login=user0&password=wrong
```

```
* Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8001 (#0)
> GET /auth?login=user0&password=wrong HTTP/1.1
> Host: localhost:8001
> User-Agent: curl/7.47.0
> Accept: */*
>
```

Абонент не найден:

```
curl -vvv http://localhost:8001/auth?login=user10&password=wrong
```

```
* Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8001 (#0)
> GET /auth?login=user10&password=wrong HTTP/1.1
> Host: localhost:8001
> User-Agent: curl/7.47.0
> Accept: */*
>
```

# RADIUS

Для авторизации пользователей в вотчере может использоваться RADIUS сервер. Это может быть удобным при большом количестве пользователей.

Настраивается через веб-интерфейс, в настройках Watcher:

Строка `radius://ldap.erlyvideo.ru:1812/secret` состоит из трёх частей: хост, порт и секрет. Измените её в соответствии с настройками вашего RADIUS сервера.

Теперь при логине пользователей вотчер будет обращаться по указанному адресу по протоколу RADIUS. В Access-Request запросе вотчер посылает логин и пароль:

User-Name

User-Password

В ответ RADIUS сервер в случае успешной авторизации может послать список групп пользователя. Для передачи используется атрибут Filter-Id(11). Каждая группа передаётся в отдельном атрибуте.

Вотчер обращается к RADIUS каждый раз при входе пользователя.

Если RADIUS отвечает Access-Асерт, вотчер пускает пользователя, сохраняет в БД хэш пароля и принадлежность к группам.

Если RADIUS отвечает Access-Reject, пользователь блокируется в БД.

Если RADIUS не ответил, вотчер ищет пользователя в БД.

Необходимо учитывать, что RADIUS должен знать про всех пользователей, в том числе администраторов. Признак, является ли пользователь администратором, не может быть передан в ответе RADIUS и назначается только через вотчер.

# API для мобильных приложений

API для разработки своих собственных мобильных приложений.

POST: /vsaas/api/login

Пример:

```
`curl -H 'Content-Type: application/json' -  
d '{"email": "email@example.com", "password": "passwwd"}' http://localhost:8080/vsaas/api/login`
```

Параметры:

HTTP request payload  
{ "email": , "password": }

Ответ запроса: JSON вида

```
{  
  success: boolean,  
  session:  
}
```

## Добавление камеры в Избранное

POST: /vsaas/my/fav/cameras

Параметры:

HTTP request payload  
{ "id": } // идентификатор камеры

Ответ запроса: JSON вида

```
{"success": true}
```

## Удаление камеры из Избранного

DELETE: /vsaas/my/fav/cameras/{cam\_id}

Параметры:

cam\_id идентификатор камеры

Ответ запроса: JSON вида

```
{"success": true}
```

## Все доступные камеры

GET: /vsaas/api/my/cameras

Параметры:

HTTP request headers  
x-vsaas-session: <session\_token>

## Список избранных камер

GET: /vsaas/api/my/cameras?filter=favorite

Параметры:

HTTP request headers

x-vsaas-session: <session\_token>

Ответ запроса: JSON вида

```
{
  "id": string
  "title": string
  "access": string // ["private", "public", "authorized"]
  "comment": string
  "postal_address": string
  "coordinates": string // "55.7512 37.6184"
  "streamer_url": string // "http://demo-watcher.flussonic.com"
  "dvr_depth": int // глубина архива в днях
  "permissions": { // доступные пользователю права
    "dvr": boolean // просмотр архива
    "ptz": boolean // управление PTZ
    "view": boolean // просмотр live
  }
  "ptz": boolean // доступно ли PTZ управление камеры
  "server": string // demo-watcher.flussonic.com
  "static": boolean // тип потока: статический или по запросу (on-demand)
  "status": { // информация о состоянии потока
    "source_error": string
    "alive": boolean
  }
  "thumbnails": boolean // включена ли генерация снэпшотов с камеры
  "urls": { // ссылки на различные протоколы получения видеопотока
    "hls": string
    "hds": string
    "jpg": string
    "rtmp": string
    "rtsp": string
    "media_info": string // URL информации о потоке
    "recording_status": string // URL состояния архива
  }
}
```

# Управление камерами

Сразу после установки модуля в системе ещё нет камер, даже если они настроены на Flussonic Media Server:

Для добавления камеры в систему нажмите «Добавить» :

Название. Должно быть заполнено английскими буквами и цифрами, поскольку будет использоваться в URL-адресах.

URL. Нужно указать адрес камеры. Например: `rtsp://mycam.local/stream0`

Дополнительно. Дополнительный адрес. Например: `rtsp://mycam.local/stream1;`  
`rtsp://mycam.local/stream2`

Дополнительно. Дополнительный адрес. Например: `rtsp://mycam.local/stream1;`  
`rtsp://mycam.local/stream2`

URL скриншотов. Адрес снятия скриншотов с камеры.

Расположение. Расположение камеры. Также можно указать его на карте.

Адрес. Почтовый адрес камеры.

Владелец. Выбор владельца камеры. Владельцы настраиваются в меню Пользователи и Группы. Владельцу камера будет доступна вне зависимости настроек доступа.

Доступ. Выбор доступа к камере: публичный, для клиентов и частный:

Публичный (public) — доступна всем пользователям Flussonic Watcher, вне зависимости от настроек групп и пользователей. При установке значения public источник появится в списке камер у всех пользователей вашего сервера. Также без ограничений по доступу камеру можно опубликовать на web-странице с постоянным доступом без авторизации.

Для клиентов (authorized) — также доступна для всех пользователей, но требуется авторизация на вашем Flussonic Watcher.

Частная (private) — доступна пользователю только при добавлении доступа к камере в разделе Пользователи и Группы.

Публичный (public) — доступна всем пользователям Flussonic Watcher, вне зависимости от настроек групп и пользователей. При установке значения public источник появится в списке камер у всех пользователей вашего сервера. Также без ограничений по доступу камеру можно опубликовать на web-странице с постоянным доступом без авторизации.

Для клиентов (authorized) — также доступна для всех пользователей, но требуется авторизация на вашем Flussonic Watcher.

Частная (private) — доступна пользователю только при добавлении доступа к камере в разделе Пользователи и Группы.

Сервер. Выбор сервера Flussonic Media Server.

Архив. глубина записи. Для того, что бы активировать эту опцию, надо настроить хранилище в настройках системы или настройках стримера;

Параметры.

Включение и отключение камеры.

Постоянно работает камера или по запросу.

Доступно изображение для предпросмотра или нет.

Ограничение доступа к архиву камеры.

Включение и отключение камеры.

Постоянно работает камера или по запросу.

Доступно изображение для предпросмотра или нет.

Ограничение доступа к архиву камеры.

Примечание. Текстовый комментарий. Например, это может быть описание того, куда камера смотрит.

Нажмите «Сохранить». После этого камера появится на странице управления и в дашборде:

# Создание клиентской мозаики

## Мозаика

— это возможность собрать на отдельной странице до 16 любых камер по выбору пользователя.

## Создание мозаики

Чтобы создать мозаику, зайдите в Flussonic Watcher с правами администратора. Затем:

- 1) Перейдите в раздел «Камеры».
- 2) Нажмите кнопку «Мозаика».
- 3) Нажмите кнопку «Добавить».
- 4) В открывшемся окне необходимо:
  - Написать название мозаики.
  - Отметить чекбокс «включена».
  - Выбрать количество камер в мозаике.
  - Перетащить нужные камеры в сетку. Если камер много, то можно воспользоваться поиском.
- 5) Нажмите «Сохранить».

Отредактировать мозаику или удалить её можно в разделе «Камеры» > «Мозаики».

Чтобы временно отключить доступ пользователей к мозаике, уберите чекбокс «включена» в настройках.

Все созданные мозаики доступны в «Мои камеры» > «Мозаика».

# Управление пользователями

После инсталляции модуля на странице управления пользователями будет только один пользователь-администратор:

## Добавление пользователя

Для добавления нового пользователя необходимо ввести:

Логин

Пароль

Внутренний серый ip-адрес для автологина

E-mail

Чекбоксами отметить:

Активен. Отметить, чтобы аккаунт был активен.

Администратор. Отметить, если пользователю необходимы права администратора.

Доступ к архиву. Отметить, если пользователю необходим доступ к архиву записей.

Также для пользователя можно оставить комментарий.

## Добавление камер

Чтобы добавить пользователю камеры, нажмите на тег «0 камер», затем «Добавить» и в открывшемся списке выделите те камеры, которые доступны пользователю.

Нажмите «Удалить», если хотите закрыть пользователю доступ к камере.

Кликнув на тег «0 групп», вы можете добавить пользователю группу камер. Подробнее в статье Группы.

После добавления камер они будут доступны пользователю в дашборде Мои камеры.

# Группы

С помощью групп вы сможете упростить навигацию по камерам и пользователям, делая логическую группировку (например, по этажам, отделам, регионам, микрорайонам и т.д.), а также облегчить процесс распределения прав видимости камер. Достаточно создать группу с камерами, а затем добавить в нее пользователя, которому в несколько кликов станут доступными все приватные камеры данной группы.

Чтобы создать группу камер и пользователей перейдите в меню «Пользователи и группы» > «Группы» > «Добавить».

Затем укажите название новой группы, описание (при необходимости) и нажмите «Сохранить».

В списке групп вы увидите созданную группу.

Кликнув кнопку «0 камер», вы сможете добавить в группу камеры из списка, а кликнув кнопку «0 пользователей», вы сможете добавить в группу пользователей.

Тоже самое можно сделать, кликнув на группу > «Камеры» > «Добавить».

В списке добавляемых пользователей вы сможете поставить флаг на видимость архива (DVR) для каждого юзера, а также возможность управлять камерой PTZ (не для всех камер).

Важно! Чтобы дать доступ к камере ограниченному количеству пользователей с помощью групп, необходимо присвоить камере статус Private. Статус Public делает камеру публичной и ее видно всем пользователям, независимо от того, в группе она или нет.

# Предоставление доступа к камерам

С помощью инструментов предоставления доступа к камерам у администратора есть возможность выдавать доступ к приватным камерам конкретным пользователям Flussonic Watcher, не создавая группу.

Для этого зайдите в раздел «Пользователи и группы» и возле конкретного пользователя нажмите на «0 расшаренных камер». Затем нажмите «Добавить».

В списке доступных камер отметьте чекбоксами нужные и нажмите «Добавить». После этого выбранный пользователь получит доступ к приватным камерам, не будучи их владельцем.

То же самое можно сделать через карточку пользователя. Нажмите в верхней части на «0 камер» и у вас появится возможность добавить одну или несколько камер, которые станут доступными для данного пользователя.

Предоставлять доступ к камерам могут только администраторы, у пользователей такой возможности пока нет.

# Серийные номера

Для Flussonic Watcher с камерными агентами необходимо настроить автопривязку пользователя к проданной камере.

Для этого:

- 1) Заведите пользователя в разделе Пользователи и Группы.
- 2) Зайдите в раздел Камеры > Серийные номера.
- 3) Добавьте серийный номер с коробки камеры и укажите пользователя, которому она принадлежит.

При первом включении камеры произойдет ее активация, провижнинг на сервер, а далее привязка пользователя к камере по серийному номеру. Также применится предустановленное значение глубины архива.

Для пользователя весь процесс первого включения будет выглядеть так:

Включил камеру в сеть.

Подключил камеру к интернету.

Зашел в Flussonic Watcher по выданному логин/паролю.

Камера уже доступна и ее можно смотреть.

# Импорт и Поиск камер

## Импорт из Flussonic Media Server

В простом случае запуска Flussonic Watcher с камерами доступными по локальной сети, можно воспользоваться как поиском по Onvif, так и импортом потоков из Flussonic Media Server.

Можно импортировать в Flussonic Watcher камеру, уже добавленную в Flussonic Media Server.

## Поиск камер

Flussonic Watcher может искать Onvif-совместимые камеры с помощью механизма WS-Discovery. Это протокол обнаружения сервисов, который работает в камерах Ubiquity, Samsung и т.д.

Некоторые камеры не смогут определиться без логина/пароля, поэтому, если не получается найти нужную камеру, то надо ввести логин/пароль и ещё раз запустить поиск.

У многих камер доступны два и более различных потоков, вы сможете добавить любой из них (H264):

После выбора профиля камеры, в поле RTSP URL установится правильное значение. Для добавления камеры останется только ввести для нее имя.

Также необходимо настроить путь для DVR в настройках системы.