

# Pixel streaming

1. [Что такое pixel-streaming?](#)
2. [Из каких частей состоит pixel-streaming?](#)
3. [Как работает pixel-streaming?](#)
4. [Как развернуть pixel-streaming?](#)
5. [Как добавить приложение в pixel-streaming?](#)
6. [Как обновить pixel-streaming?](#)

## Что такое pixel-streaming?

**pixel-streaming** – это система которая позволяет запускать множество приложений на множестве компьютеров, транслировать их пользователям через браузер с возможностью управления и общения

## Из каких частей состоит pixel-streaming?

**Coordinator** – модуль системы pixel-streaming который отвечает за внешнее взаимодействие с пользователем

**Session-server** – модуль системы pixel-streaming который отвечает за создание и уничтожение сессий на отдельном компьютере

**WebRTC-Session** – модуль системы pixel-streaming который отвечает за управление пользователями внутри сессии

**Application** – приложение выполненное на Unreal Engine с включённым pixel streaming plugin

**NGINX** – программа осуществляет поддержку SSL и маршрутизацию для [coordinator, session-server, WebRTC-Session]

**MongoDB** – база данных которая используется coordinator для хранения информации доступной клиентам, и session-server для хранения информации о запущенных сессиях

## Как работает pixel-streaming?

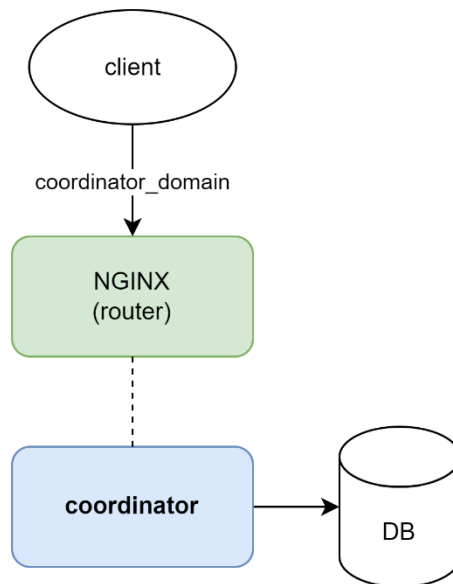
Pixel-streaming – состоит из 3х частей

### 1. Coordinator

Coordinator – часть системы pixel-streaming которая отвечает за всё внешнее взаимодействие с пользователем.

Coordinator знает доступные к запуску приложения, доступные сервера сессий, запущенные сессии.

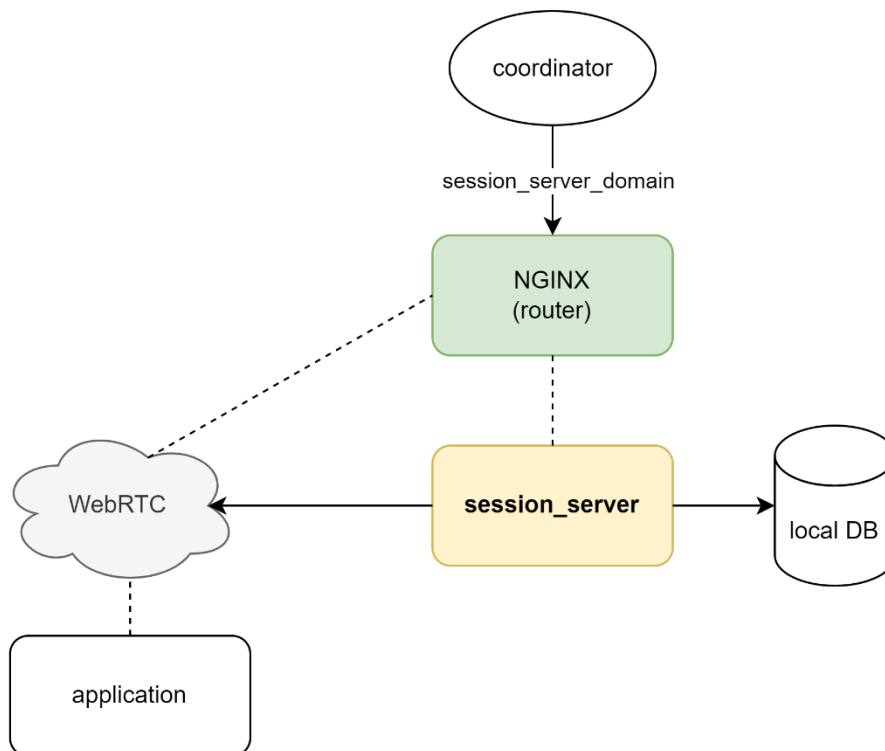
Через coordinator пользователь может посмотреть список доступных приложений, создать или запланировать сессию.



## 2. Session-server

Session-server – часть системы pixel-streaming которая отвечает за создание и уничтожение сессий на отдельном серверном компьютере.

Session-server создаёт сессию по запросу coordinator, после завершения сессии session-server оповещает coordinator о завершении сессии.

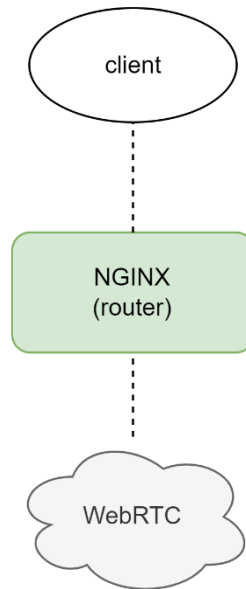


## 3. WebRTC-Session

WebRTC-Session – часть системы pixel-streaming которая отвечает за передачу управления, видео и аудио потока пользователям внутри сессии.

WebRTC-Session это связка из двух приложений WebRTC сервера который отвечает за передачу видеопотока приложения к пользователю, и передачу управления от пользователя к приложению.

Если все пользователи сессии неактивны в течение определённого количества времени, сессия завершится.



## Как развернуть pixel-streaming?

### a) Coordinator (Linux)

1. git pull <http://212.220.216.185:3000/GraffTech/pixel-streaming-coordinator>
2. cd pixel-streaming-coordinator/lib
3. git submodule init
4. git submodule update
5. npm install pm2 -g
6. cd pixel-streaming-coordinator
7. npm install
8. создайте .env и allowed\_cors.js файлы описанные в README.md внутри pixel-streaming-coordinator
9. pm2 start app.js --name coordinator
10. pm2 save
11. pm2 startup
12. execute in terminal output of pm2 startup
13. настройте nginx и откройте порты 80 и 443 в файрволе и роутере

### b) Session-server (Windows)

1. git pull <http://212.220.216.185:3000/GraffTech/pixel-streaming-session-server-package>
2. откройте проводник и установите всё из папки pixel-streaming-session-server/should\_install
3. запустите powershell от имени администратора

3. cd pixel-streaming-session-server-package
4. python enable\_policy.py
5. создайте файл config.py на примере config.py.example
5. python install.py
6. используйте "pm2 status" или "pm2 monit" или "pm2 logs" чтобы проверить что session-server запущен
7. запустите "python generate\_nginx\_config.py" после чего будет сгенерирован маршрутизирующий конфиг nginx для сервера сессий и сессий с расширением .conf
8. настройте nginx и откройте порты 80 и 443 в файрволе и роутере
9. перейдите в глобальную базу данных и добавьте информацию о сервере сессий в коллекцию "session\_server", пример ниже:

```
limit: 4
url: "https://a1.test.sess.graff.tech/s1"
▼ title: Array
  0: "mosharov"
  1: "ivazowsky"
```

## Как добавить приложение в pixel-streaming?

1. откройте файл config.py в pixel-streaming-session-server-package, добавьте элемент в переменную applications (название приложения И путь к приложению)
2. сконфигурируйте сервер сессий с помощью "python configure.py"
3. перезапустите сервер с помощью "python restart.py"
4. перейдите в глобальную базу данных и добавьте информацию о новом приложении в коллекцию "title", пример ниже:

```
app_title: "ivazowsky"
title: "Айвазовский City"
language: "en"
location: "Russia, Tyumen."
description: "A new residential area located in a picturesque location on the coast..."
feature_description: "Winner of the prestigious Urban Awards 2019 in the nomination «The Bes..."
icon: "https://images.graff.tech/pixel-streaming/applications/ivazowsky/icon..."
logo: "https://images.graff.tech/pixel-streaming/applications/ivazowsky/logo..."
preview: "https://images.graff.tech/pixel-streaming/applications/ivazowsky/previ..."
```

5. перейдите в глобальную базу данных и добавьте информацию о новом приложении в коллекцию "session\_server" в документе текущего сервера в разделе "title", пример ниже:

```
limit: 4
url: "https://a1.test.sess.graff.tech/s1"
▼ title: Array
  0: "mosharov"
  1: "ivazowsky"
```

# Как обновить pixel-streaming?

## a) Coordinator

1. перейдите в директорию приложения
1. git pull
2. pm2 restart coordinator
3. pm2 status (чтобы проверить статус **coordinator**)

## b) Session-server

1. перейдите в директорию pixel-streaming-session-server-package
2. git pull
3. python upgrade.py
4. python restart.py
5. pm2 status (чтобы проверить статус **session-server**)