Date	Version	Description	Author
17.01.2020	1.0	Final version	Dominik Kotarski

BUMPY

for the Cycling Advocacy of the Zagreb Cyclists' Union

Installation instructions

Index of Contents

1. Introduction

1.1 Purpose of the Document

The purpose of this document is to provide installation instructions for the backend service and web interface modules of bumpy.

1.2 Intended Audience

- The customer
- The project's team
- Project supervisors
- All other project stakeholders

1.3 References

Cyclist's Union: http://sindikatbiciklista.hr/en/
Fakultet elektrotehnike i računarstva: https://www.fer.unizg.hr/en
Politecnico di Milano: https://www.polimi.it/en/

2. Instructions

These instructions are written for and tested on Ubuntu server 18.04.3. If you are using a different linux distribution, some installation steps such as package and service installation may differ. If you encounter difficulties, please refer to the documentation of your distribution's package manager (Ubuntu's apt) and init system (Ubuntu's systemd).

2.1 Prerequisites

Make sure you are in a shell session with a sudo enabled user. We will use this user to setup the project's files and run the services.

2.2 Installing dependencies

You will need to install the latest version of python3, python3-pip, virtualenv rabbitmq, nginx, mongo-db:

```
sudo apt update
sudo apt install python3-pip python3 rabbitmq-server nginx
mongodb-server
```

2.3 Downloading the code and setting up directories

First, install and update your git installation

```
sudo apt update
sudo apt install git
```

Second, create a folder for bumpy of which you will be the owner and will have full permissions for. We will create this folder in our home directory. Clone the bumpy project repository (DSD-Cycling-Advocacy) into that folder.

```
cd ~
mkdir bumpy
cd bumpy
git clone https://github.com/SandraKuzmic/DSD-Cycling-Advocacy.git
```

2.4 Setting up backend

2.4.1 Setting up a virtual environment

First, install virtualenv, a tool for managing python virtual environments. A virtual environment is used in order to localize python version and packages to a project.

```
sudo apt install virtualenv
```

Then, navigate to the backend project root and create a python virtual environment. Note that we use a -p prefix, indicating we want to use python3 for our virtual environment.

```
cd DSD-Cycling-Advocacy/backend
virtualenv venv -p python3
```

NOTE: Ubuntu 18.04.3's default python3 version is 3.6.9. Our project was tested and runs with this python version. We recommend to take a look at pyenv if you need to use a non-distribution-default python3 with virtualenv.

Next, we need to activate the virtual environment. We do this by sourcing a shell script, which configures your shell session to run python related commands from inside our virtual environment. In the case of a default bash shell on ubuntu, we source the plain, no extension activate script.

```
source venv/bin/activate
```

Your shell prompt should change to something like this

```
(venv) user@host:~/bumpy/DSD-Cycling-Advocacy/backend$
```

Now, make sure you are running the correct python version inside your environment

```
python --version
```

In our case, this outputs Python 3.6.9.

2.4.2 Installing backend specific dependencies and web server

Now, we will install all the dependencies for our backend python service. We do this by running pip, python's package manager. We use the -r option, which takes a provided

file, and parses it, downloading and installing all listed packages. Make sure you have activated the virtual environment from the previous step first, otherwise, you will install these for your whole system.

```
pip install -r requirements.txt
```

Additionally, we will install gunicorn, which is not a dependency, but a production wsgi server for python web applications.

```
pip install gunicorn
```

Before we proceed to creating a service for our gunicorn server, you might want to test if it works by launching it manually from the shell. Make sure you are located in the DSD-Cycling-Advocacy/backend directory

```
gunicorn --bind 0.0.0.0:5000 wsgi:app
```

You should see gunicorn launching and outputting some launch information. You can now close gunicorn with ctrl+c.

2.4.3 Setting up gunicorn systemd service

We will set up a simple systemd service to keep our gunicorn server up and running at all times. Copy the provided systemd.service file to /etc/systemd/system/.

```
sudo cp util/bumpy/bumpy-backend.service
/etc/systemd/system/bumpy-backend.service
```

Edit the file, replacing all red text with your information. You can adjust the worker count, defined by the red number 3, as necessary. Official gunicorn documentation suggests using 2*CPU +1 workers.
For more information on systemd service files consult the systemd documentation.

```
sudo nano /etc/systemd/system/bumpy-backend.service
```

```
[Unit]
Description=Gunicorn instance to serve bumpy-backend
After=network.target

[Service]
User=youruser
Group=www-data
WorkingDirectory=/home/youruser/bumpy/DSD-Cycling-Advocacy/backend
Environment="PATH=/home/youruser/bumpy/DSD-Cycling-Advocacy/backend/venv
/bin"
ExecStart=/home/youruser/bumpy/DSD-Cycling-Advocacy/backend/venv/bin/gun
icorn --workers 3 --bind unix:bumpy-backend.sock -m 007 wsgi:app

[Install]
WantedBy=multi-user.target
```

Now, we can start the service that we just defined, and enable it so it starts on boot.

```
sudo systemctl start bumpy-backend
sudo systemctl enable bumpy-backend
```

You can now deactivate the virtual environment

```
deactivate
```

The prompt should now return to its previous state

```
user@host:~/bumpy/DSD-Cycling-Advocacy/backend$
```

At this point, we still need to setup a reverse proxy server to forward the requests to the service we just defined. However, we will set up the frontend first.

2.5 Setting up frontend

Setting up frontend is pretty simple, as you only need to make a small change to the code and then build the node app. The rest of the work is in configuring nginx, which we need to configure for our backend service anyway.

Open the following file for editing and change the http://161.53.67.132:5000 to localhost.

```
cd ~
nano bumpy/DSD-Cycling-Advocacy/frontend/bumpy-web/src/setupProxy.js
```

```
...
ls
app.use(
   proxy('/v1', { target: 'http://161.53.67.132:5000', changeOrigin: true })
);
...
```

The resulting string should look like this:

```
'http://localhost/api/'
```

To build our node app, you need to install npm. Do this by executing these two commands

```
curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -
sudo apt install -y nodejs
```

Now, we will build our frontend app. Cd into the frontend app directory and run the commands to build the app.

```
cd ~/bumpy/DSD-Cycling-Advocacy/frontend/bumpy-web
sudo npm install
sudo npm run build
```

2.6 Setting up nginx to reverse proxy the traffic into our apps

Copy the provided nginx configuration file to `/etc/nginx/sites-available/`.

```
cd ~/bumpy/DSD-Cycling-Advocacy/backend
sudo cp util/bumpy.nginx /etc/nginx/sites-available/bumpy
```

Open the file for editing and replace the red text with your information

```
sudo nano /etc/nginx/sites-available/bumpy
```

```
server {
      listen 80;
      server_name server_domain_or_IP;
      root
/home/youruser/bumpy/DSD-Cycling-Advocacy/frontend/bumpy-web/build;
      index index.html;

      location = /api {
          return 302 /api/;
      }

      location /api/ {
      include proxy_params;
      proxy_pass
http://unix:/home/youruser/bumpy/DSD-Cycling-Advocacy/backend/bumpy-back
end.sock:/;
      }

      location / {
      }
}
```

Now, we need to enable our nginx configuration file. Do this by creating a symbolic link in the sites-enabled directory.

```
sudo ln -s /etc/nginx/sites-available/bumpy /etc/nginx/sites-enabled
```

Also, remove the default nginx configuration file from sites enabled

```
sudo rm /etc/nginx/sites-enabled/default
```

Check your configuration for errors

```
sudo nginx -t
```

If the check passes, you may restart the nginx service

```
sudo systemctl restart nginx
```

NOTE: If you want to configure https for your frontend, take a look at this link

2.6 Configuring the android application

Open the following file for editing and change the red text to your information.

```
cd ~
sudo nano
bumpy/DSD-Cycling-Advocacy/android/Bumpy/app/src/main/java/com/cycling_a
dvocacy/bumpy/net/service/BumpyServiceBuilder.java
```

```
...

   private static final String BASE_URL = "http://161.53.67.132:5000/v1/";

...
```

The resulting string should look like this:

```
"http://your-ip-or-domain/api/v1/"
```

Afterwards, consult the following link in order to publish your android app

https://developer.android.com/studio/publish

End of document