Prerequisites

python	Version 2.7
virtualenv or virtualenvwrapper	To create virtualenvs

Install System Dependencies

```
sudo apt-get install libncurses5-dev
```

Start and Run Project

1.- Create virtualenv virtualenv Wifibytes

```
mkvirtualenv Wifibytes
```

2.- Activate virtualenv

```
source Wifibytes/bin/activate
```

3.- Clone the repo into virtualenv

```
cd Wifibytes
git clone https://USUARIO@bitbucket.org/cactusagency/wifibytes-project.git
```

4.- Enter to Django project directory

```
cd Wifibytes-project
```

5.- Install all requirements

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

6.- Migrate BBDD

```
python manage.py migrate --settings=Wifibytes.settings.local
```

7.- Create super user

```
python manage.py createsuperuser --settings=Wifibytes.settings.local
```

8.- Run server

```
python manage.py runserver --settings=Wifibytes.settings.local
```

Check http://localhost:8000

Fix Error in Reportab version

locate util.py in your virtualenv site-packages replace this:

```
if not (reportlab.Version[0] == "2" and reportlab.Version[2] >= "1"):
    raise ImportError("Reportlab Version 2.1+ is needed!")

REPORTLAB22 = (reportlab.Version[0] == "2" and reportlab.Version[2] >= "2")
```

to:

```
if not (reportlab.Version[:3] >="2.1"):
    raise ImportError("Reportlab Version 2.1+ is needed!")

REPORTLAB22 = (reportlab.Version[:3] >="2.1")
```

Install postgresql	
sudo apt-get install -y postgresql-9.5 postgresql-contrib-9.5 sudo apt-get install -y postgresql-doc-9.5 postgresql-server-dev-9.5	Ubuntu 16.04
sudo apt-get install -y postgresql-9.3 postgresql-contrib-9.3 sudo apt-get install -y postgresql-doc-9.3 postgresql-server-dev-9.3	Ubuntu 14.04

Install NGINX with reverse proxy

```
sudo apt-get install -y nginx
```

Install GIT

```
sudo apt-get install git
```

Create unix user

```
sudo adduser wifibytes
sudo su wifibytes
```

GIT ssh key

Go to home

```
cd
```

Generate ssh key

```
ssh-keygen
```

Show key and copy

```
cat ~/.ssh/id_rsa.pub
```

Add this key to git repository

Clone project

```
git clone [url-repository]
```

Create virtualenv

```
mkvirtualenv wifibytes
```

Edit .bashrc and add this lines:

```
export WORKON_HOME=$HOME/.virtualenvs
source /usr/local/bin/virtualenvwrapper.sh
```

Django requirements

Go to inside project folder and install requirements

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

Postgres database init

Create bbdd

```
sudo su postgres
createdb wifibytes
createuser wifibytes
```

Access to psql

```
psql
```

Add new password

```
ALTER USER "username" WITH PASSWORD 'newpassword';
```

Exit to psql

```
\q
```

Django migrations to BBDD

Run migrations

Go to inside project and execute:

```
python manage.py migrate --settings=wifibytes.settings.production
```

NGINX Configuration

Login to root user

```
sudo su root
```

Go to sites-avaiable — **cd/etc/nginx/sites-available**

Create new file config — touch wifibytes && nano -c wifibytes

Add this lines:

```
upstream wifibytes_django {
server unix:///home/wifibyres/wifibytes.sock;
}
server {
listen 80;
server_name DOMINIO_DEL_PROYECTO.com;
client_max_body_size 0;
charset utf-8;
location /media {
    alias /home/wifibytes/wifibytes-project/wifibytes/media;
}
location /static {
    alias /home/wifibytes/wifibytes-project/wifibytes/assets;
}
location / {
    uwsgi_pass wifibytes_django;
    uwsgi_param QUERY_STRING $query_string;
    uwsgi_param REQUEST_METHOD $request_method;
    uwsgi_param CONTENT_TYPE $content_type;
    uwsgi_param CONTENT_LENGTH $content_length;
    uwsgi_param REQUEST_URI $request_uri;
    uwsgi_param PATH_INFO $document_uri;
    uwsgi_param DOCUMENT_ROOT $document_root;
    uwsgi_param SERVER_PROTOCOL $server_protocol;
    uwsgi_param REMOTE_ADDR $remote_addr;
    uwsgi_param REMOTE_PORT $remote_port;
    uwsgi_param SERVER_ADDR $server_addr;
    uwsgi_param SERVER_PORT $server_port;
    uwsgi_param SERVER_NAME $server_name;
    uwsgi_param UWSGI_SCHEME http;
}
}
```

Create symbolic link

```
ln -s /etc/nginx/sites-available/wifibytes /etc/nginx/sites-enabled/wifibytes
```

Reset nginx

```
/etc/init.d/nginx restart
```

Access in our user

```
sudo su wifibytes
```

Go to home and create this files: cd

.uwsgi.ini

```
[uwsgi]
# Django-related settings
# the base directory (full path)
chdir = /home/wifibytes/wifibytes-project/wifibytes
module = wifibytes.wsgi
home = /home/wifibytes/.virtualenvs/wifibytes
env=DJANGO_SETTINGS_MODULE=wifibytes.settings.production
master = true
processes = 2
socket = /home/wifibytes/wifibytes.sock
chmod-socket = 666
vacuum = true
loop = gevent
async = 50
threads = true
socket-timeout = 30
```

supervisord.conf

```
[unix_http_server]
file=/home/wifibytes/supervisor.sock ; (the path to the socket file)
[supervisord]
logfile=/home/wifibytes/logs/supervisord.log ; (main log file;default $CWD/supervisord.log)
pidfile=/home/wifibytes/supervisord.pid ; (supervisord pidfile;default supervisord.pid)
childlogdir=/home/wifibytes/logs/ ; ('AUTO' child log dir, default $TEMP)
nondaemon=false
nocleanup=true
[rpcinterface:supervisor]
supervisor.rpcinterface_factory = supervisor.rpcinterface:make_main_rpcinterface
[supervisorctl]
serverurl=unix:///home/wifibytes/supervisor.sock ; use a unix:// URL for a unix socket
[program:wifibytes]
command=/home/wifibytes/.virtualenvs/wifibytes/bin/uwsgi –ini uwsgi.ini
environment=PATH="/home/wifibytes/.virtualenvs/wifibytes/bin:/usr/bin/wget:/usr/bin/curl:/usr/bin/unzip:/usr/bin/zip"
logfile=/home/wifibytes/logs/wifibytes.log
logfile_maxbytes=50MB
logfile_backups=10
loglevel=info
environment=PATH="/home/wifibytes/.virtualenvs/wifibytes/bin"
logfile=/home/wifibytes/logs/worker.log
logfile_maxbytes=50MB
logfile_backups=10
loglevel=info
```

Create logs directory and vinculate to supervisord
mkdir logs supervisord -c supervisord.conf

Reset supervisord
supervisorctl restart all