Gunicorn

We need python3-pip, python3-venv and nginx to be installed on our system

First setup a virtual environment with python and activate it

I have made a different directory to go with this django project

mkdir django

cd django

python3 -m venv django1

source django1/bin/activate

Install django and gunicorn

pip install django gunicorn

Start django demo project in the same directory so that we can get manage.py here

django-admin startproject demo ~/django

To migrate the database if needed (optional)

python manage.py migrate

And we can directly test if this is working or not by running

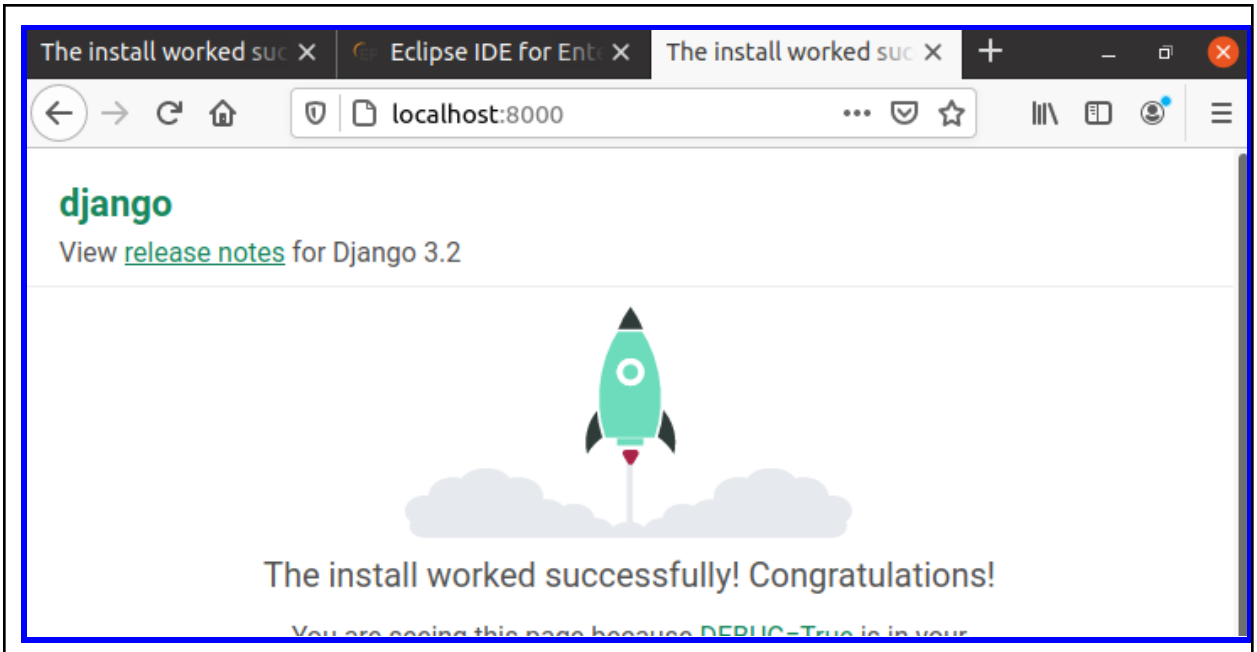
python manage.py runserver

Or we can define the port to run on by

python manage.py runserver 0..0.0.0:8000

```
(django1) bibek@bibek-lfTechnology:~/django$ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
November 14, 2021 - 09:52:16
Django version 3.2.9, using settings 'demo.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```



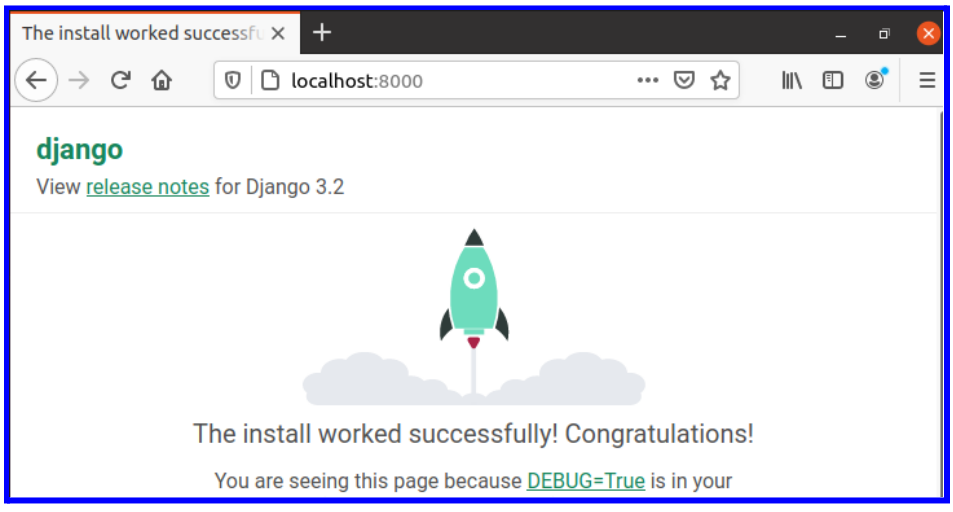
So it confirms that we can proceed now

To run this demo project through gunicorn

gunicorn --bind 0.0.0.0:8000 demo.wsgi

```
^C(django1) bibek@bibek-lfTechnology:~/django$ gunicorn --bind 0.0.0.0:8000 de
.wsgi
[2021-11-14 15:41:45 +0545] [10210] [INFO] Starting gunicorn 20.1.0
[2021-11-14 15:41:45 +0545] [10210] [INFO] Listening at: http://0.0.0.0:8000 (
10210)
[2021-11-14 15:41:45 +0545] [10210] [INFO] Using worker: sync
[2021-11-14 15:41:45 +0545] [10212] [INFO] Booting worker with pid: 10212
```

And again we can view the starter page is running



Now deactivate the environment and setup the reverse proxy through nginx

- **To secure our application**
- **And we don't need to execute the app server command gunicorn to host our app**

Make gunicorn.socket file inside system directory

sudo vim /etc/systemd/system/gunicorn.socket

And paste the following

```
[Unit]
Description=gunicorn socket

[Socket]
ListenStream=/run/gunicorn.sock

[Install]
WantedBy=sockets.target
~
```

Save and exit

Make gunicorn.service file and set the configurations

sudo vim /etc/systemd/system/gunicorn.service

And set the configuration according to your system

```
[Unit]
Description=gunicorn daemon
Requires=gunicorn.socket
After=network.target

[Service]
User=bibek
Group=www-data
WorkingDirectory=/home/bibek/django
ExecStart=/home/bibek/django/django1/bin/gunicorn \
          --access-logfile gunicorn.log \
          --error-logfile gunicorn.error.log \
          --workers 3 \
          --bind unix:/run/gunicorn.sock \
          demo.wsgi:application

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

Variables to set

- **User**: your user name

- **Working Directory**: the directory where we initialize the virtual environment and installed all the requirements of gunicorn
- **--access-log file**: to get the log of gunicorn
- **--error-logfile**: to get the error message of gunicorn
- **--workers** to set the number of worker process to serve the requests
- **--bind**: to bind the unix socket of gunicorn

Save and exit

Start the gunicorn service and enable the service

sudo systemctl start gunicorn.socket

sudo systemctl enable gunicorn.socket

Now configure the nginx to serve the gunicorn appserver

Make a file in sites-available of nginx

sudo vim /etc/nginx/sites-available/demo

I have given the same name as of starterproject of django

And and the configuration as below

```
server {
    listen 80;
    server_name localhost;

    location = /favicon.ico { access_log off; log_not_found off; }
    location /static/ {
        root /home/bibek/django;
    }

    location / {
        include proxy_params;
        proxy_pass http://unix:/run/gunicorn.sock;
    }
}
```

Adjust the parameter

- **Server name**: as per our host
- **Root directory**: where your project is located
- **proxy pass**: to map to the gunicorn.sock

Save And exit

Create soft link pf demo from sites-available to sites-enable of nginx to enable demo application

sudo ln -s /etc/nginx/sites-available/demo /etc/nginx/sites-enabled/

You can delete the default page rested in site-enabled directory if default page comes browsing localhost to disable default page

nginx -t

To check errors in the configuration file of nginx.

If it gives ok

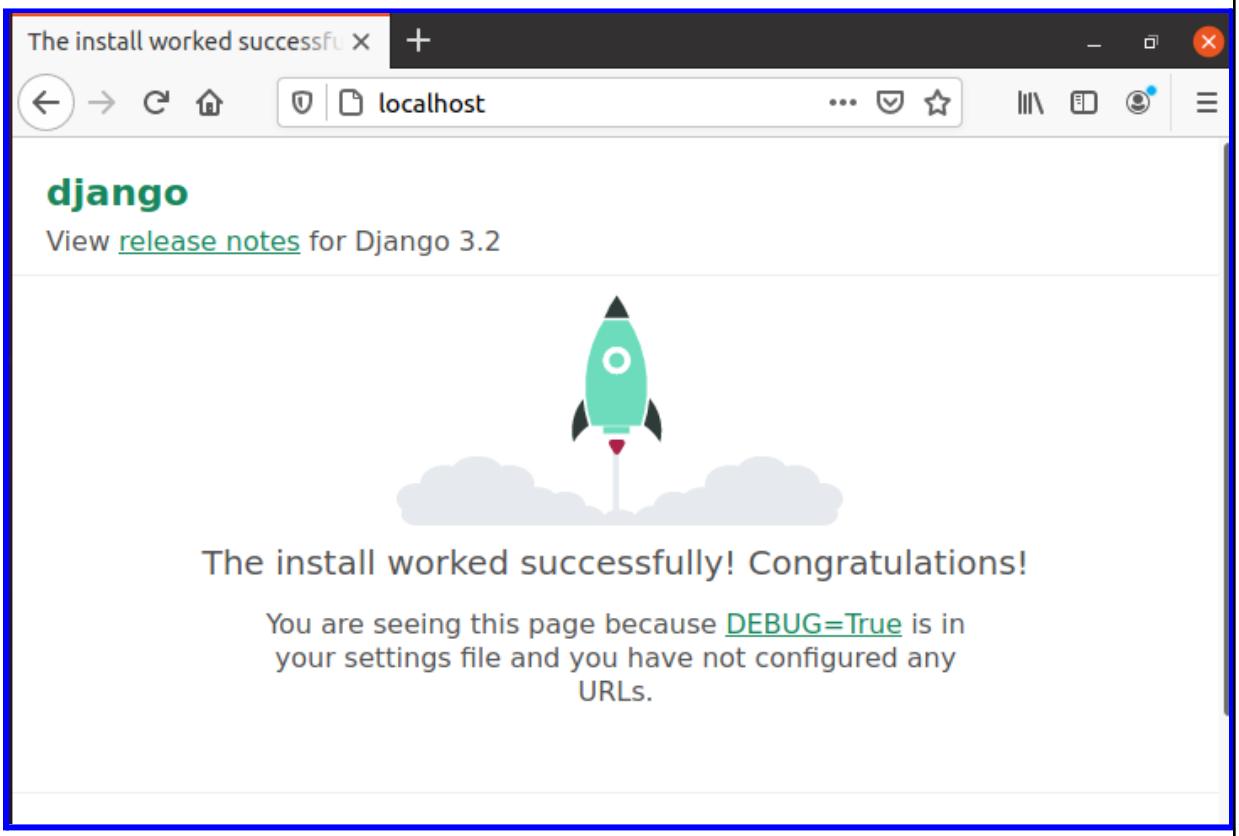
```
bibek@bibek-lfTechnology:~/django$ sudo !!
sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
bibek@bibek-lfTechnology:~/django$
```

Then we can restart the nginx service

sudo systemctl restart nginx

As nginx is serving on port 80

We can directly browse the starter web page on localhost



We can see the log file and error file of gunicorn in main directory of virtual environment

```
bibek@bibek-lfTechnology:~/django$ ls
conf  db.sqlite3  demo  django1  gunicorn.error.log  gunicorn.log  manage.py
bibek@bibek-lfTechnology:~/django$
```

And if we check the status of gunicorn we can see the three worker processes are running

```
error-logfile gunicorn.error.log --workers 3 --bind unix:/run
error-logfile gunicorn.error.log --workers 3 --bind unix:/run
error-logfile gunicorn.error.log --workers 3 --bind unix:/run
error-logfile gunicorn.error.log --workers 3 --bind unix:/run
```

To change the default port of gunicorn to 8089

Make a conf directory where our demo project is created

mkdir conf

Make a file gunicorn_config.py inside conf directory to set the port and other variables

vim conf/gunicorn_config.py

```
command ='/home/bibek/django/django1/bin/gunicorn'
pythonpath = '/home/bibek/django/demo'
bind = 'localhost:8089'
workers = 3
~
```

Parameters to be adjusted

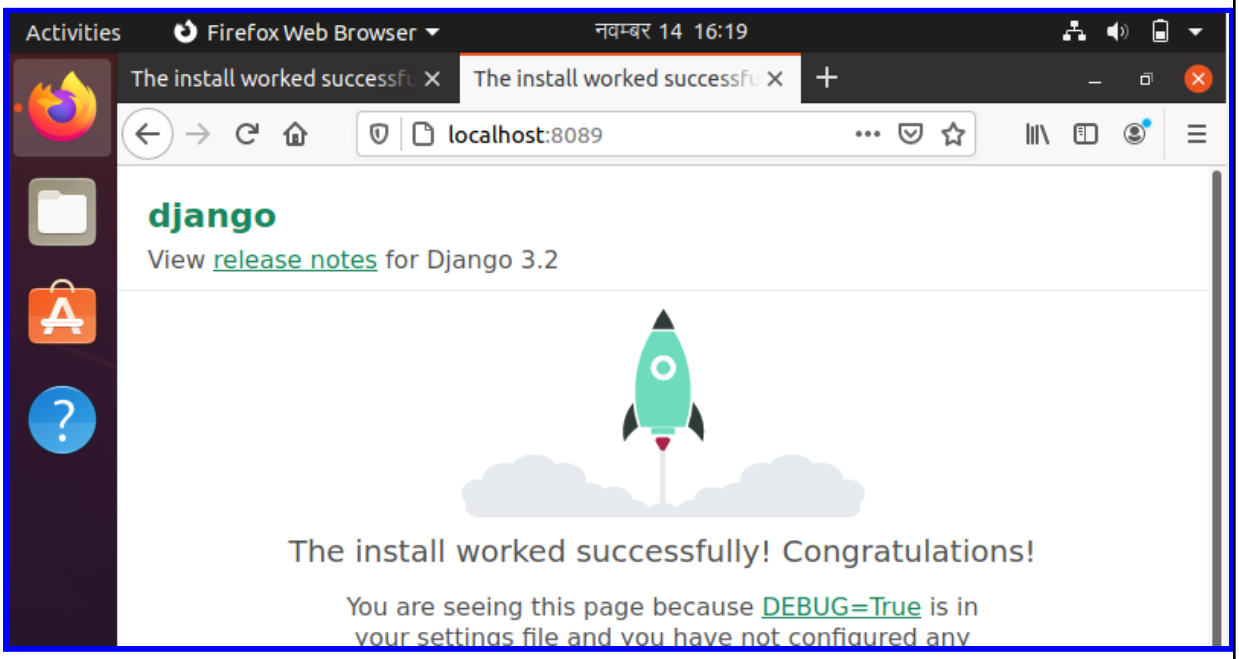
- **command**: the path to execute the gunicorn inside virtual env
- **pythonpath**: path to the starter page inside virtual env
- **bind**: ‘server:port’
- **workers** : number of worker node process

Activate the virtual environment and run gunicorn giving the path of conf file to serve

gunicorn -c conf/gunicorn_config.py demo.wsgi

```
bibek@bibek-lfTechnology:~/django$ ls
conf  db.sqlite3  demo  django1  gunicorn.error.log  gunicorn.log  manage.py
bibek@bibek-lfTechnology:~/django$ source django1/bin/activate
(django1) bibek@bibek-lfTechnology:~/django$ gunicorn -c conf/gunicorn_config.
py demo.wsgi
[2021-11-14 16:16:25 +0545] [10507] [INFO] Starting gunicorn 20.1.0
[2021-11-14 16:16:25 +0545] [10507] [INFO] Listening at: http://127.0.0.1:8089
 (10507)
[2021-11-14 16:16:25 +0545] [10507] [INFO] Using worker: sync
[2021-11-14 16:16:25 +0545] [10509] [INFO] Booting worker with pid: 10509
[2021-11-14 16:16:25 +0545] [10510] [INFO] Booting worker with pid: 10510
[2021-11-14 16:16:25 +0545] [10511] [INFO] Booting worker with pid: 10511
Not Found: /static/admin/css/fonts.css
^C[2021-11-14 16:18:44 +0545] [10507] [INFO] Handling signal: int
[2021-11-14 10:33:44 +0000] [10509] [INFO] Worker exiting (pid: 10509)
[2021-11-14 10:33:44 +0000] [10511] [INFO] Worker exiting (pid: 10511)
[2021-11-14 10:33:44 +0000] [10510] [INFO] Worker exiting (pid: 10510)
[2021-11-14 16:18:44 +0545] [10507] [INFO] Shutting down: Master
(django1) bibek@bibek-lfTechnology:~/django$
```

Now if we browse the localhost at 8089 port we can see the starter page



In this way we can set the default port for gunicorn to serve he application