2. Gunicorn

- **Create a Django starter project in a separate virtual environment.**
- **Deploy the 3 instances of application using Gunicorn in 8089 port.**
- **Dump access log in a file in non-default pattern.**
- **Dump error log in a file.**

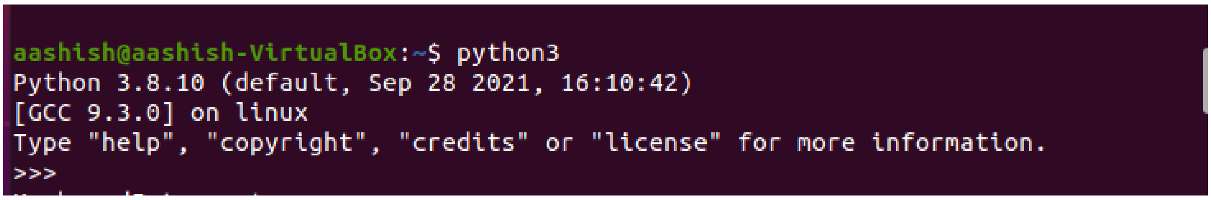
Answer:

First of all we need to download the following things into our system;

- **Python**
- **Pip**
- **Virtual Environment (venv)**

To check if python is installed or not we use;

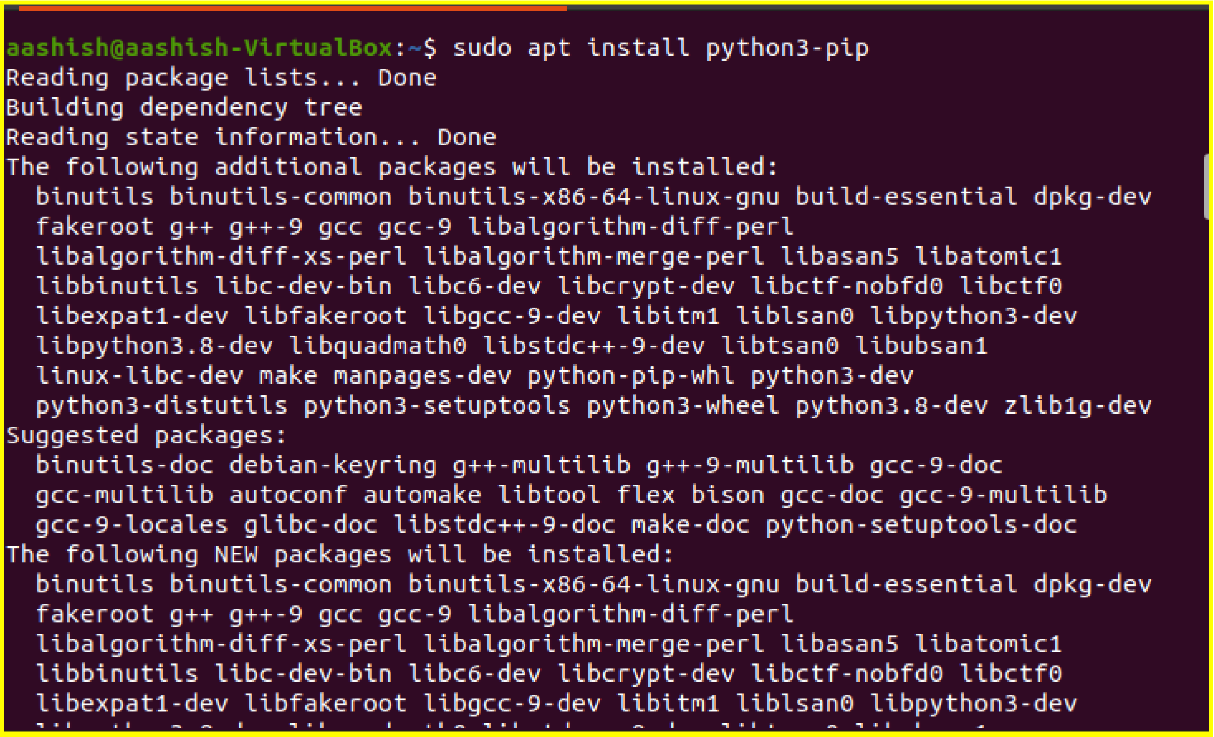
- **python3**
- **python --version**



```
aashish@aashish-VirtualBox:~$ python3
Python 3.8.10 (default, Sep 28 2021, 16:10:42)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

To install the pip we use;

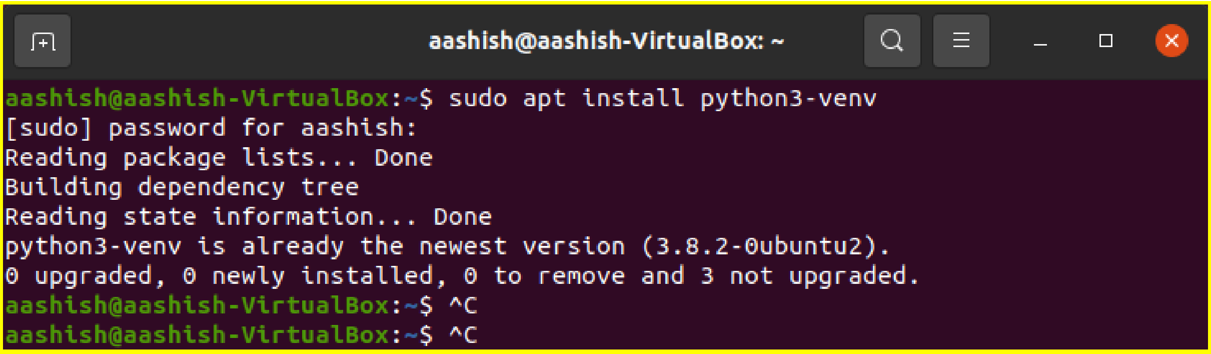
- **sudo apt install python3-pip3**



```
aashish@aashish-VirtualBox:~$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev
  fakeroot g++ g++-9 gcc gcc-9 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1
  libbinutils libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd0 libctf0
  libexpat1-dev libfakeroot libgcc-9-dev libitm1 liblsan0 libpython3-dev
  libpython3.8-dev libquadmath0 libstdc++-9-dev libtsan0 libubsan1
  linux-libc-dev make manpages-dev python-pip-whl python3-dev
  python3-distutils python3-setuptools python3-wheel python3.8-dev zlib1g-dev
Suggested packages:
  binutils-doc debian-keyring g++-multilib g++-9-multilib gcc-9-doc
  gcc-multilib autoconf automake libtool flex bison gcc-doc gcc-9-multilib
  gcc-9-locales glibc-doc libstdc++-9-doc make-doc python-setuptools-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev
  fakeroot g++ g++-9 gcc gcc-9 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1
  libbinutils libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd0 libctf0
  libexpat1-dev libfakeroot libgcc-9-dev libitm1 liblsan0 libpython3-dev
```

To install virtual environment we use;

- **sudo apt install python3-venv**



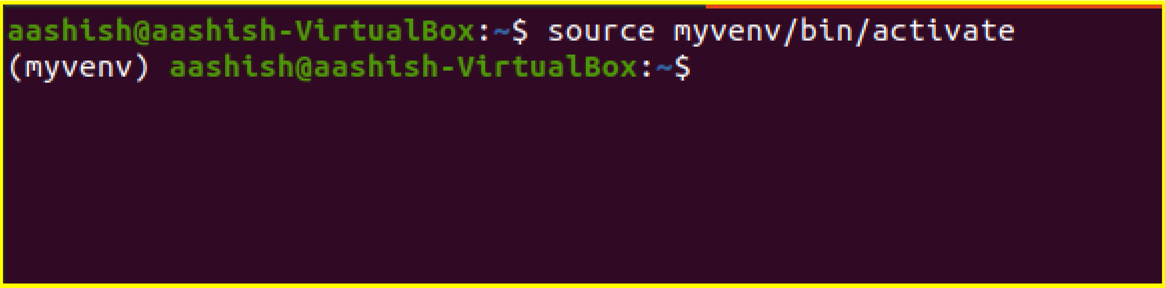
```
aashish@aashish-VirtualBox:~$ sudo apt install python3-venv
[sudo] password for aashish:
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-venv is already the newest version (3.8.2-0ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
aashish@aashish-VirtualBox:~$ ^C
aashish@aashish-VirtualBox:~$ ^C
```

To create a new virtual environment(**myvenv**) we use;

- **python3 -m venv myvenv**

And, to activate it we use;

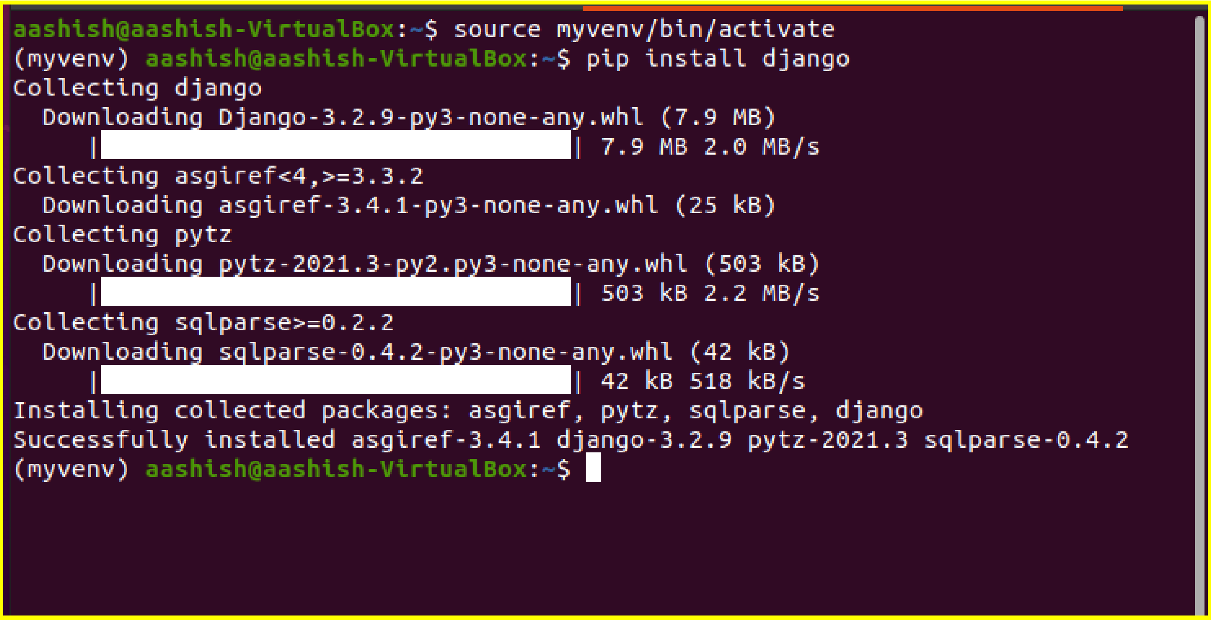
- **source myvenv/bin/activate**



```
aashish@aashish-VirtualBox:~$ source myvenv/bin/activate
(myvenv) aashish@aashish-VirtualBox:~$
```

Now, we install **Django** and **Gunicorn** in the **myvenv** or virtual environment.
To install Django we use;

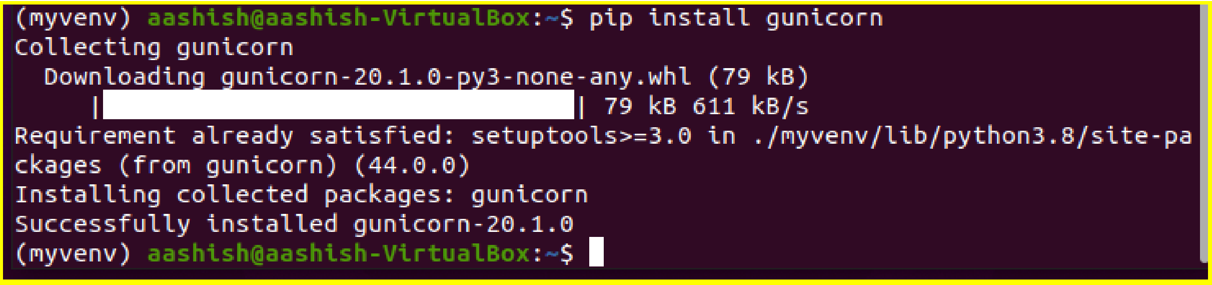
- **pip install django**



```
aashish@aashish-VirtualBox:~$ source myvenv/bin/activate
(myvenv) aashish@aashish-VirtualBox:~$ pip install django
Collecting django
  Downloading Django-3.2.9-py3-none-any.whl (7.9 MB)
     |████████████████████████████████| 7.9 MB 2.0 MB/s
Collecting asgiref<4,>=3.3.2
  Downloading asgiref-3.4.1-py3-none-any.whl (25 kB)
Collecting pytz
  Downloading pytz-2021.3-py2.py3-none-any.whl (503 kB)
     |████████████████████████████████| 503 kB 2.2 MB/s
Collecting sqlparse>=0.2.2
  Downloading sqlparse-0.4.2-py3-none-any.whl (42 kB)
     |████████████████████████████████| 42 kB 518 kB/s
Installing collected packages: asgiref, pytz, sqlparse, django
Successfully installed asgiref-3.4.1 django-3.2.9 pytz-2021.3 sqlparse-0.4.2
(myvenv) aashish@aashish-VirtualBox:~$
```

To install Gunicorn we use;

- **pip install gunicorn**



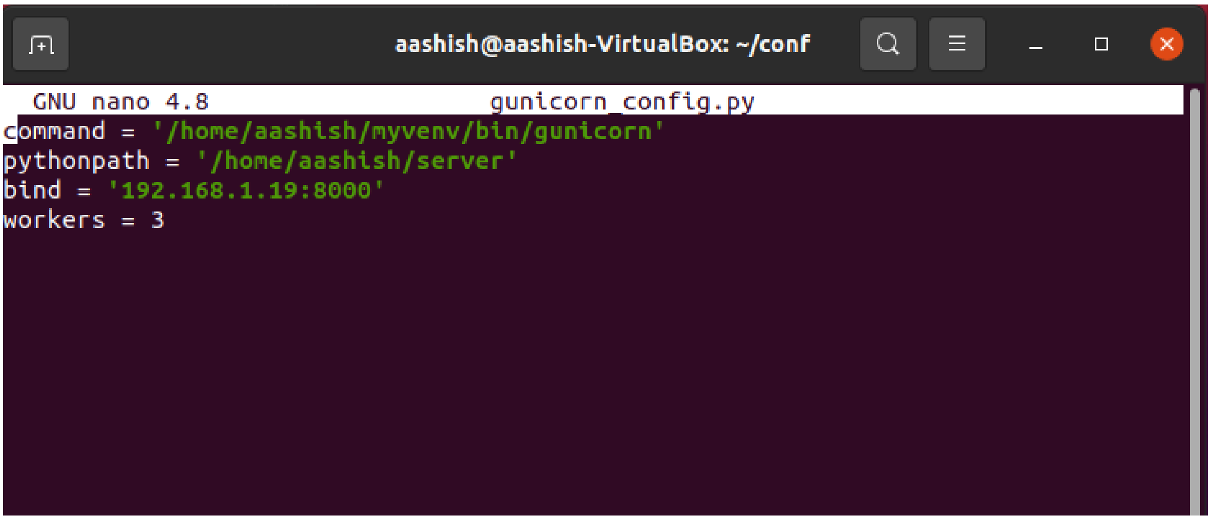
```
(myvenv) aashish@aashish-VirtualBox:~$ pip install gunicorn
Collecting gunicorn
  Downloading gunicorn-20.1.0-py3-none-any.whl (79 kB)
     |████████████████████████████████| 79 kB 611 kB/s
Requirement already satisfied: setuptools>=3.0 in ./myvenv/lib/python3.8/site-pa
ckages (from gunicorn) (44.0.0)
Installing collected packages: gunicorn
Successfully installed gunicorn-20.1.0
(myvenv) aashish@aashish-VirtualBox:~$ 
```

To create a new **Django** project named **server** we use;

- **django-admin startproject server**

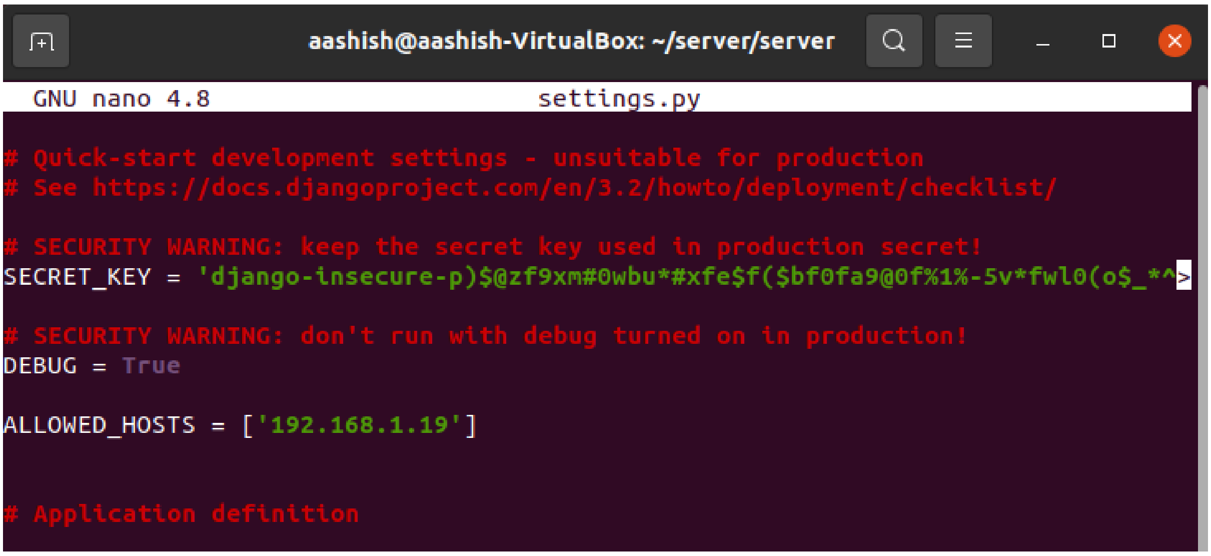
Now, we need to create a configuration file for the gunicorn server. For that we use **gunicorn_config.py** as configuration file name. We create a new directory named as **config** where **gunicorn_config.py** file is created. We edit the file as following;

- nano config/gunicorn_config.py
- **command = ‘/home/aashish/myvenv/bin/gunicorn’**
- **pythonpath = ‘/home/aashish/server’**
- **bind = ‘192.168.1.19:8000’**
- **workers = 3**



```
  GNU nano 4.8                    gunicorn_config.py
command = '/home/aashish/myvenv/bin/gunicorn'
pythonpath = '/home/aashish/server'
bind = '192.168.1.19:8000'
workers = 3
```

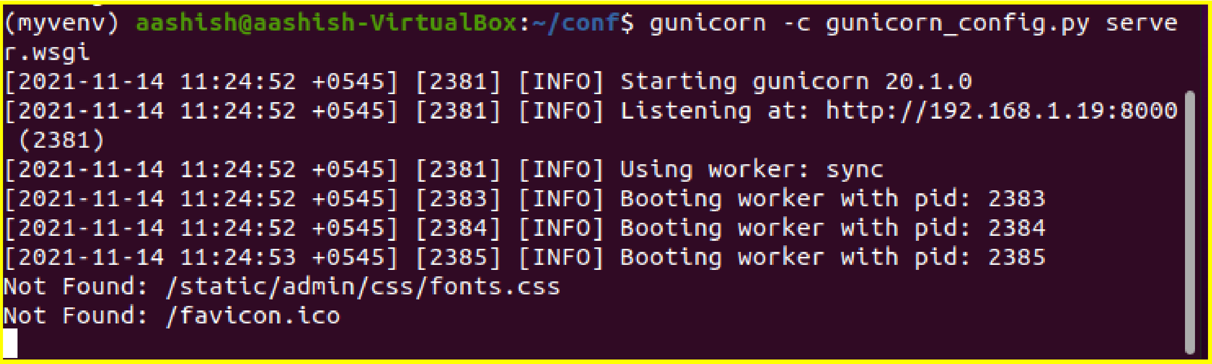
Then, we edit the **settings.py** file which is located at path **/home/aashish/server/server.** We add the host’s IP address on allowed_hosts array as follows;



```
  GNU nano 4.8                          settings.py

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-p)$@zf9xm#0wbu*#xfe$f($bf0fa9@0f%1%-5v*fwl0(o$_*^>

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ['192.168.1.19']


# Application definition
```

Now, we deploy the app on Gunicorn server using following command;

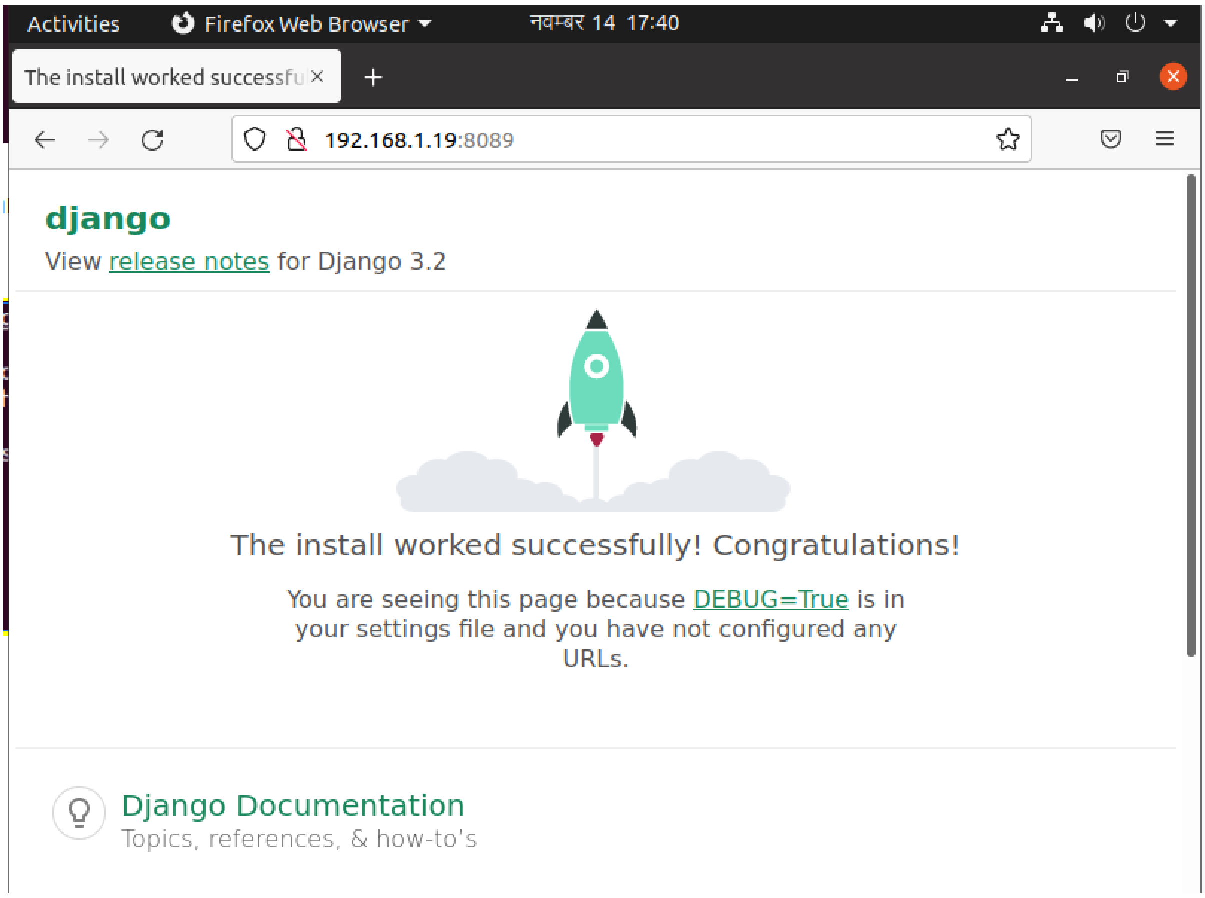
- **gunicorn - c gunicorn_config.py server.wsgi**



```
(myvenv) aashish@aashish-VirtualBox:~/conf$ gunicorn -c gunicorn_config.py serve
r.wsgi
[2021-11-14 11:24:52 +0545] [2381] [INFO] Starting gunicorn 20.1.0
[2021-11-14 11:24:52 +0545] [2381] [INFO] Listening at: http://192.168.1.19:8000
 (2381)
[2021-11-14 11:24:52 +0545] [2381] [INFO] Using worker: sync
[2021-11-14 11:24:52 +0545] [2383] [INFO] Booting worker with pid: 2383
[2021-11-14 11:24:52 +0545] [2384] [INFO] Booting worker with pid: 2384
[2021-11-14 11:24:53 +0545] [2385] [INFO] Booting worker with pid: 2385
Not Found: /static/admin/css/fonts.css
Not Found: /favicon.ico
```

Since, we have to deploy the 3 instances(workers) of the application using Gunicorn in 8089 port.
We changed the port from **8000** to **8089** in the **gunicorn_config.py** file.

Lastly, to test the app we enter **192.168.1.19:8089** on the web browser.



django

View release notes for Django 3.2

The install worked successfully! Congratulations!

You are seeing this page because DEBUG=True is in your settings file and you have not configured any URLs.

Django Documentation

Topics, references, & how-to's

Hence the app is deployed and running successfully on **Gunicorn server** port **8089**.

Thank You.