Python Deployment Instructions:

***** These instructions are for Ubuntu v18.04 *****
Follow the platform up until you go to log into the server, make sure you install Ubuntu 18.04, not 16.04. Once you try to connect with SSH, follow these instructions.

If when connecting you get this error:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0664 for 'test.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "test.pem": bad permissions
ubuntu@ec2-13-56-78-106.us-west-1.compute.amazonaws.com: Permission denied (publickey).
➜  notouchy
➜  notouchy sudo chmod 0600 test.pem
```

Execute the following to fix:

```
sudo chmod 400 {{name of the pem file}}.pem
```

yes the image shows “chmod 0600” not 400 - both will work, 400 is just a higher restriction

Once you are inside the ubuntu server:

```
sudo apt-get update
sudo apt-get install nginx
```

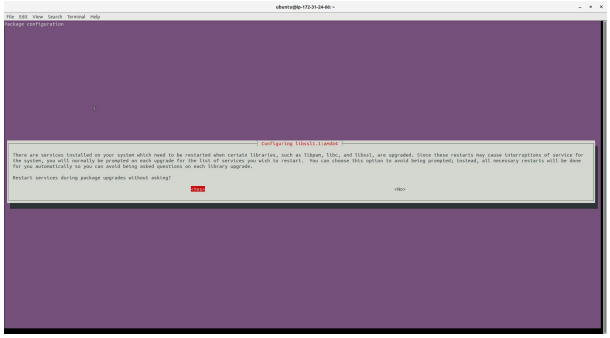
Clone repo (!! NO SUDO !!):

```
git clone {{ repo link }}
```

Install Python3 Virtual Environment:

```
sudo apt-get install python3-venv
```

Select <YES> when prompted on this screen



Change Directory to repo name:

```
cd {{reponame}}
```

Create Environment:

```
python3 -m venv venv
```

Activate Environment:

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

Install Dependencies:

```
pip install django==1.10
pip install bcrypt
```

Install Gunicorn:

```
pip install gunicorn
```

Navigate to settings.py in project folder and run:

```
sudo vim settings.py
```

Edit settings.py

```
# inside settings.py
# modify these lines
DEBUG = False
ALLOWED_HOSTS = [ '{{yourEC2.public.ip}}' ]    # keep the quotes!

# add this line at the bottom - do not delete or modify the STATIC_URL line
STATIC_ROOT = os.path.join(BASE_DIR, "static/")
```

Navigate to manage.py file:

python manage.py collectstatic
python manage.py makemigrations
python manage.py migrate

Test Gunicorn:

gunicorn {{project_name}}.wsgi

Should see the following:

```
(venv) ubuntu@ip-172-31-24-66:~/dj_ninja_gold$ gunicorn ninja_gold.wsgi
[2019-06-21 00:13:54 +0000] [2253] [INFO] Starting gunicorn 19.9.0
[2019-06-21 00:13:54 +0000] [2253] [INFO] Listening at: http://127.0.0.1:8000 (2253)
[2019-06-21 00:13:54 +0000] [2253] [INFO] Using worker: sync
[2019-06-21 00:13:54 +0000] [2256] [INFO] Booting worker with pid: 2256
^C[2019-06-21 00:13:59 +0000] [2253] [INFO] Handling signal: int
[2019-06-21 00:13:59 +0000] [2253] [INFO] Shutting down: Master
```

Hit CTRL-C to shut it down

Do not worry about turning off the environment, it can be left on throughout this process

Create service file:

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

Put the following in that file - note: you will need to use vim for this - platform has info on vim:

[Unit]
Description=gunicorn daemon
After=network.target
[Service]
User=ubuntu
Group=www-data
WorkingDirectory=/home/ubuntu/{{myRepoName}}
ExecStart=/home/ubuntu/{{myRepoName}}/venv/bin/gunicorn --workers 3 --bind unix:/home/ubuntu/{{myRepoName}}/{{projectName}}.sock {{projectName}}.wsgi:application
[Install]

WantedBy=multi-user.target

IMPORTANT! Look at the file path on ExecStart -- this path MUST be accurate to the location of your Virtual Environment bin folder -- then double check the unix:/home… path to make sure that is the path to your repo. The {{projectName}}.sock will create a file in that location and we will need this file to reference in our NGINX setup later.

Run the following commands:

```
sudo systemctl daemon-reload
sudo systemctl restart gunicorn
sudo systemctl status gunicorn
```

If there is a GREEN dot next to gunicorn.service and it says active running like this:

```
(venv) ubuntu@ip-172-31-19-68:~/dj_ninja_gold$ sudo systemctl status gunicorn
● gunicorn.service - gunicorn daemon
   Loaded: loaded (/etc/systemd/system/gunicorn.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2019-06-20 20:52:26 UTC; 5s ago
 Main PID: 3907 (gunicorn)
    Tasks: 4 (limit: 1152)
   CGroup: /system.slice/gunicorn.service
           ├─3907 /home/ubuntu/venv/bin/python3 /home/ubuntu/venv/bin/gunicorn --workers 3 --bind unix:/home/ubuntu/dj_ninja_gold/ninja_gold.sock ninja_gold.wsgi:application
           ├─3925 /home/ubuntu/venv/bin/python3 /home/ubuntu/venv/bin/gunicorn --workers 3 --bind unix:/home/ubuntu/dj_ninja_gold/ninja_gold.sock ninja_gold.wsgi:application
           ├─3927 /home/ubuntu/venv/bin/python3 /home/ubuntu/venv/bin/gunicorn --workers 3 --bind unix:/home/ubuntu/dj_ninja_gold/ninja_gold.sock ninja_gold.wsgi:application
           └─3929 /home/ubuntu/venv/bin/python3 /home/ubuntu/venv/bin/gunicorn --workers 3 --bind unix:/home/ubuntu/dj_ninja_gold/ninja_gold.sock ninja_gold.wsgi:application

Jun 20 20:52:26 ip-172-31-19-68 systemd[1]: Started gunicorn daemon.
Jun 20 20:52:26 ip-172-31-19-68 gunicorn[3907]: [2019-06-20 20:52:26 +0000] [3907] [INFO] Starting gunicorn 19.9.0
Jun 20 20:52:26 ip-172-31-19-68 gunicorn[3907]: [2019-06-20 20:52:26 +0000] [3907] [INFO] Listening at: unix:/home/ubuntu/dj_ninja_gold/ninja_gold.sock (3907)
Jun 20 20:52:26 ip-172-31-19-68 gunicorn[3907]: [2019-06-20 20:52:26 +0000] [3907] [INFO] Using worker: sync
Jun 20 20:52:26 ip-172-31-19-68 gunicorn[3907]: [2019-06-20 20:52:26 +0000] [3925] [INFO] Booting worker with pid: 3925
Jun 20 20:52:27 ip-172-31-19-68 gunicorn[3907]: [2019-06-20 20:52:27 +0000] [3927] [INFO] Booting worker with pid: 3927
Jun 20 20:52:27 ip-172-31-19-68 gunicorn[3907]: [2019-06-20 20:52:27 +0000] [3929] [INFO] Booting worker with pid: 3929
(venv) ubuntu@ip-172-31-19-68:~/dj_ninja_gold$ l
apps/  db.sqlite3  manage.py*  ninja_gold/  ninja_gold.sock=
```

Then check to see if you have the lines “Booting worker with pid: ….” for the last 3 lines like above. You will also then have a “{{project_name}}.sock” file in your repository.

Debug info:
There is a possibility that you have a green dot but have the following output:

```
(venv) ubuntu@ip-172-31-24-66:~/dj_ninja_gold$ sudo systemctl status gunicorn
● gunicorn.service - gunicorn daemon
   Loaded: loaded (/etc/systemd/system/gunicorn.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2019-06-21 00:23:29 UTC; 2s ago
 Main PID: 2391 (gunicorn)
    Tasks: 1 (limit: 1152)
   CGroup: /system.slice/gunicorn.service
           └─2391 /home/ubuntu/venv/bin/python3 /home/ubuntu/venv/bin/gunicorn --workers 3 --bind unix:/home/ubuntu/dj_ninja_gold/ninja_gold.sock ninja_gold.wsgi:application

Jun 21 00:23:29 ip-172-31-24-66 systemd[1]: Started gunicorn daemon.
Jun 21 00:23:30 ip-172-31-24-66 gunicorn[2391]: [2019-06-21 00:23:30 +0000] [2391] [INFO] Starting gunicorn 19.9.0
Jun 21 00:23:30 ip-172-31-24-66 gunicorn[2391]: [2019-06-21 00:23:30 +0000] [2391] [ERROR] Retrying in 1 second.
Jun 21 00:23:31 ip-172-31-24-66 gunicorn[2391]: [2019-06-21 00:23:31 +0000] [2391] [ERROR] Retrying in 1 second.
Jun 21 00:23:32 ip-172-31-24-66 gunicorn[2391]: [2019-06-21 00:23:32 +0000] [2391] [ERROR] Retrying in 1 second.
(venv) ubuntu@ip-172-31-24-66:~/dj_ninja_gold$
```

Most likely you cloned your repo with sudo or you created your virtual env with sudo in which case you should terminate the server and start over unless you are comfortable with CHMOD commands or deleting the repo and env and reinstalling them.

If you see something like this:

```
(venv) ubuntu@ip-172-31-24-66:~/dj_ninja_gold$ sudo systemctl status gunicorn
● gunicorn.service - gunicorn daemon
   Loaded: loaded (/etc/systemd/system/gunicorn.service; enabled; vendor preset: enabled)
   Active: failed (Result: exit-code) since Fri 2019-06-21 00:22:35 UTC; 1s ago
  Process: 2339 ExecStart=/home/ubuntu/dj_ninja_gold/venv/bin/gunicorn --workers 3 --bind unix:/home/ubuntu/dj_ninja_gold/ninja_gold.sock ninja_gold.wsgi:application (code=exited, status=203/EXEC)
 Main PID: 2339 (code=exited, status=203/EXEC)

Jun 21 00:22:35 ip-172-31-24-66 systemd[1]: Started gunicorn daemon.
Jun 21 00:22:35 ip-172-31-24-66 systemd[2339]: gunicorn.service: Failed to execute command: No such file or directory
Jun 21 00:22:35 ip-172-31-24-66 systemd[2339]: gunicorn.service: Failed at step EXEC spawning /home/ubuntu/dj_ninja_gold/venv/bin/gunicorn: No such file or directory
Jun 21 00:22:35 ip-172-31-24-66 systemd[1]: gunicorn.service: Main process exited, code=exited, status=203/EXEC
Jun 21 00:22:35 ip-172-31-24-66 systemd[1]: gunicorn.service: Failed with result 'exit-code'.
```

You will need to read the error message and try to figure out what went wrong by reading the message. For example in this example error message, the path to the venv folder is incorrect. After you make the changes necessary to fix your specific error, test it again by running:

sudo systemctl daemon-reload
sudo systemctl restart gunicorn
sudo systemctl status gunicorn

Common bug: It is essential that both Gunicorn and Django be installed in the same environment, so double check env variables if you have a red dot

Repeat the gunicorn steps until you see a green light and you see a sock file like this:

```
(venv) ubuntu@ip-172-31-24-66:~/dj_ninja_gold$ ls -al
total 28
drwxrwxr-x 5 ubuntu ubuntu   4096 Jun 21 02:06 .
drwxr-xr-x 8 ubuntu ubuntu   4096 Jun 21 01:56 ..
drwxrwxr-x 8 ubuntu ubuntu   4096 Jun 21 01:49 .git
-rw-rw-r-- 1 ubuntu ubuntu     28 Jun 21 01:49 .gitignore
drwxrwxr-x 4 ubuntu ubuntu   4096 Jun 21 02:06 apps
-rwxrwxr-x 1 ubuntu ubuntu    808 Jun 21 01:49 manage.py
drwxrwxr-x 3 ubuntu ubuntu   4096 Jun 21 02:06 ninja_gold
srwxrwxrwx 1 ubuntu www-data    0 Jun 21 02:06 ninja_gold.sock
```

At this point, return to the platform and read the NGINX instructions