Installing Nginx

sudo apt update

sudo apt install nginx

Checking Nginx Status

systemctl status nginx

A blue screen with white text

Description automatically generated with medium confidence

Managing Nginx Process (additional information)

sudo systemctl stop nginx

To stop your web server

sudo systemctl start nginx

To start your web server

sudo systemctl restart nginx

To restart your web server

sudo systemctl reload nginx

If you are only making configuration changes, Nginx can often reload without dropping connections

sudo systemctl disable nginx

By default, Nginx is configured to start automatically when the server boots. If this is not what you want, you can disable this behavior

sudo systemctl enable nginx

To re-enable the service to start up at boot

Setting Up Server Frontend

sudo nano /etc/nginx/sites-available/frontend

add following lines

server {

listen 80;

listen [::]:80;

root /var/www/sample/html/build;

index index.html index.htm index.nginx-debian.html;

server\_name sample 3.230.228.45;

location / {

**try\_files $uri $uri/ /index.html?q=$uri&$args;**

}

}

Setting Up Server Backend

sudo nano /etc/nginx/sites-available/backend

eg: we can change your\_domain to Data\_Recon\_Main

sudo nano /etc/nginx/sites-available/Data\_Recon\_Main

add the following contents :

server {

listen 80;

server\_name 172.31.81.4;

location / {

proxy\_pass http://172.31.81.4:8000;

}

}

You can replace the ip address with your actual domain address such as [www.test](http://www.test)

sudo ln -s /etc/nginx/sites-available/your\_domain /etc/nginx/sites-enabled/

this creates a symbolic link to the /etc/nginx/sites-enabled folder

sudo nano /etc/nginx/nginx.conf

Find the server\_names\_hash\_bucket\_size directive and remove the # symbol to uncomment the line.

sudo nginx -t

to test to make sure that there are no syntax errors in any of your Nginx files

sudo systemctl restart nginx

Installing Gunicorn

First Create a virtual environment for the project

Sudo apt install python3-venv

python3 -m venv django-env

this will create a folder named django-env, this can be any name in the current directory

source django-env/bin/activate

pip install -r requirements.txt

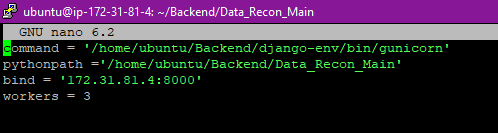
pip install gunicorn

create a directory and name it conf

mkdir conf

sudo nano conf/gunicorn\_conf.py

add the following lines to the file



Bind to the your IP address and port

Give the correct path to gunicorn for command

Give the correct path to your project in pythonpath

gunicorn -c conf/gunicorn\_conf.py Data\_Recon\_Main.wsgi –timeout 1200

this will run the project with the ip and port specified in the gunicorn\_conf file

replace Data\_Recon\_Main with your project name

reference:

1. <https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-20-04>
2. <https://youtu.be/YnrgBeIRtvo>
3. <https://realpython.com/django-nginx-gunicorn/>

A screenshot of a computer screen

Description automatically generated with medium confidence

If Connection already in use while running gunicorn:

**sudo fuser -k 8000/tcp**

for creating new screen

**screen -S Backend**

now execute the runserver command (gunicorn -c conf/conf.py Data\_Recon\_Main.wsgi)

and then press **Ctrl + AD**

for killing a screen

**screen -X -S [session # you want to kill] quit**

**For SSL**

<https://serverspace.io/support/help/ssl-certificate-on-nginx-ubuntu-20-04/>

<https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-nginx-on-centos-7>

/etc/nginx/sites-available/Data\_Recon\_Backend

server {

#listen 80 default\_server;

#listen [::]:80 default\_server ipv6only=on;

listen 443 ssl;

server\_name psyber.io;

ssl\_certificate /etc/letsencrypt/live/example.com/fullchain.pem;

ssl\_certificate\_key /etc/letsencrypt/live/example.com/privkey.pem;

……….

}

for redirect

server {

listen 80;

server\_name example.com;

return 301 https://$host$request\_uri;

}

**sed -i -e 's/\r$//' StartAdapter.sh**