**PASOS PARA EL DESPLIEGUE DE DJANGO A PRODUCCION EMPLEANDO AMAZON SERVER EC2**

1. sudo ufw app list

sudo ufw allow OpenSSH

sudo ufw enable

sudo ufw status (para verificar que se activen)

Nota: Esto es para admitir conexiones SSH maquina virtual puerto 22

1. sudo apt-get update

sudo apt-get upgrade

Nota: Esto es para descargar actualizaciones sistema operativo Linux)

1. sudo apt install python3-venv nginx (**Requerimiento**)
2. En este paso es clonar nuestro repositorio (Una vez clonado con el comando **ls** verificamos la clonación.
3. python3 -m venv my\_env (Creación de entorno virtual, con el comando **ls** lo verificamos)
4. source my\_env/bin/activate (Acceder al entorno virtual)
5. pip install django gunicorn (Requerimiento)
6. cd MonsterAPP (Para movernos a nuestro proyecto, en este caso el repositorio MonsterAPP)
7. python manage.py migrate (aqui saldrá error porque habría que instalar todos los módulos descargados en el desarrollo, por ejemplo: Jazzmin (pip install django-jazzmin), crispy\_forms (pip install django-crispy-forms), etc. Una vez se descargue todo lo necesario procedemos con el siguiente paso)
8. pip install whitenoise
9. python3 manage.py collectstatic
10. Verificamos en en settings de nuestra aplicación este la dirección IP de nuestra instancia en ALLOWED\_HOSTS. Para editar podemos usar el comando: **nano SISTEMA\_ANALITICO/settings.py**
11. sudo ufw allow 8000 (Para habilidar el puerto de desarrollo)
12. python3 manage.py runserver 0:8000
13. sudo nano /etc/systemd/system/gunicorn.socket

(Agregamos la siguiente configuración:)

[Unit]

Description=gunicorn socket

[Socket]

ListenStream= /run/gunicorn.sock

[Install]

WantedBy=sockets.target

1. sudo nano /etc/systemd/system/gunicorn.service

(Agregamos la siguiente configuración:)

[Unit]

Description= gunicorn daemon

Requires= gunicorn.socket

After= network.target

[Service]

User=ubuntu

Group=www-data

WorkingDirectory= /home/ubuntu/MonsterAPP

#ExecStart= /home/ubuntu/my\_env/bin/gunicorn --access-logfile - --workers 3 --bind unix:/run/gunicorn.sock SISTEMA\_ANALITICO.wsgi.application

ExecStart=/home/ubuntu/my\_env/bin/gunicorn \

--access-logfile - \

--workers 3 \

--bind unix:/run/gunicorn.sock \

SISTEMA\_ANALITICO.wsgi:application

[Install]

WantedBy=multi-user.target

1. sudo systemctl start gunicorn.socket (Iniciar gunicorn)
2. sudo systemctl enable gunicorn.socket (habilitar gunicorn)
3. sudo systemctl status gunicorn.socket (comprobar habilitación)
4. sudo nano /etc/nginx/sites-available/monsterapp (Configurar archivo nginx, se puede nombrar de cualquier manera, en micaso lo nombre “monsterapp”.

Agregamos la siguiente configuración (teniendo en cuenta nuestro nombre o IP del servidor):

server{

listen 80;

server\_name 18.218.94.173;

location =https://shape-host.b-cdn.net/favicon.ico?x23847 { access\_log off; log\_not\_found

off; }

location /staticfiles/ {

alias /home/ubuntu/SISTEMA\_ANALITICO/staticfiles;

}

location / {

include proxy\_params;

proxy\_pass http://unix:/run/gunicorn.sock;

}

}

1. sudo ln -s /etc/nginx/sites-available/monsterapp /etc/nginx/sites-enabled (Para habilitar archivo nginx)
2. sudo nginx -t (Para verificar correcta configuración de nginx)
3. sudo systemctl restart nginx (Reiniciar nginx)
4. sudo ufw delete allow 8000 (para deshabilitar puerto de desarrollo)
5. sudo ufw allow 'Nginx Full' (Para hacer accesible a nginx)

Notas:

Resetear gunicorn: sudo systemctl restart gunicorn

Detener servidor nginx: sudo systemctl stop nginx

Token GitHUB: ghp\_5L3WDWLUpv8APfrKzvPAUK38a368tD3AmdbU

ghp\_5L3WDWLUpv8APfrKzvPAUK38a368tD3AmdbU