**EC2 Deployment**

**Phase – 1 Launch Instance**

**Phase – 2 Inbound outbound rules**

**Phase – 3 Project Setup (use domain / if not step 4 use ip)**

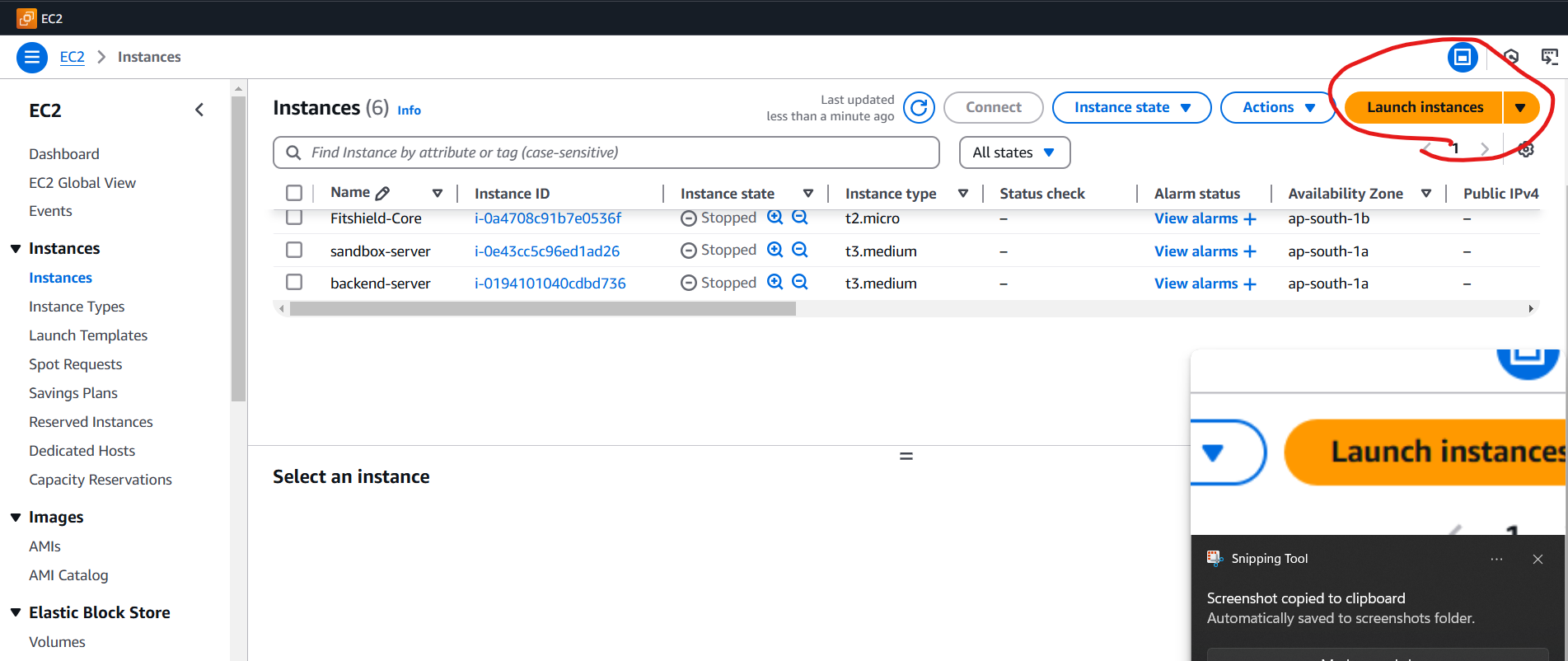
**Phase – 4 point ip to domain**

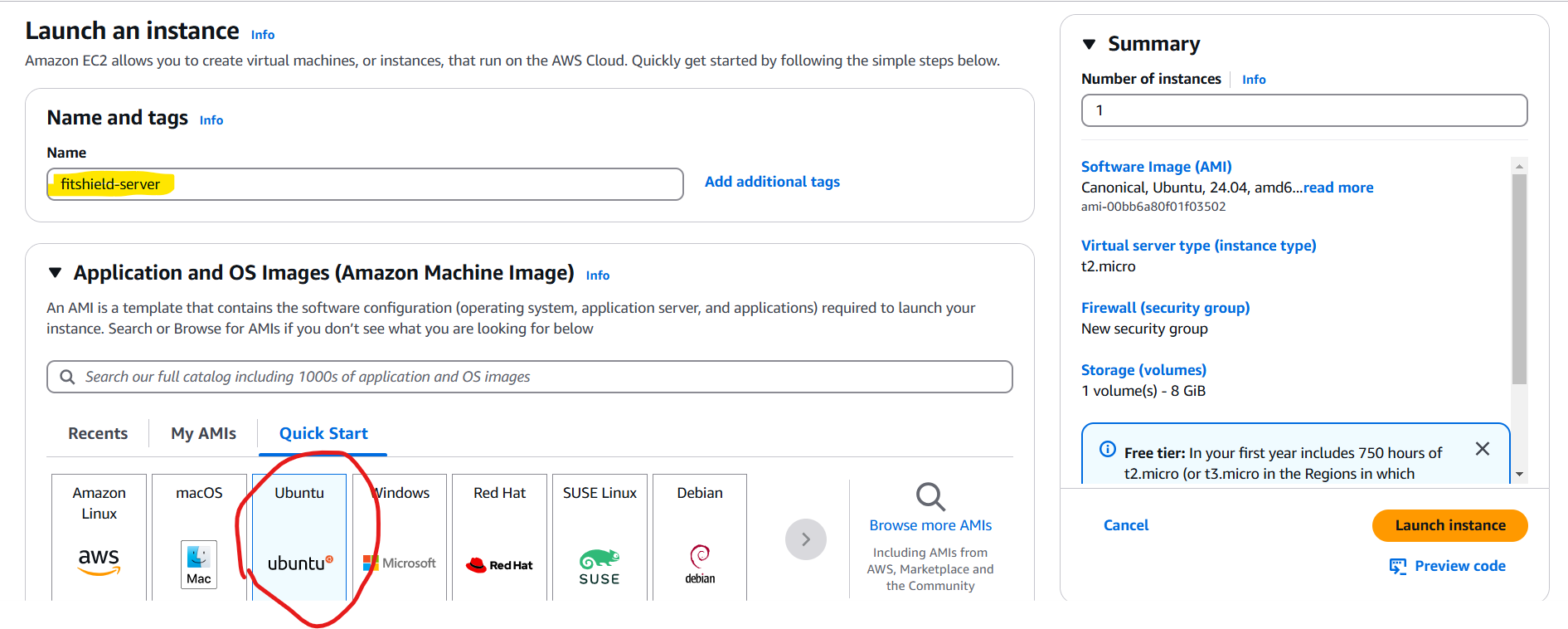
**Phase – 6 Test it on Postman**

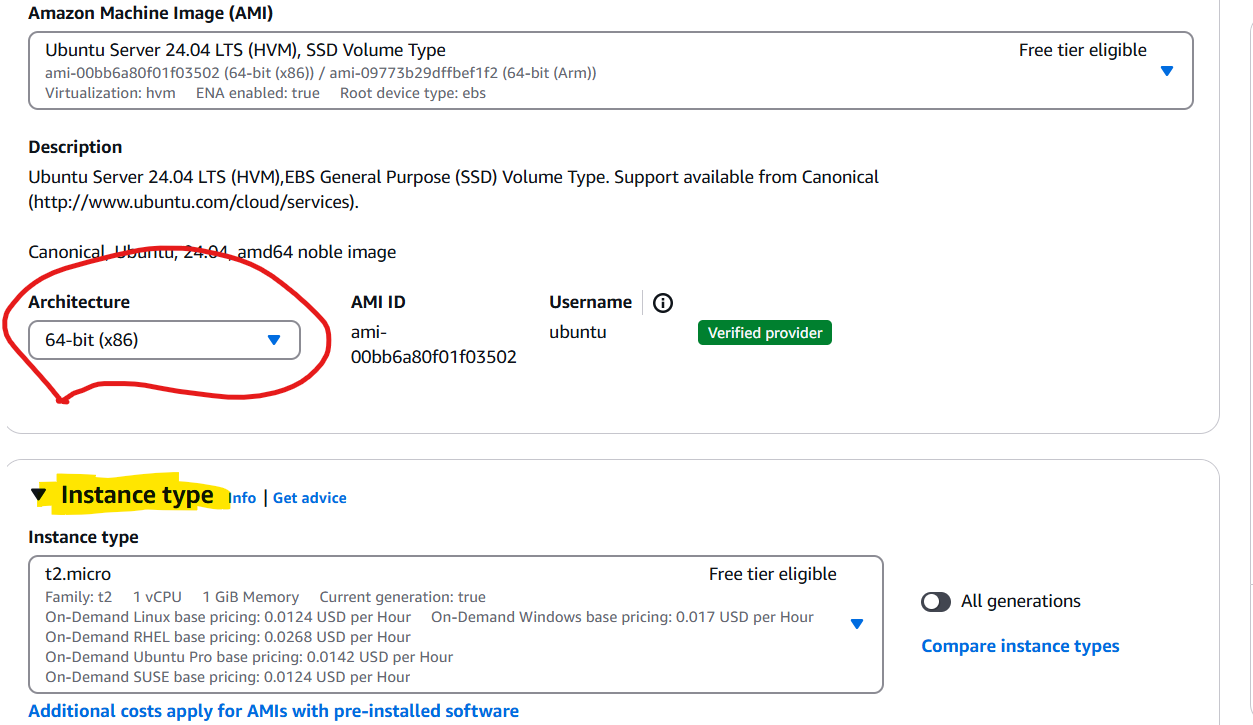
**Phase – 7 Connect with SSL for future deployment**

**Phase – 8 View all Configurations**

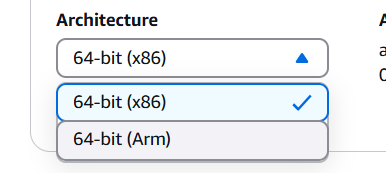
**Phase – 1 Launch Instance**

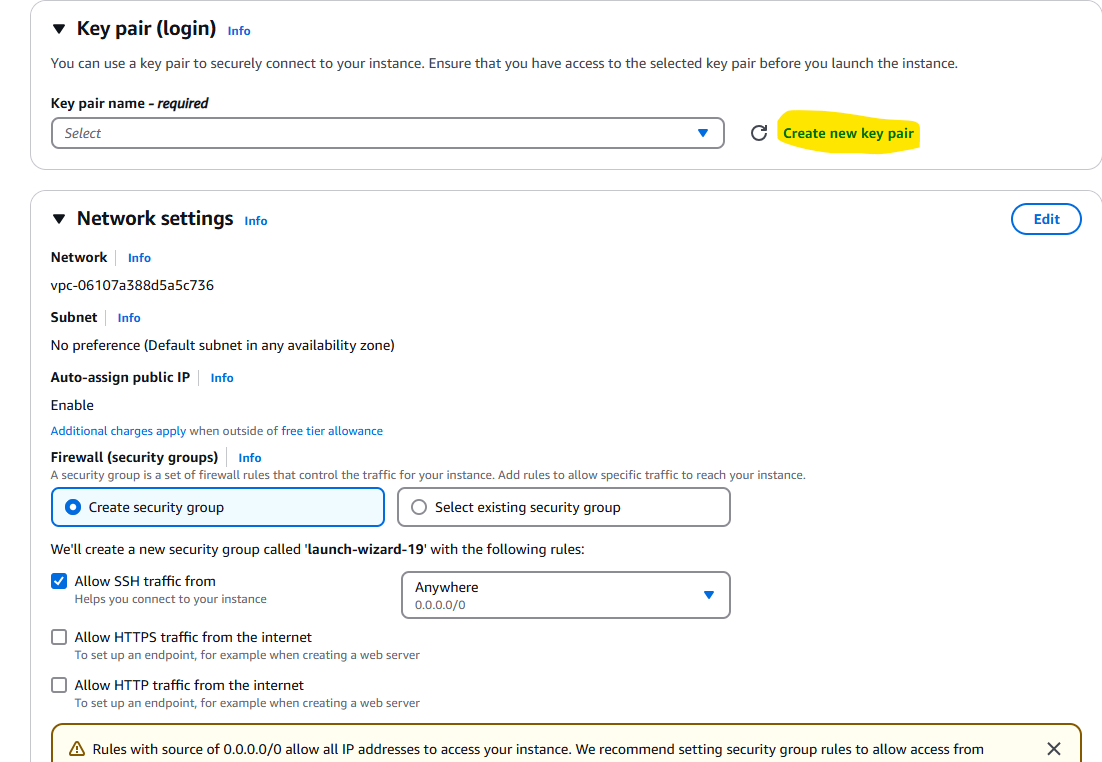
****

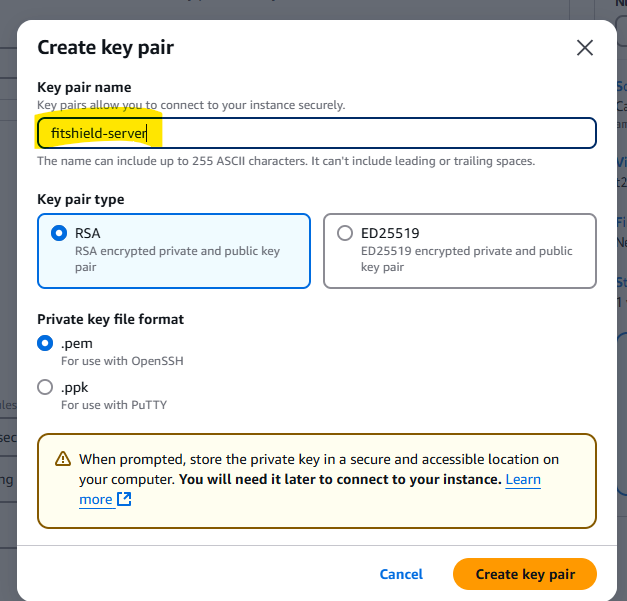
****

****

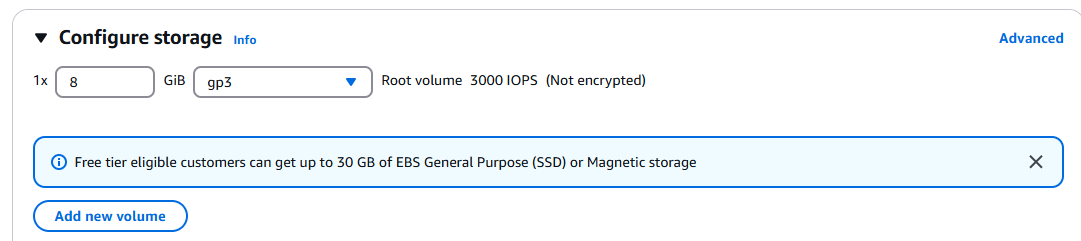
Here if architechture is 64 but few option available there  
if its Arm then t4g.medium other option will get available

****

****

****

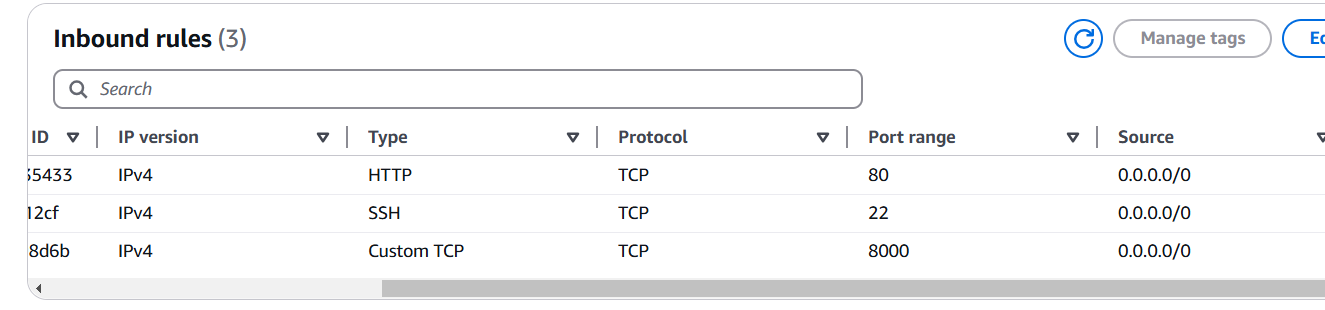
Create Key pair to access server later for future deployment direct via prompt.

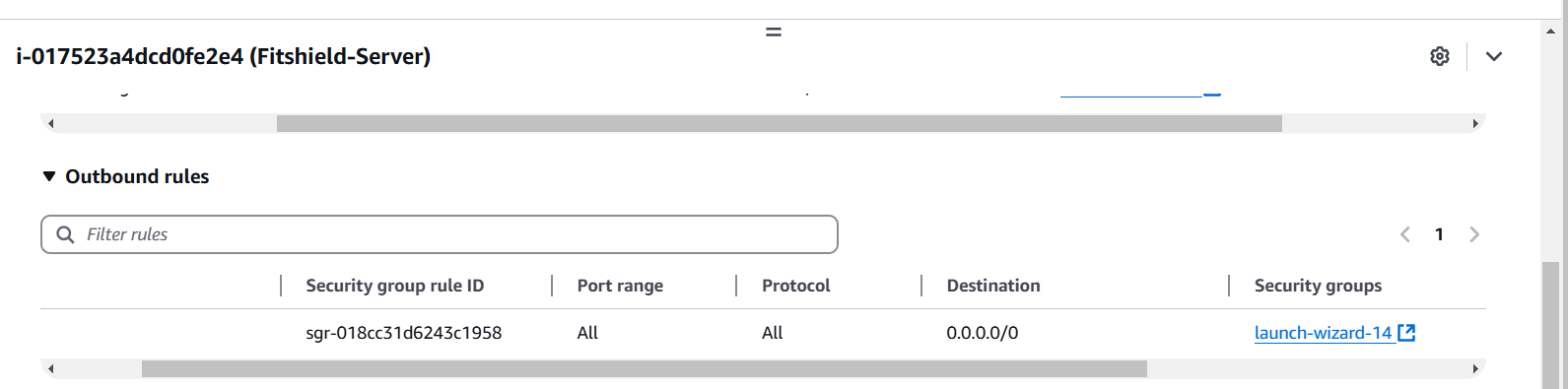
****

Total 30 gb we can use in free tiar update it if upgradation want

**Instance Launch is Done!**

**Phase – 2 Inbound outbound rules**

****

****

**Inbound Outbound rules setup Done!**

**Phase – 3 Project Setup**

**fitshield/**

**├── db.sqlite3**

**├── dishes\_output**

**├── env**

**├── fitshield/**

**│ ├── \_\_init\_\_.py**

**│ ├── \_\_pycache\_\_/**

**│ ├── asgi.py**

**│ ├── settings.py**

**│ ├── urls.py**

**│ ├── wsgi.py**

**├── fitshield\_webapp/**

**│ ├── AI/**

**│ ├── \_\_init\_\_.py**

**│ ├── \_\_pycache\_\_/**

**│ ├── admin.py**

**│ ├── apps.py**

**│ ├── migrations/**

**│ ├── models.py**

**│ ├── payment/**

**│ ├── templates/**

**│ ├── tests.py**

**│ ├── urls.py**

**│ ├── utils/**

**│ ├── view/**

**├── manage.py**

**├── requirements.txt**

**├── runtime.txt**

**└── tempCodeRunnerFile.exe**

**1. Update and Install Dependencies**

sudo apt update

sudo apt install python3-pip python3-dev nginx

sudo pip3 install virtualenv

**Check the Current Working Directory(optional)**

Pwd

Output will look like this - **/home/ubuntu/**

**2. Create a Project Directory and Virtual Environment**

**git clone https://**

make sure settings.py file contains \* or domain name not localhost else bad request 400 error will show

if there is any error in project library or code error it will show internal error

**enter username & password**

**username:** dpfitshield44 **password:** ghp\_JHtGCYVRP8zQl3WlANZjzy31Z4lDaH2JbYpq

go into project folder and create env there – cd fitshield

virtualenv env

source env/bin/activate

pip install django gunicorn

pip install -r requirements.txt

**3. Allow Port 8000 in Firewall (For Testing Django Server Directly on EC2)**

sudo ufw allow 8000

**4. Run Django Server for Testing (Optional)**

python manage.py runserver 0.0.0.0:8000

Stop the server using Ctrl + C after testing.

**5. Run Gunicorn to Serve Django App**

gunicorn --bind 0.0.0.0:8000 fitshield.wsgi:application

Stop the server using Ctrl + C after testing.

MOST IMP TO DEACTIVATE!!!!

**deactivate**

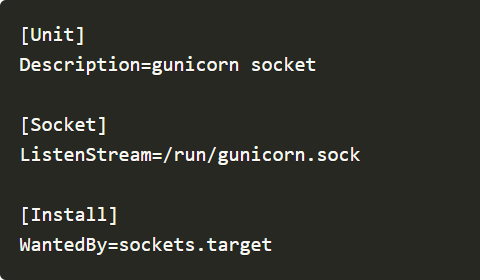
**Where You Do NOT Need to Deactivate? (Optional)**

* While setting up Gunicorn (gunicorn.service file).
* While configuring Nginx.
* While setting up the systemd service (gunicorn.socket).

**6. Let's create a system socket file for gunicorn now**

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

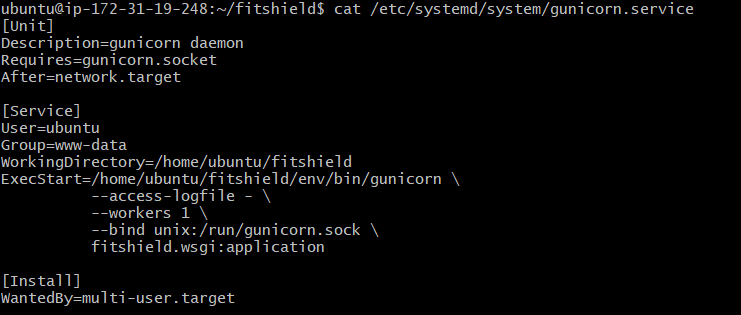
**Add the following content:**



**7. Create Gunicorn Service File**

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

**Add the following content:**



**8. Start and Enable Gunicorn (copy both together)**

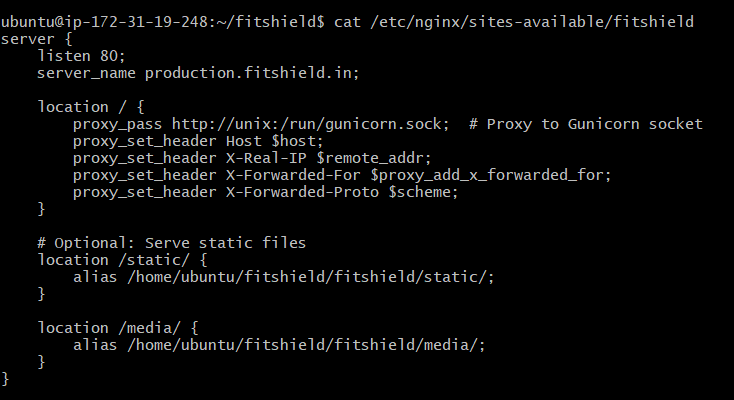
sudo systemctl start gunicorn.socket

sudo systemctl enable gunicorn.socket

**Configuring Nginx as a reverse proxy**

**9. Create Nginx Configuration for Django**

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



**10. Enable Nginx Site and Restart Nginx**

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

sudo systemctl restart nginx

**You are good to go !! but before that check it once and restart everything**

**Remove default one if exist**

**sudo rm default**

**ls**

**Restart Everything**

sudo service nginx restart

sudo service gunicorn restart

**11. Allow HTTP and HTTPS Traffic in Firewall**

sudo ufw allow 'Nginx Full'

sudo systemctl restart nginx

**12. Enable Gunicorn to Start on Boot**

sudo systemctl daemon-reload

sudo systemctl restart gunicorn

sudo systemctl enable gunicorn

**13. Verify Everything is Running**

sudo systemctl status gunicorn

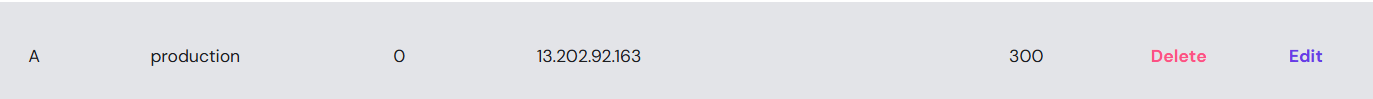
sudo systemctl status nginx

**14 . watch logs (optional)**

sudo journalctl -u gunicorn --no-pager | tail -n 20

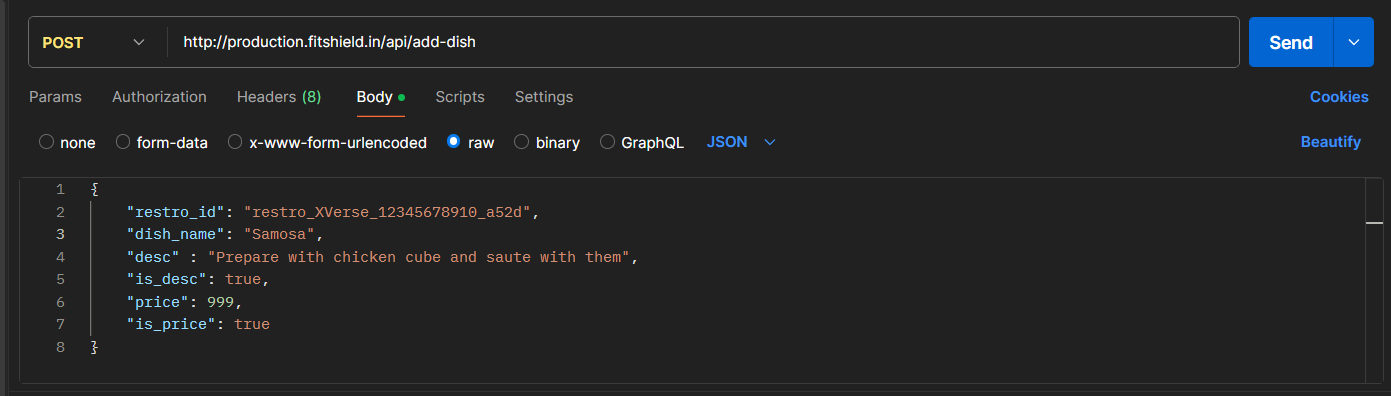
sudo journalctl -u nginx --no-pager | tail -n 20

**Phase – 4 point ip to domain – Hostinger, GoDaddy etc**

****

**Pointing to server is done!**

**Phase – 5 Test it on Postman**

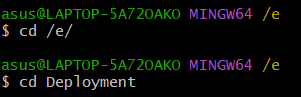
****

**Working Properly!**

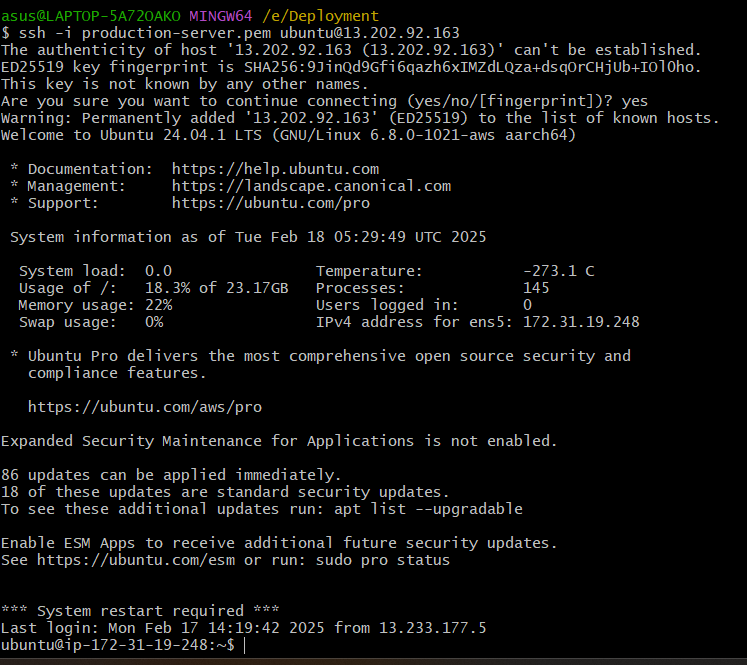
**Phase – 6 Connect with SSL for future deployment**

**1️. Connect to EC2 via SSH**

**Here make sure path is set properly , containing .pem file**

****

ssh -i FitshieldServerPair.pem [ubuntu@13.202.92.163](mailto:ubuntu@13.202.92.163)

****

While connecting to SSH if permission got denied(optional)

change permission to security advanced and remove all user and apply

icacls "E:\Deployment\FitshieldServerPair.pem" /inheritance:r

icacls "E:\Deployment\FitshieldServerPair.pem" /grant:r "%USERNAME%:R"

Change the settings.py file related to **AllowedHosts**

**Fetch the Latest Changes Without Merging**

**clear**

**ls**

**cd fitshield**

git fetch origin main

**username:** dpfitshield44 **password:** ghp\_JHtGCYVRP8zQl3WlANZjzy31Z4lDaH2JbYpq

git reset --hard origin/main

source env/bin/activate

python manage.py runserver 0.0.0.0:8000

deactivate

sudo systemctl daemon-reloadsudo systemctl start gunicorn

**exit**

**Updating Latest on Server Done!**

**Phase – 7 View all Configurations**

**1️. Connect to EC2 via SSH**

**Here make sure path is set properly , containing .pem file**

ssh -i FitshieldServerPair.pem [ubuntu@13.202.92.163](mailto:ubuntu@13.202.92.163)

**2️. View the Gunicorn Configuration Files**

**Gunicorn Socket File**

cat /etc/systemd/system/gunicorn.socket

**Gunicorn Service File**

cat /etc/systemd/system/gunicorn.service

**3️. View the Nginx Configuration File**

cat /etc/nginx/sites-available/fitshield

**If it doesn’t exist, check:**

ls /etc/nginx/sites-available/

**If you're unsure which config file is used, run:**

nginx -T | grep "server\_name"

**4️. Check Firewall & Open Ports**

sudo ufw status

sudo netstat -tulnp | grep LISTEN

**5️. Check Running Services (Gunicorn & Nginx)**

sudo systemctl status gunicorn

sudo systemctl status nginx

sudo nginx -t

**6️. View Environment Variables & Django Logs**

printenv | grep DJANGO

journalctl -u gunicorn --no-pager | tail -50

**Listed all View all Configurations!**