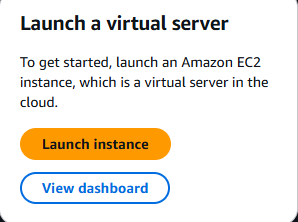
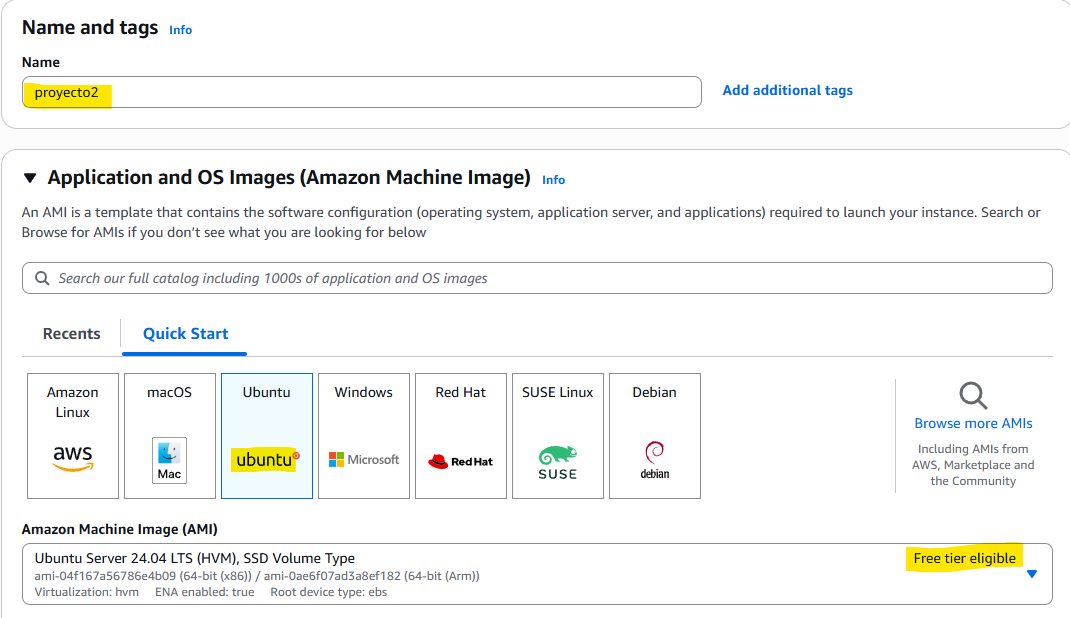


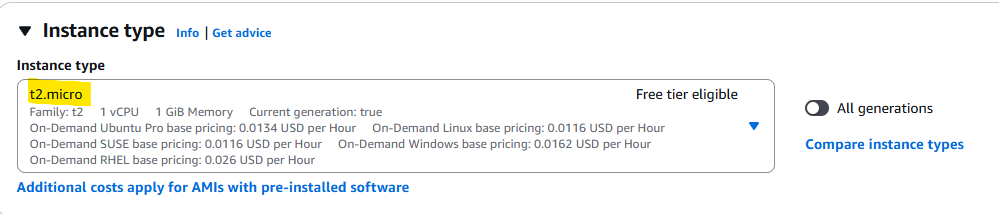
1. Entramos a EC2



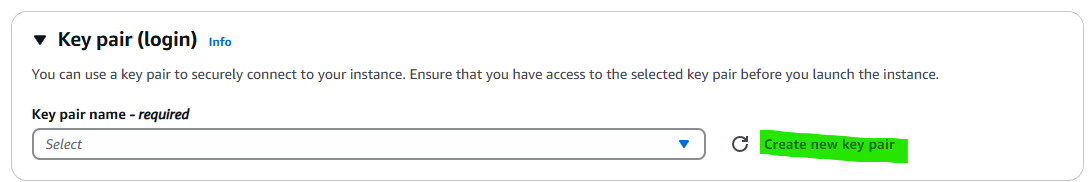
1. Launch instance



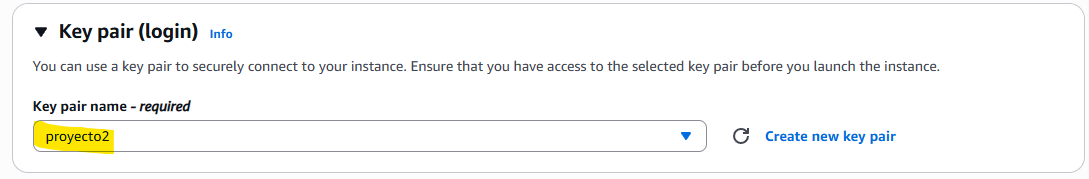
1. Elegimos el nombre, el sistema operativo de la instancia y una free tier elegible



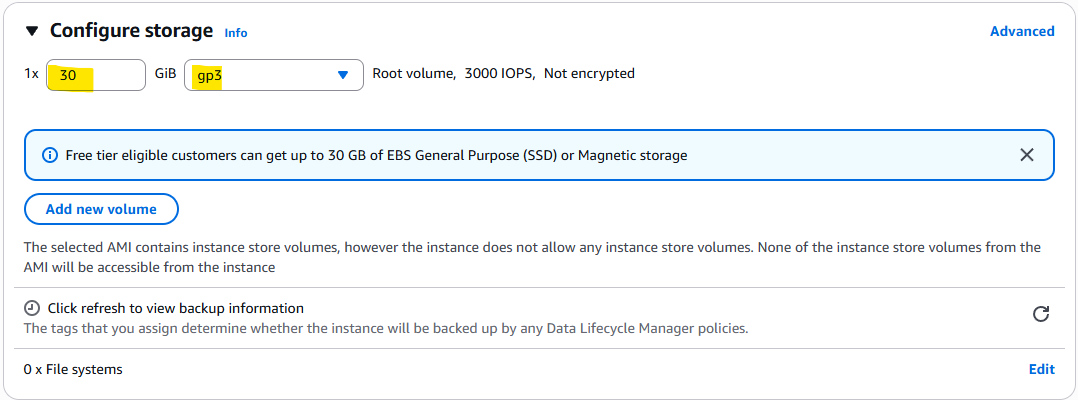
1. Elegimos el tipo de instancia gratis también



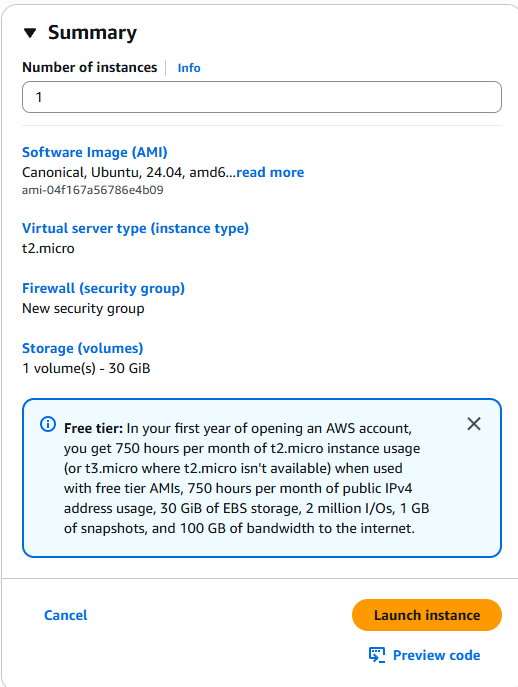
1. Creamos la llave, Guardamos el archivo .pem en una ubicación para acceder a ella después



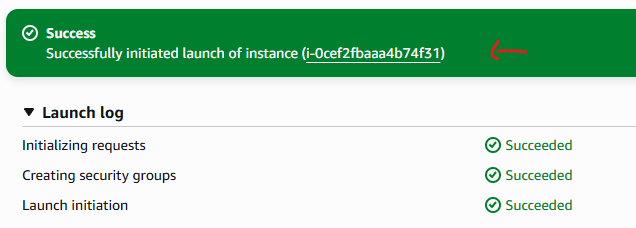
1. Seleccionamos lo máximo permitido gratuito que son 30GB y gp3 (gp3 es el tipo más reciente de volumen EBS SSD de uso general en AWS. Es una **mejora directa de gp2**, y tiene **más rendimiento y más control)**



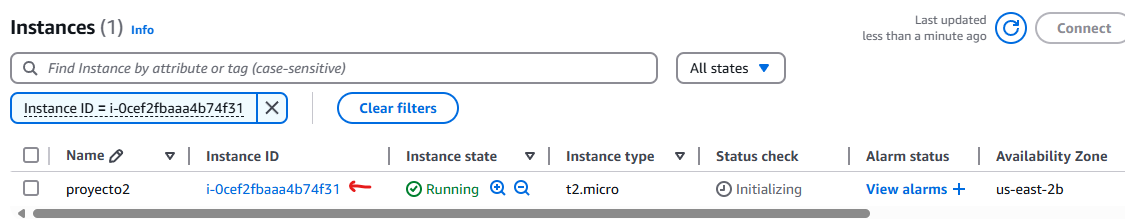
1. Creamos la instancia



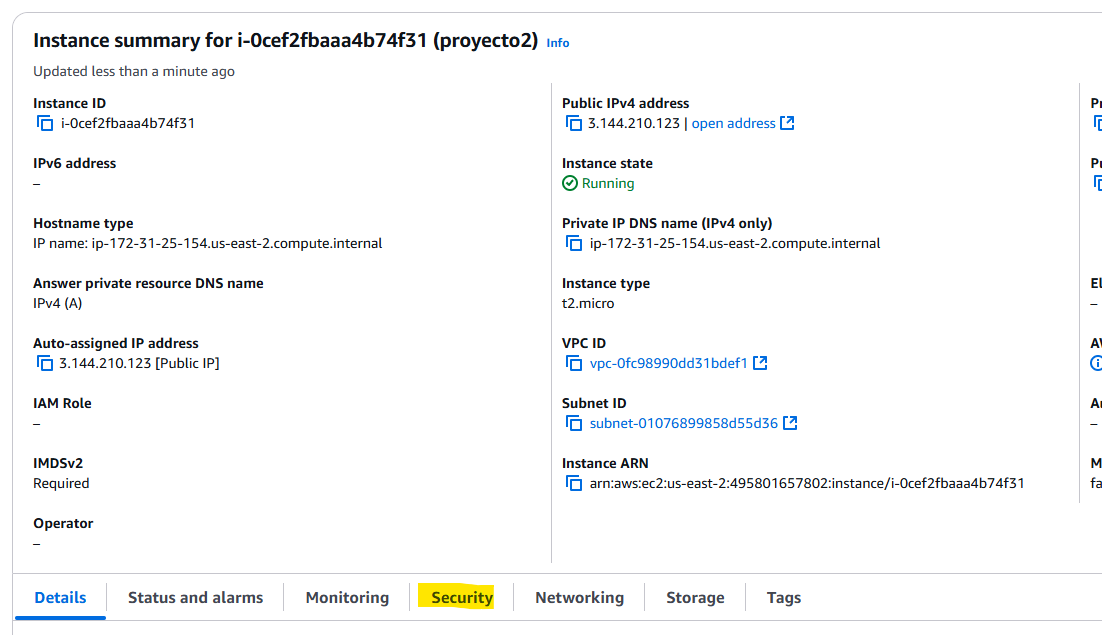
1. Launch instance



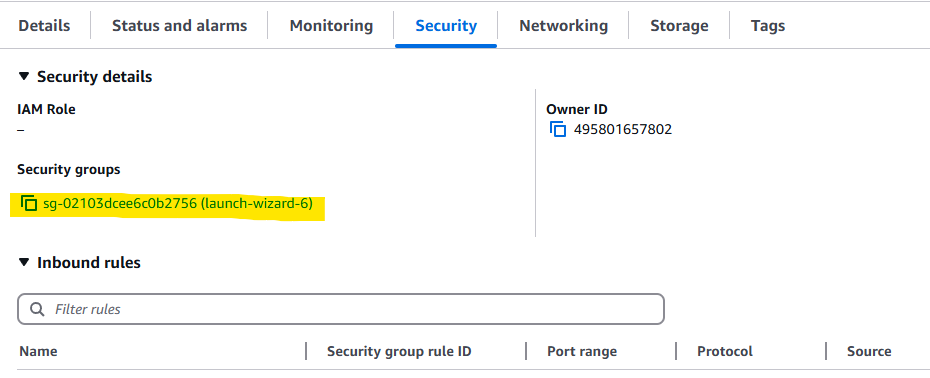
1. Click en la instancia



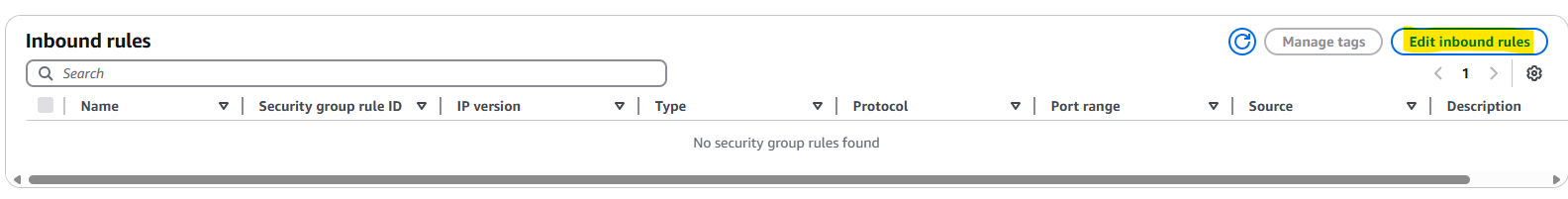
1. Click en security

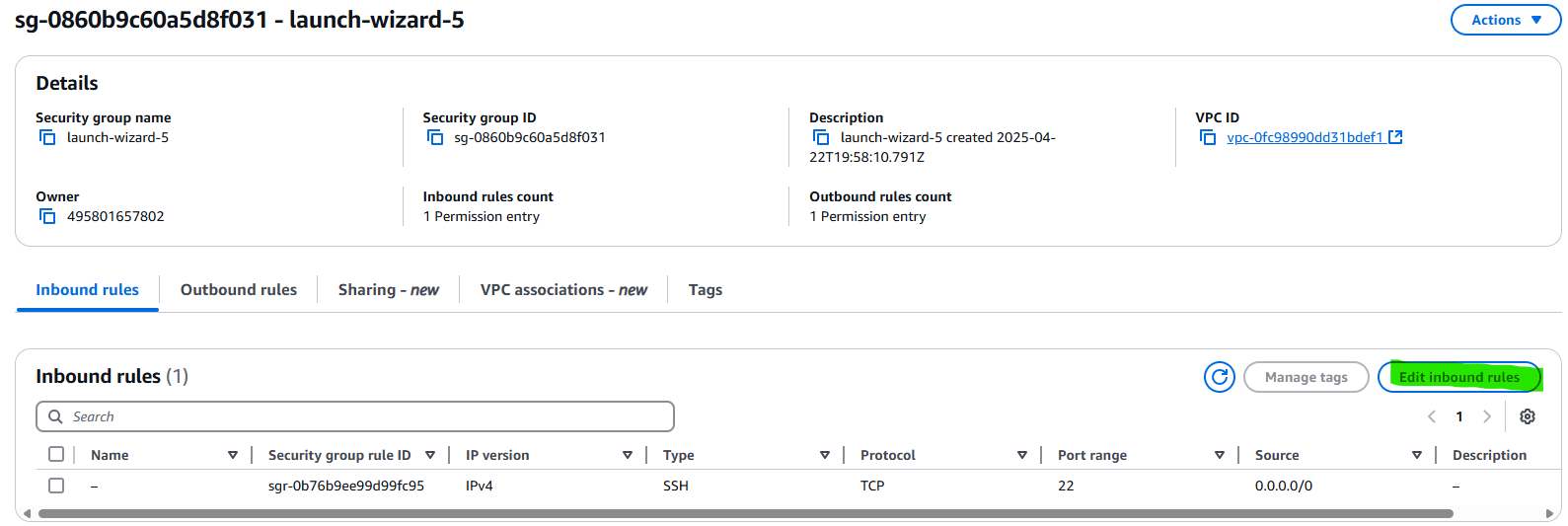


1. Security Groups

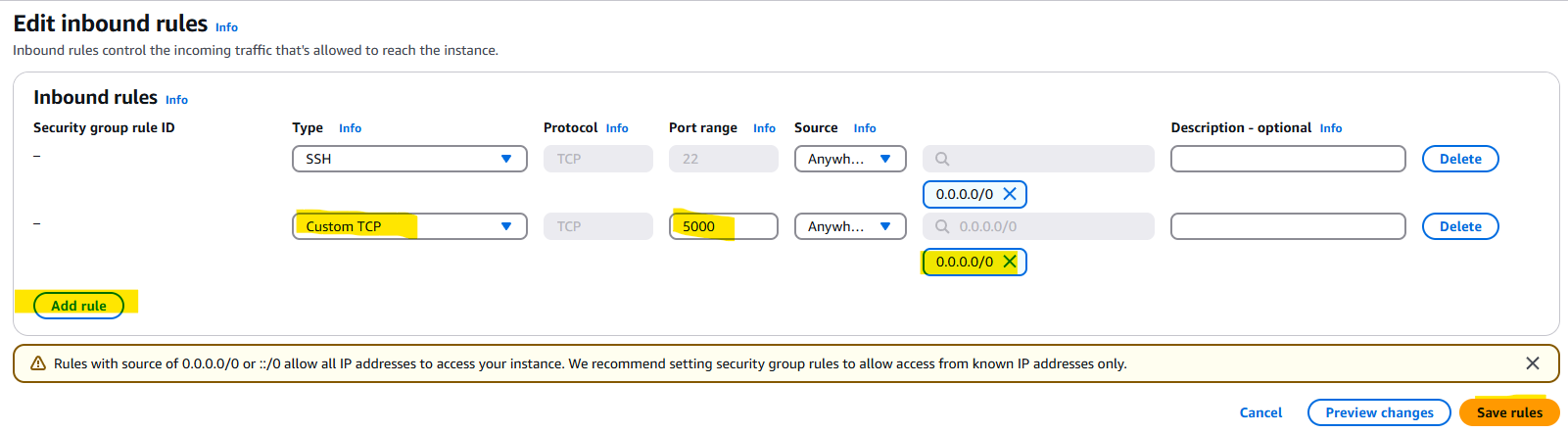


1. Editar los Inbound Rules

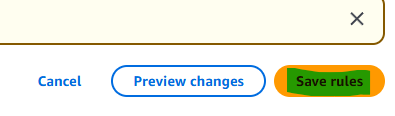




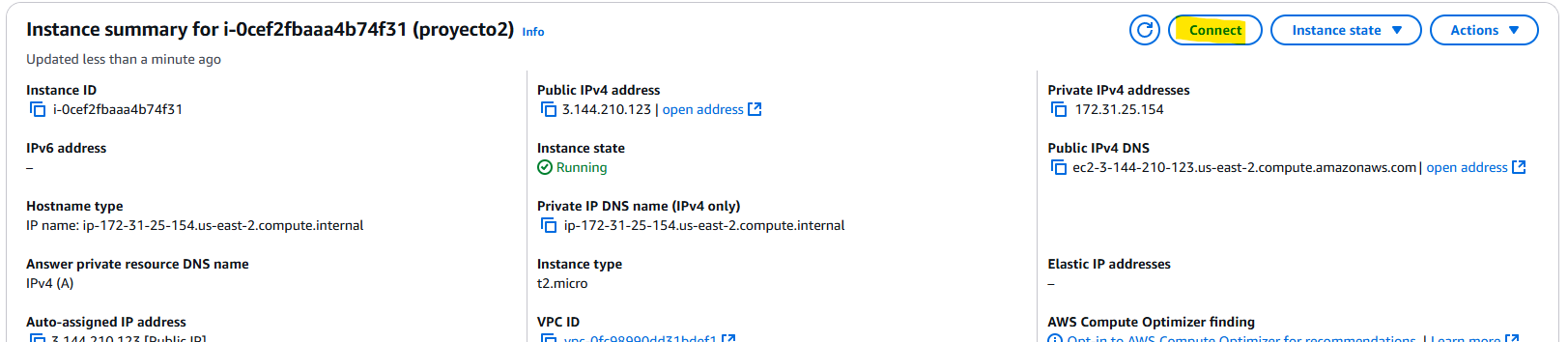
1. Add Rule y agregamos el puerto 5000 y 0.0.0.0

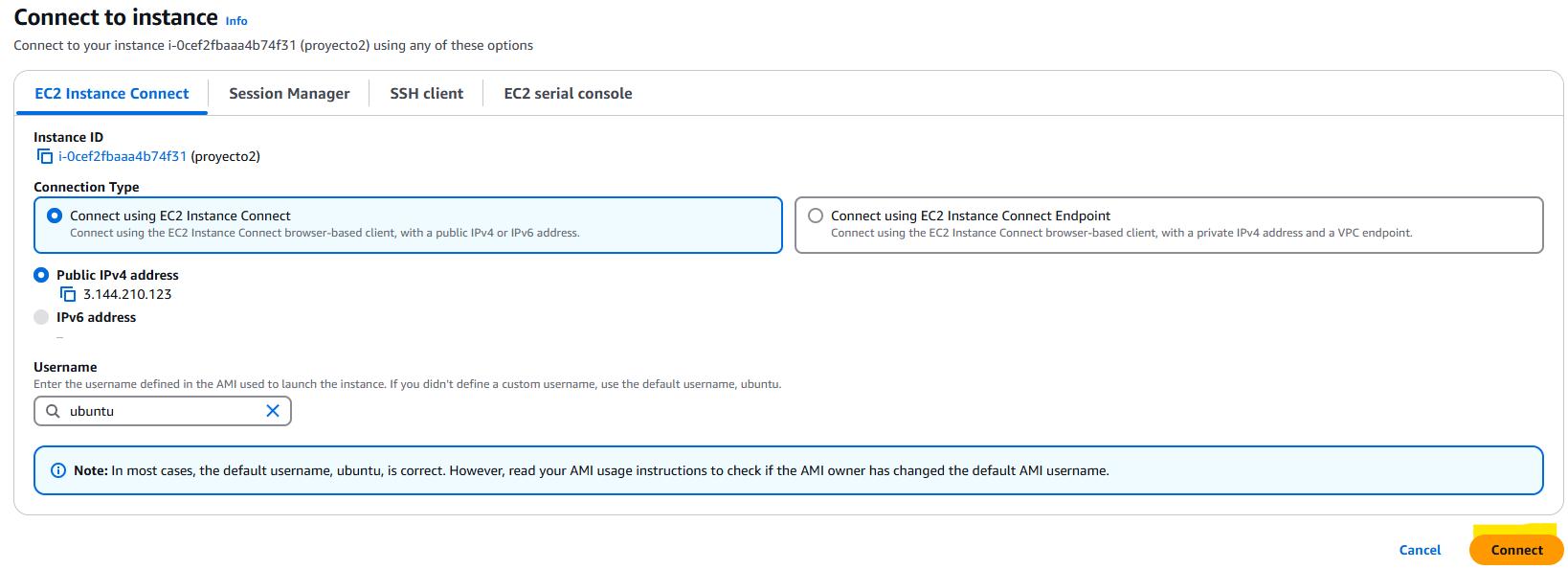


1. Save Rules, ya esta instancia permite recibir request a través del puerto 5000



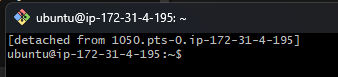
1. Click en conect para conectarse a la instancia





1. sudo apt update (para actualizer el indice de paquetes)
2. sudo apt install python3-pip
3. sudo apt install python3-venv -y (permite crear entornos virtuales)
4. python3 -m venv venv (creamos el entorno virtual)
5. source venv/bin/activate (activamos el entorno virtual)
6. pip install flask joblib scikit-learn torch pandas gunicorn
7. cd ~ (asegurarnos que estamos en home)
8. Dentro de la consola de nuestra maquina usar Git Bash:
9. cd "/c/Users/b\_o\_x/Desktop/Uniandes/Ciclo 4/Machine Learning y Procesamiento de Lenguaje Natural/Proyecto 2/AWS"
10. chmod 400 proyecto2.pem
11. ssh -i proyecto2.pem [ubuntu@3.144.210.123](mailto:ubuntu@3.144.210.123)
12. desde la consola de AWS
13. nano main.py (pegar el contenido del codigo de la api para crear el archivo main.api)
14. Guarda con Ctrl + O, luego Enter, y sal con Ctrl + X
15. nano entrenar\_modelo.py (creamos los archivos del modelo ver el archivo entrenar\_modelo)
16. Guarda con Ctrl + O, luego Enter, y sal con Ctrl + X
17. python3 entrenar\_modelo.py (para que los archivos se creen con las instancias del entorno virtual de la maquina AWS)
18. uvicorn main:app --host 0.0.0.0 --port 5000 (para ejecutar la API)
19. sudo apt install uvicorn
20. uvicorn main:app --host 0.0.0.0 --port 5000 (para ejecutar la API)
21. pip install fastapi uvicorn
22. uvicorn main:app --host 0.0.0.0 --port 5000
23. pip install flask
24. pip install flasgger
25. python main.py
26. http://3.144.210.123:5000/docs

Conectarse SSH:

1. Ubicamos Git Bash en la dirección donde se encuentra el archivo .pem
2. chmod 400 proyecto\_1.pem (cambiar permisos de archivo)
3. ssh -i proyecto\_1.pem ubuntu@<IP\_PUBLICA> (Conectarse usando SSH)
4. ssh -i proyecto\_1.pem [ubuntu@3.142.12.195](mailto:ubuntu@3.142.12.195)
5. yes (escribir yes)
6. Ya estamos dentro de nuestra instancia creada EC2, estamos dentro de nuestra maquina virtual en AWS, ya aquí podemos scribir:
7. Screen
8. source venv/bin/actívate (activar entorno virtual)
9. python3 main.py
10. Dejar la API corriendo y desconectarte de screen
11. Ahora haz lo siguiente:
12. **Presiona** Ctrl + A (manteniendo Ctrl, presiona A)
13. Luego **suelta todo** y presiona **D**
14. Esto no cierra tu API.  
    **Esto solamente te desconecta de screen**, pero tu API **sigue corriendo** en segundo plano.
15. 
16. Volver más tarde a tu sesión de screen
17. screen -r
18. http://3.142.12.195:5000/docs
19. Perfecto, si ya hiciste Ctrl + A luego D, tu API **sigue corriendo en segundo plano** dentro de la sesión screen. Ahora puedes acceder al punto final desde cualquier navegador o herramienta (como Swagger, Postman o curl).

http://3.144.210.123:5000/docs