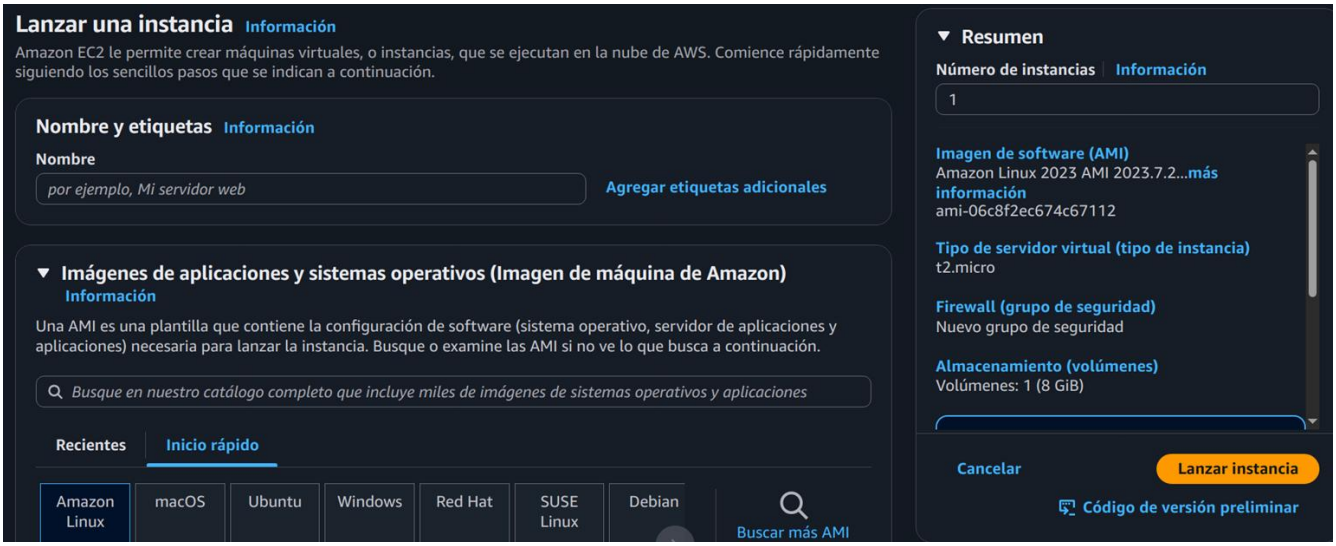
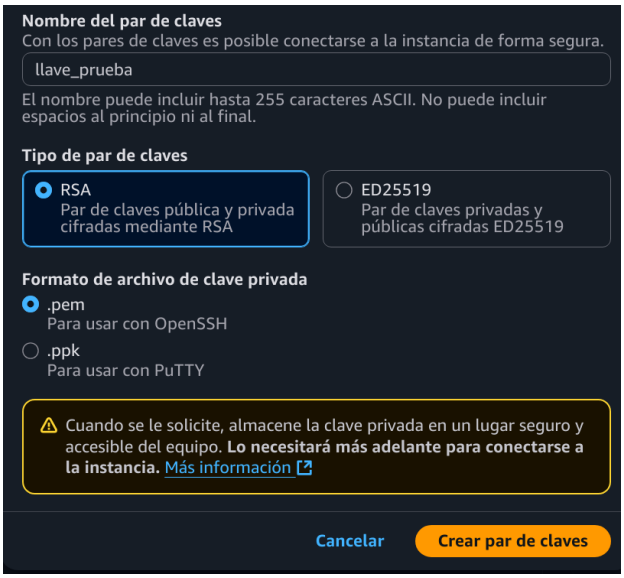
INSTRUCTIVO AWS X EMQX

1. Creación de Instancia en AWS

En primer lugar, debemos de iniciar sesión en nuestra cuenta de AWS. Después, en el buscador de servicios que provee la plataforma, escribimos EC2, Este servicio nos permite crear instancias desde el cual podemos alojar desde nuestro Front-end, Back-end hasta LLM. Ahora, damos clic en el botón “Lanzar la instancia” y nos aparecerá la siguiente ventana.



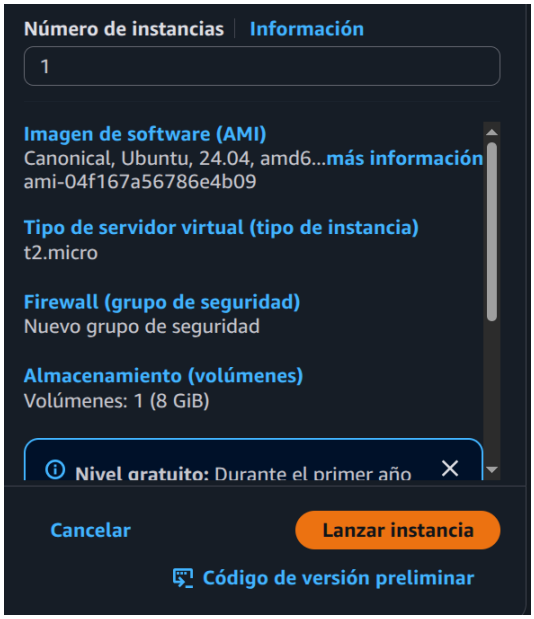
Aquí seleccionaremos Ubuntu y el tipo de instancia (en nuestro caso la capa gratuita). Luego, crearemos un nuevo par de claves, para ello haremos clic en dicha opción, digitaremos el Nombre del par de claves, el tipo y el formato. Para este ejemplo, seleccionaremos “RSA” como tipo y “pem” como formato.



Una vez creado el nuevo par de claves, se descargará un archivo con la extensión .pem (u otra previamente seleccionada). A continuación, será necesario seleccionarlo desde el menú desplegable. Asimismo, se deberá configurar la red en donde marcaran las siguientes opciones:

- Permitir tráfico de SSH desde Cualquier lugar en nuestro caso. Si se quiere que solamente se pueda acceder a la instancia desde una IP o varias en específica se debe seleccionar alguna de las otras opciones
- Permitir el tráfico de HTTP y HTTPS

Con todo lo anterior hecho, solamente nos queda hacer clic en el botón “Lanzar instancia” y esperar unos minutos mientras se crea,



2. Despliegue

Para este ejemplo, se desplegará un back-end creado en FastApi, el cual permite detectar personas mediante un modelo entrenado en MobileNet SSD.

Link del repositorio: *https://github.com/Fullops/Api_Rest_Detection_Raspberry.git*

Teniendo en cuenta lo anterior, deberemos de conectarnos a la instancia que previamente habíamos creado. Para ello, ingresaremos a EC2 > Instancias y seleccionaremos la que se había creado y daremos clic en el botón “Conectar”.

A partir de este punto, todo el proceso se debe realizar desde una consola. Se recomienda utilizar **Git Bash** en sistemas Windows o la terminal predeterminada en sistemas Linux. Ahora, abriremos una nueva consola, nos ubicaremos en la misma carpeta donde se encuentra el archivo que se había descargado y ejecutaremos los siguientes comando:

Chmod 400 "<nombre de la llave>.pem"
ssh -i "<nombre de la llave>.pem" ubuntu@<DNS publica que provee Amazon>

Con esto ya deberíamos estar conectados a la instancia. Para validar en la terminal debera decir ubuntu@<ip de la instancia>

```
[fullops@fullops Descargas]$ chmod 400 "llave_prueba.pem"
[fullops@fullops Descargas]$ ssh -i "llave_prueba.pem" ubuntu@ec2-3-145-153-44.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-3-145-153-44.us-east-2.compute.amazonaws.com (3.145.153.44)' can't be established.
ED25519 key fingerprint is SHA256:DFcbGjCrg42bo/W3uZFcnPX7VgJNdir7TbkjH8msRgg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-145-153-44.us-east-2.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1024-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Thu May 29 01:04:15 UTC 2025

  System load:  0.0               Processes:             104
  Usage of /:   24.9% of 6.71GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for enX0: 172.31.6.74
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-6-74:~$
```

A continuación, actualizarán los paquetes de Ubuntu y se instalará python3 y Nginx utilizando los siguientes comandos:

sudo apt-get update
sudo apt install python3 python3-pip python3-venv
sudo apt install nginx
sudo apt install libgl1

Una vez hecho lo anterior, se clonará el repositorio de github, se creara el entorno virtual e instalarán las respectivas librerías. Se recomienda ejecutar previamente el proyecto que hemos creado para verificar si no faltan algunas librerías o existe alguna incompatibilidad entre versiones de python.

Luego, creamos un archivo con “sudo nano /etc/nginx/sites-available/fastapi” y pegamos lo siguiente:

```
server {
    listen 80;
    server_name <Dirección IPv4 pública>;

    location / {
        proxy_pass http://127.0.0.1:<puerto en donde corre el servidor>;
        proxy_set_header Host $host;
```

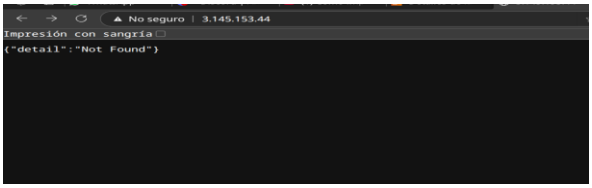
```
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

Ahora, simplemente tendremos que guardar con Ctrl+O+Enter, y nos saldremos de nano con Ctrl+x.

A su vez, tendremos activar el sitio y comprobar si el sitio se ejecuta.

sudo ln -s /etc/nginx/sites-available/fastapi /etc/nginx/sites-enabled
sudo nginx -t
sudo systemctl restart nginx

A continuación, deberemos ejecutar el proyecto que habíamos clonado y con ello deberíamos ver lo siguiente, si ingresamos a “http:<Dirección IPv4 pública>:



Para finalizar, nos quedaría crear un servicio que permita que el proyecto se ejecute como un proceso del sistema. Con ello, tendremos que ejecutar “sudo nano /etc/systemd/system/fastapi.service” y pegar lo siguiente:

```
[Unit]
Description=FastAPI Uvicorn App
After=network.target

[Service]
User=ubuntu
WorkingDirectory=/home/ubuntu/<nombre de la carpeta o repositorio clonado>
ExecStart=/home/ubuntu/<nombre de la carpeta o repositorio clonado>/env/bin/uvicorn app.main:app
--port 8000
Restart=always

[Install]
WantedBy=multi-user.target
```

Ahora debemos guardar y recargar los servicios con “sudo systemctl daemon-reexec” y “sudo systemctl daemon-reload”. Como ultimo paso, estaría Iniciar y habilitar el servicio con los comandos “sudo systemctl start fastapi” y “sudo systemctl enable fastapi”.

Nota: Si queremos verificar el estado del servicio que anteriormente creamos, solamente tenemos de ingresar el comando “sudo systemctl status fastapi”.

3. Creación de MQTT Cloud (EMQX)