

Se accede al servicio de EC2 de AWS

The screenshot shows the AWS EC2 Dashboard for the Europe (Stockholm) Region. On the left, a sidebar lists various EC2 services like Instances, Images, and Elastic Block Store. The main panel displays 'Resources' with counts for running instances, Auto Scaling Groups, Dedicated Hosts, Elastic IPs, Instances, Key pairs, Load balancers, Placement groups, Security groups, Snapshots, and Volumes. It also features a 'Launch instance' button and a 'Service health' section. A sidebar on the right details 'EC2 Free Tier' offers, noting 2 offers in use, end of month forecast, and exceeds free tier limits. The bottom navigation bar includes CloudShell and Feedback.

Apretamos a “Launch Instance” para crear nuestro nuevo servidor en la nube y accedemos al panel de configuración:

Aquí definimos el nombre del servidor: tesis_jesus

The screenshot shows the 'Launch an instance' configuration page. In the 'Name and tags' section, the name 'thesis_jesus' is entered. Under 'Application and OS Images (Amazon Machine Image)', it shows an AMI selection for Canonical, Ubuntu, 22.04 LTS. The 'Summary' panel indicates 1 instance, software image, virtual server type (t3.micro), firewall (New security group), and storage (1 volume - 8 GiB). A callout box highlights the 'Free tier' offer: 'In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free'. At the bottom, there are 'Cancel', 'Launch instance' (in orange), and 'Review commands' buttons.

Seleccionamos el sistema operativo del servidor: Ubuntu 22.04

The screenshot shows the AWS Lambda console interface. In the center, there's a search bar labeled "Search" and a dropdown menu showing "Application and OS Images (Amazon Machine Image)". Below this, a section titled "Quick Start" displays various AMI options: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. The "Ubuntu" option is selected. To the right, a "Summary" panel shows the following details:

- Number of instances: 1
- Software Image (AMI): Canonical, Ubuntu, 22.04 LTS (read more)
- Virtual server type (instance type): t3.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 8 GiB

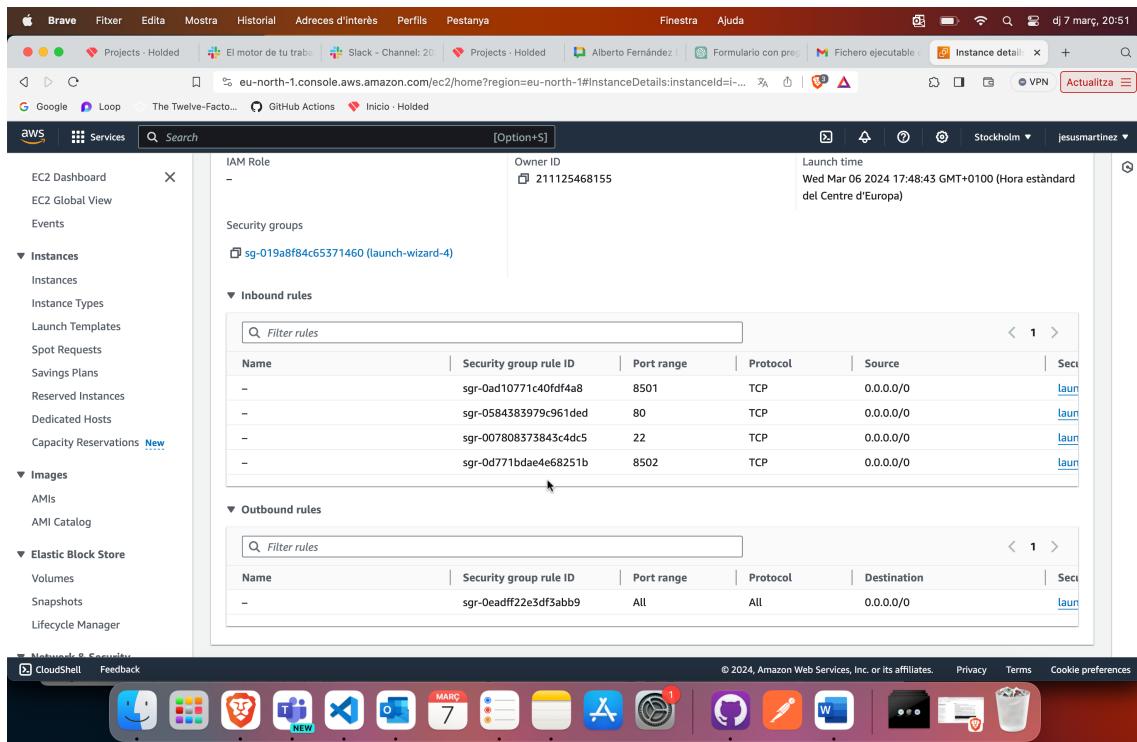
A tooltip for the "Free tier eligible" button indicates: "free tier: in your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free". At the bottom right of the summary panel are buttons for "Cancel", "Launch instance", and "Review commands".

Seleccionamos el tipo de instancia. De la familia T3, 2 cpus y 2GB de memoria.
Definimos la llave pem para poder conectarnos con seguridad al servidor

The screenshot shows the AWS Lambda console interface, similar to the previous one but with different configuration options. The "Instance type" section is now active, showing the "t3.small" option selected. The "Key pair (login)" section shows a key pair named "jesus_server" selected. The "Network settings" section is also visible. The "Summary" panel on the right remains the same, showing the launch configuration and a tooltip for the "Free tier eligible" button.

Dejamos los otros parámetros por defecto y creamos el servidor mediante el botón “Launch Instance”.

Modificaremos los accesos al servidor en el apartado security, para aceptar conexiones por el puerto 22 (ssh), 80 (http) y 8501 y 8502 para streamlit, que es la librería empleada para desarrollar la interfaz web.



The screenshot shows the AWS EC2 Instances details page for an instance named "launch-wizard-4". The instance was launched on Wednesday, March 06, 2024, at 17:48:43 GMT+0100. The security group "sg-019a8f84c65371460" is assigned to the instance. The Inbound rules table lists four rules:

Name	Security group rule ID	Port range	Protocol	Source
-	sgr-0ad10771c40fdf4a8	8501	TCP	0.0.0.0/0
-	sgr-0584383979c961ded	80	TCP	0.0.0.0/0
-	sgr-007808373843dc5	22	TCP	0.0.0.0/0
-	sgr-0d771bdae4e68251b	8502	TCP	0.0.0.0/0

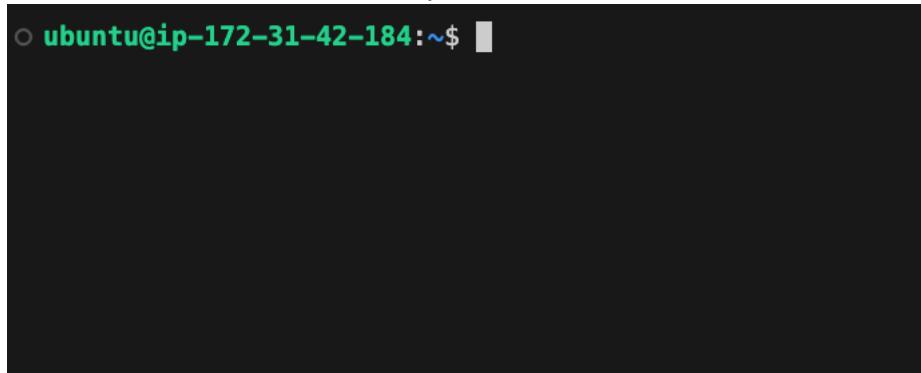
The Outbound rules table shows one rule:

Name	Security group rule ID	Port range	Protocol	Destination
-	sgr-0eadff22e3df3abb9	All	All	0.0.0.0/0

Para conectarnos al servidor mediante ssh, lo hacemos de la siguiente manera a través de una terminal:

```
ssh -i "jesus_server.pem" ubuntu@ec2-16-171-134-47.eu-north-1.compute.amazonaws.com
```

Una vez dentro del servidor, lo podemos verificar viendo la terminal.



```
ubuntu@ip-172-31-42-184:~$
```

A continuación podemos proceder a subir los ficheros necesarios y a hacer la instalación de librerías. Utilizaremos Python3.10 como soporte.

```
sudo apt-get update -y
sudo apt install python3-pip -y
sudo apt install python3.10-venv -y
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

El resultado del directorio en el servidor es el siguiente:

```
✓ jugadores
  ↗ app.py
  ↗ jugadores_bestmodel.keras
  ↗ jugadores_scaler.joblib
  ↗ modelling.py
  ↗ nohup.out
✓ ninos
  ↗ app.py
  ↗ modelling.py
  ↗ ninos_bestmodel.keras
  ↗ ninos_scaler.joblib
> venv
  ↗ data.xlsx
  ↗ nohup.out
  ↗ requirements.txt
```

De la siguiente manera ponemos en marcha la aplicación de los jugadores:

```
● (venv) ubuntu@ip-172-31-42-184:~/jesus$ cd jugadores
● (venv) ubuntu@ip-172-31-42-184:~/jesus/jugadores$ streamlit run app.py

Collecting usage statistics. To deactivate, set browser.gatherUsageStats to False.

You can now view your Streamlit app in your browser.

Network URL: http://172.31.42.184:8501
External URL: http://16.171.134.47:8501
```

De la siguiente manera ponemos en marcha la aplicación de los niños:

```
● (venv) ubuntu@ip-172-31-42-184:~/jesus$ cd ninos/
○ (venv) ubuntu@ip-172-31-42-184:~/jesus/ninos$ streamlit run app.py

Collecting usage statistics. To deactivate, set browser.gatherUsageStats to False.

You can now view your Streamlit app in your browser.

Network URL: http://172.31.42.184:8502
External URL: http://16.171.134.47:8502
```

Si accedemos a las respectivas urls, podremos acceder a las pagina web.