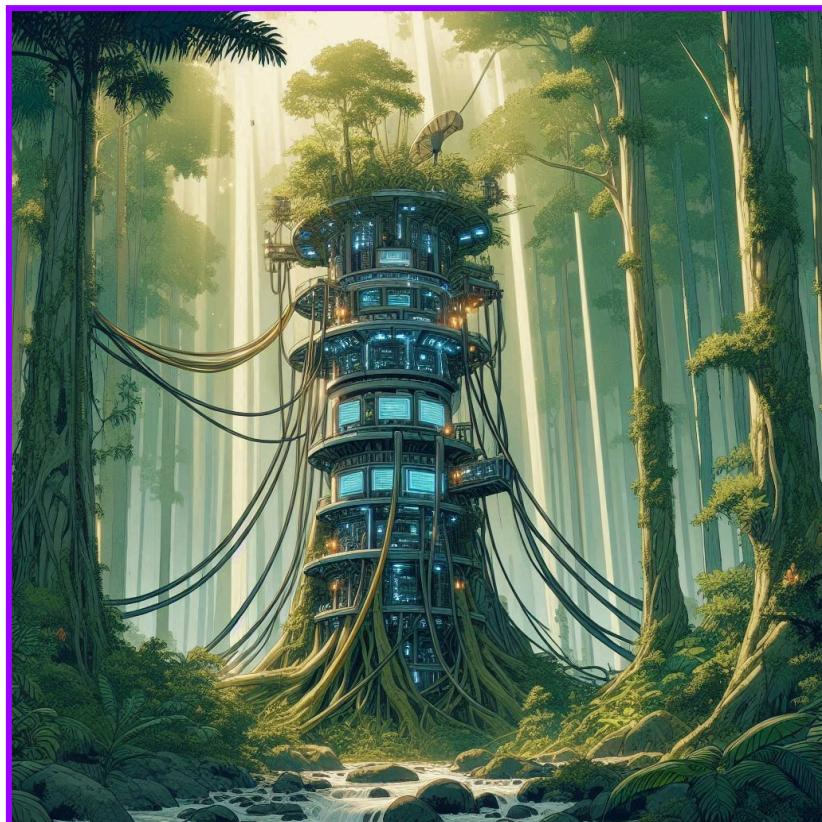


EJERCICIOS SISTEMAS

INFORMÁTICOS TEMA 4:

AWS EC2



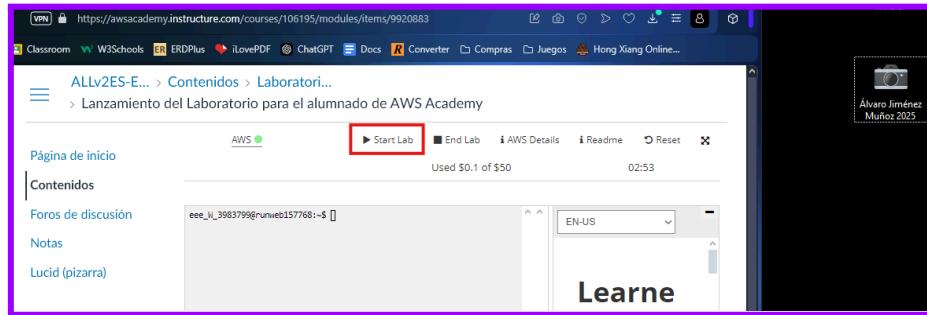
1.º DAM

ÍNDICE

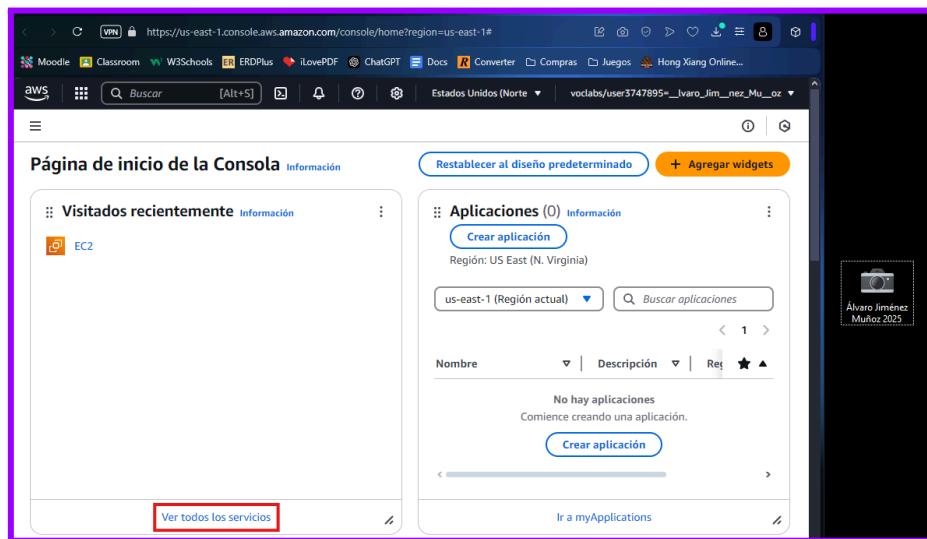
Pág. 3. <u>Actividad 1</u>
Pág. 8. <u>Actividad 2</u>
Pág. 12. <u>Actividad 3</u>
Pág. 17. <u>Actividad 4</u>
Pág. 21. <u>Actividad 5</u>
Pág. 21. <u>Bibliografía</u>

Actividad 1

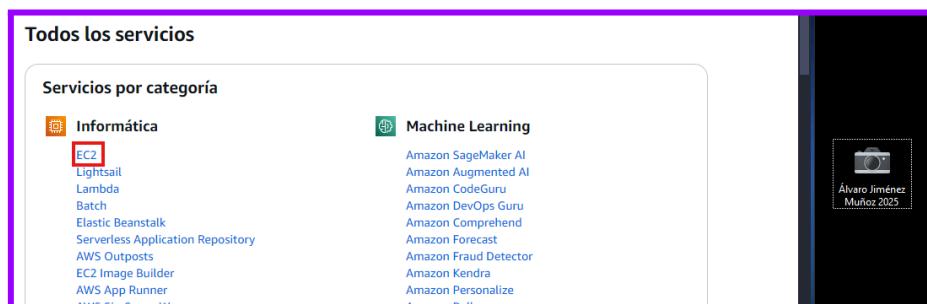
Para crear una instancia EC2 con **Amazon Web Services**, lo primero que se debe hacer es lanzar el laboratorio desde la web de **AWS Academy** en el que se realizarán todos los ejercicios:



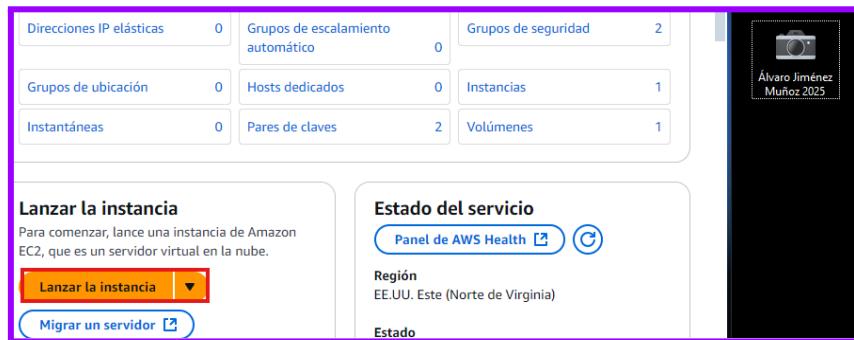
Una vez dentro se le da a “ver todos los servicios”:



Y se selecciona EC2:



En la nueva pestaña que se abrirá, se le da al botón “**Lanzar la instancia**”, para empezar a configurarla:

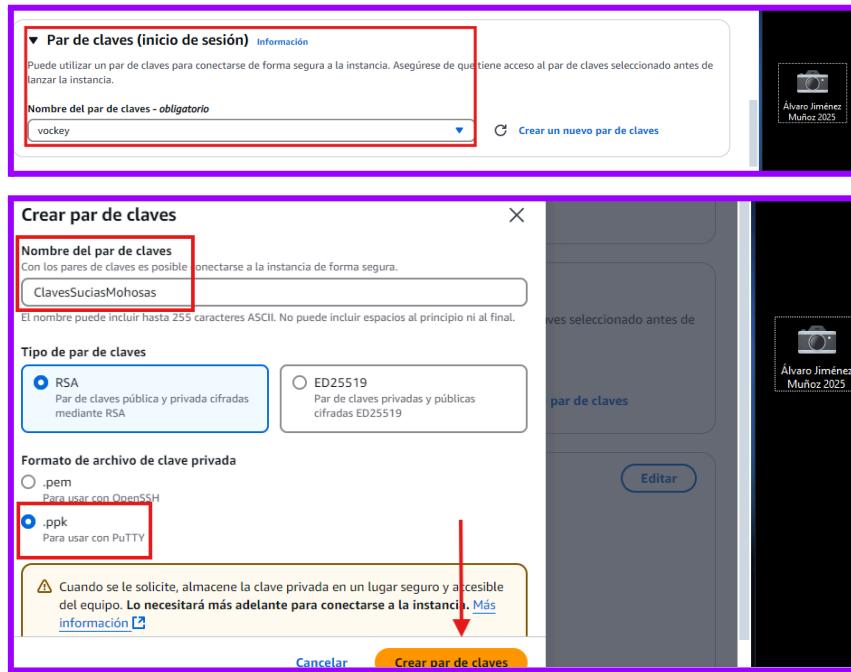


Ya, habiendo hecho todo eso, se establecen todos los ajustes iniciales de la instancia:

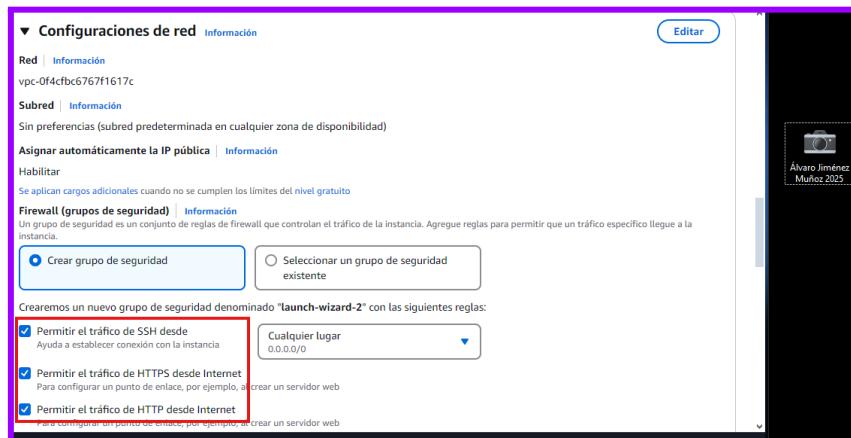
The three screenshots show the configuration steps for launching an AWS Lambda instance:

- Nombre y etiquetas**: Shows the "Nombre" field filled with "UbuntuDockerAWS" and a red box highlighting it. Below is a search bar and a list of recent AMI images.
- Imagenes de máquinas de Amazon (AMI)**: Shows the "ubuntu" AMI selected from a grid of options. A red box highlights the "ubuntu" option. Below the grid is a search bar and a "Buscar más AMI" (Search more AMI) button.
- Imagenes de máquinas de Amazon (AMI)**: Provides detailed information about the selected AMI: "Ubuntu Server 24.04 LTS (HVM), SSD Volume Type". It includes fields for "Arquitectura" (Architecture), "ID de AMI" (AMI ID), "Nombre de usuario" (User name), and "Proveedor verificado" (Verified provider).
- Tipo de instancia**: Shows the "t2.micro" instance type selected. A red box highlights the "t2.micro" option. Below it is a table with instance details: Family: t2, 1 vCPU, 1 GiB Memoria, Generación actual: true, Bajo demanda Windows base precios: 0.0163 USD per Hour, Bajo demanda Ubuntu Pro base precios: 0.0134 USD per Hour, Bajo demanda SUSE base precios: 0.0116 USD per Hour, Bajo demanda RHEL base precios: 0.026 USD per Hour, Bajo demanda Linux base precios: 0.0116 USD per Hour. It also includes links for "Apto para la capa gratuita" (Free tier eligible), "Todas las generaciones" (All generations), and "Comparar tipos de instancias" (Compare instance types).

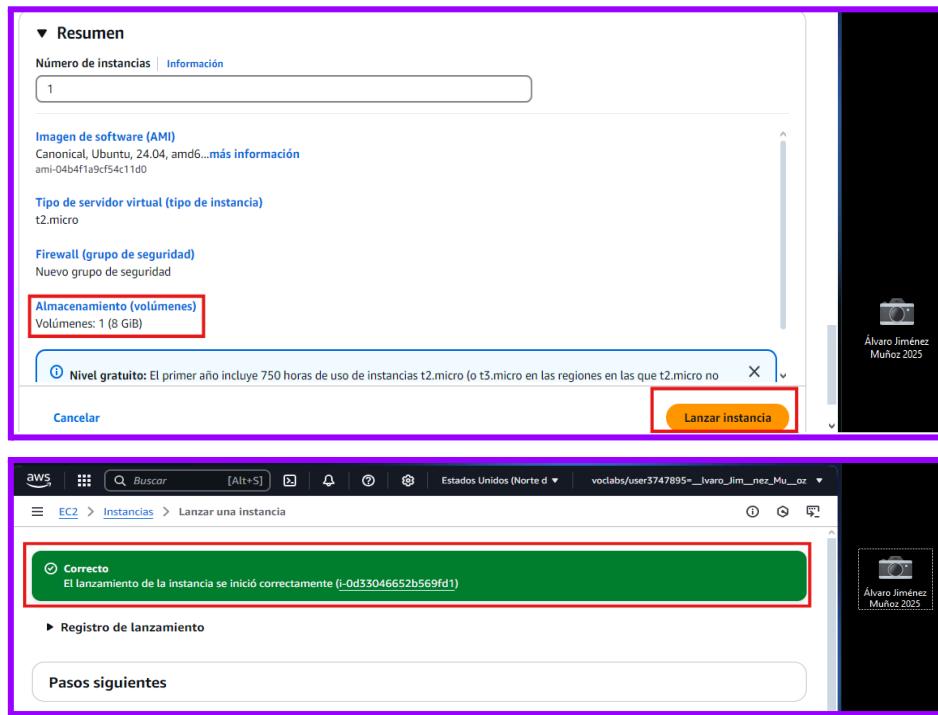
Se establece el acceso por clave y se descarga el archivo que las contiene, estableciendo como formato de la clave privada “.ppk” para que sea compatible con Putty:



Ya por último se configura el firewall para que permita el tráfico de SSH, HTTP y HTTPS:



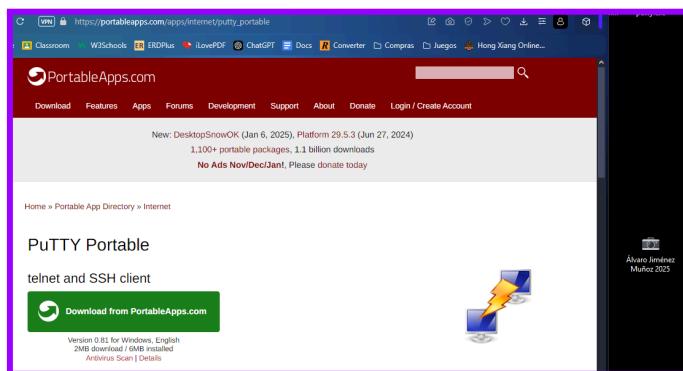
Después de eso, se lanzaría la instancia, dejando el resto de configuraciones como vienen por defecto:



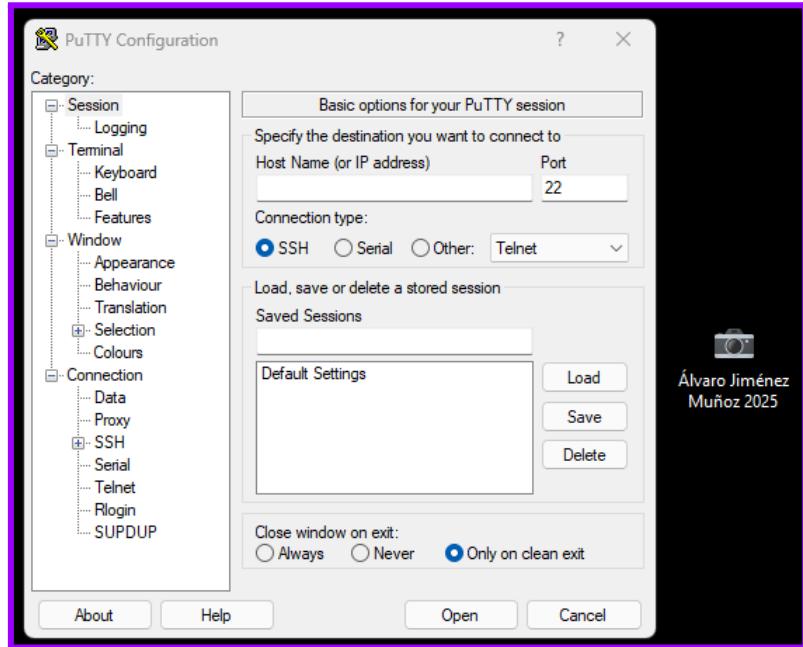
En esa captura puede verse cómo la instancia se ha lanzado correctamente.

Actividad 2

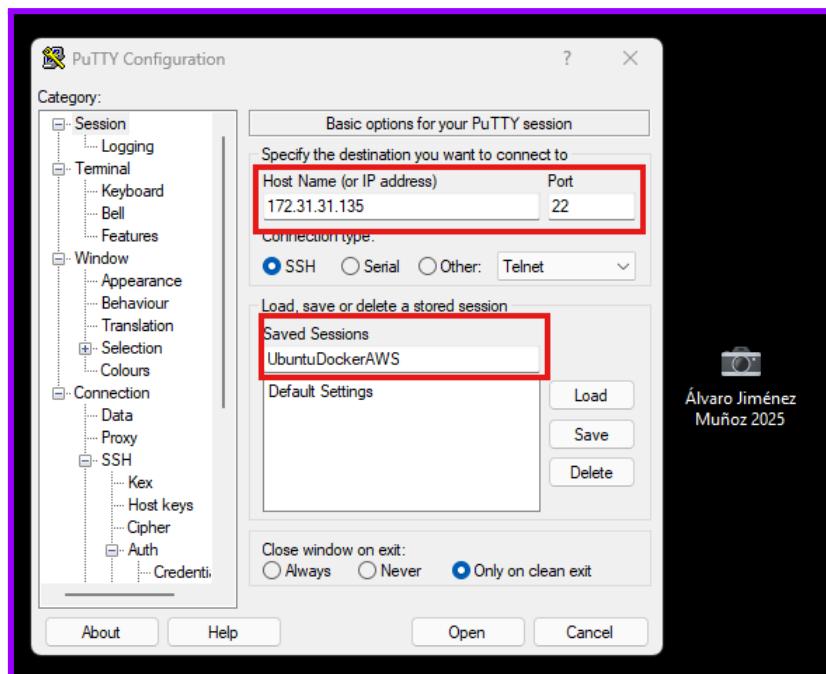
Para acceder a la instancia usando la aplicación “Putty” desde el escritorio, evitando así tener que acceder a ella desde la web de Amazon, primero se descarga la aplicación desde el navegador, en concreto la versión portable, que es más que suficiente y no necesita permisos de administrador para ejecutarse:



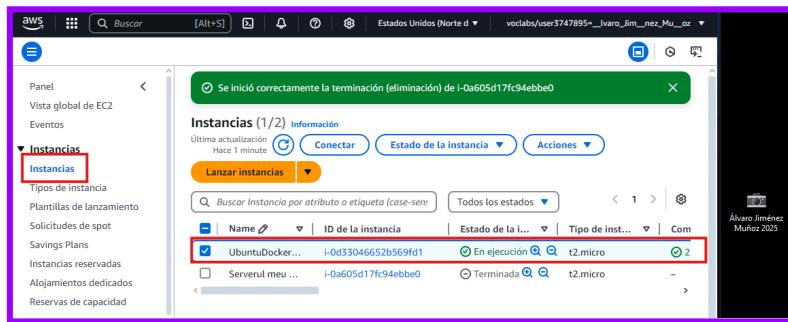
Una vez descargada, se ejecuta y saldrá la siguiente ventana:



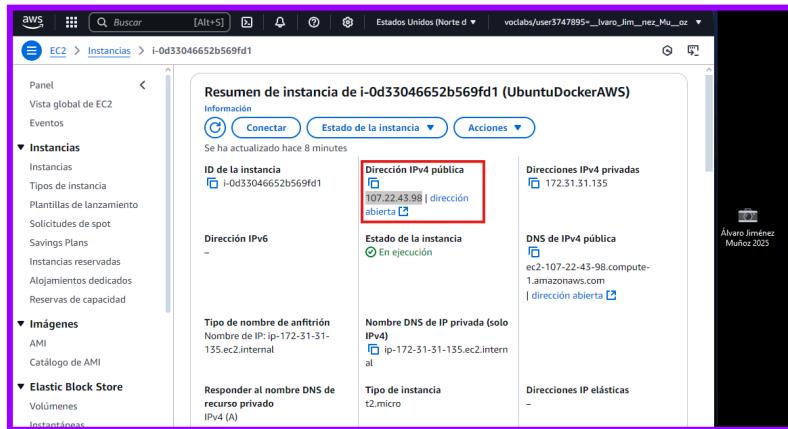
Aquí se deberán establecer una serie de ajustes para poder acceder a la instancia; para ello, primero se deberá establecer la IP junto con el puerto y el nombre de la sesión:



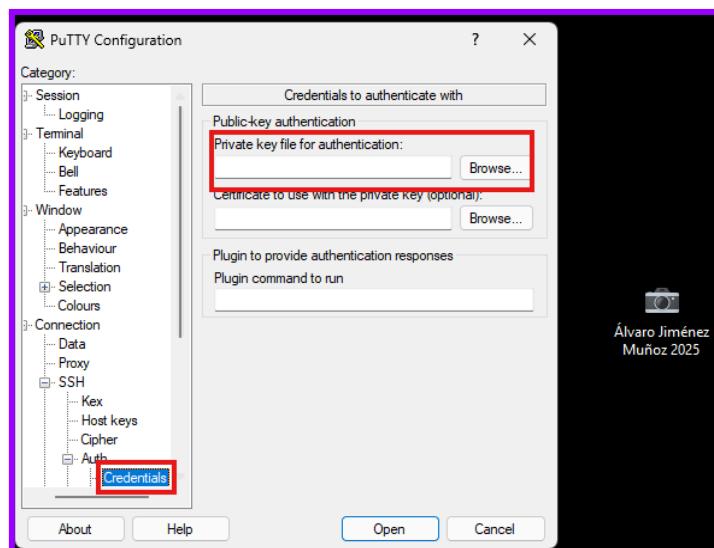
Para obtener la IP hay que ir al apartado de instancias en EC2 y ahí saldrá la instancia antes creada:

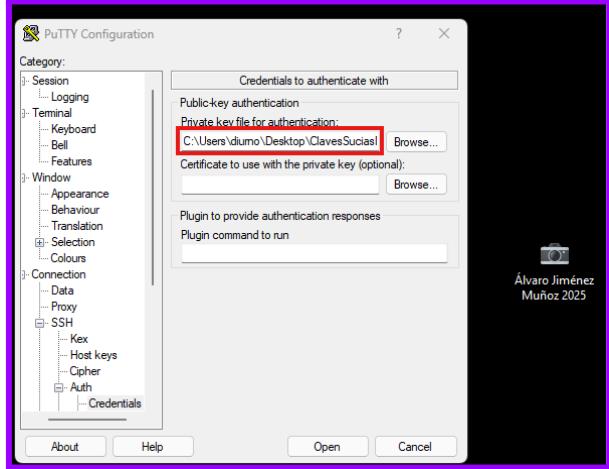


Al clicar en ella, saldrá una nueva ventana con una serie de datos como la IP, que se copiará y pegará en Putty:

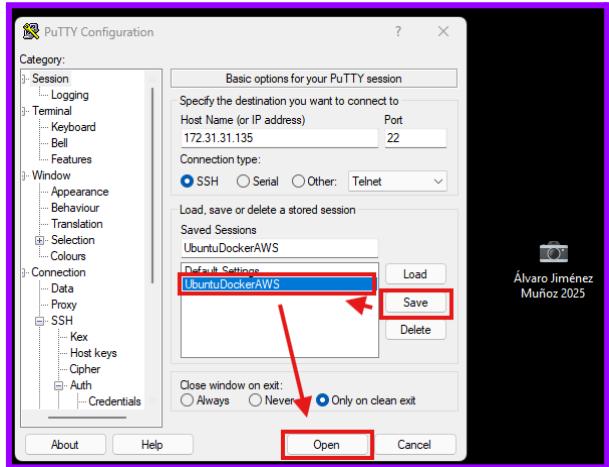


Además de eso, se deberá navegar al siguiente apartado dentro de Putty, donde se cargará el archivo con las claves antes descargadas al crear la instancia:

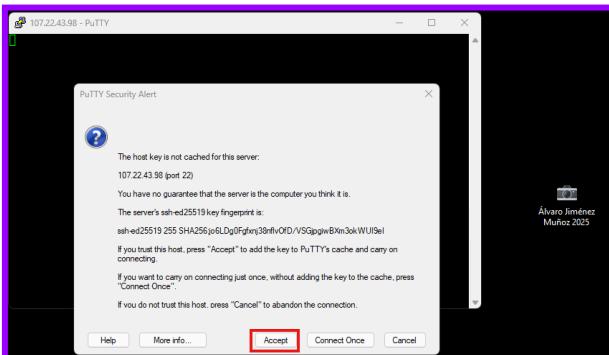




Habiendo establecido esas 3 cosas ya, se vuelve a la pestaña inicial y con el nombre de la sesión escrito se clica en save, en el nombre generado al hacerlo y después en open:



Ya con todo eso hecho se lanzará la sesión, no sin antes mostrar un aviso de seguridad que se acepta sin más:



Después pedirá por último loguearse, por lo que se escribe **ubuntu**:

```
107.22.43.98 - Putty
login as: ubuntu

System load: 0.0      Processes: 104
Usage of /: 25.4% of 6.71GB  Users logged in: 0
Memory usage: 21%      IPv4 address for enX0: 172.31.31.135
Swap usage: 0%
* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.
https://ubuntu.com/aws/pro
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

Last login: Mon Jan 20 11:39:20 2025 from 83.60.148.114
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

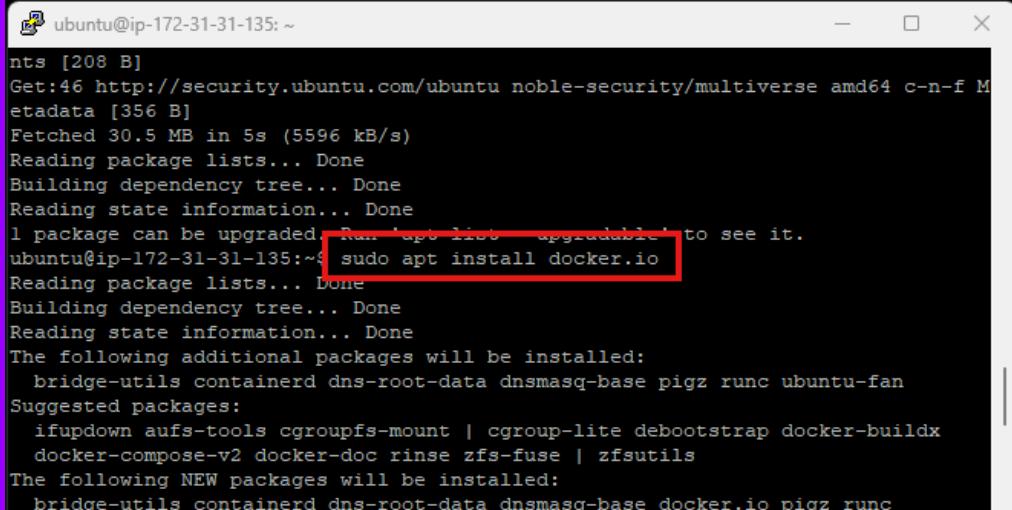
ubuntu@ip-172-31-31-135:~$
```

Habiendo hecho eso, ya se habría lanzado la instancia con **PuTTY**.

Lo siguiente es actualizar los paquetes de **Ubuntu**:

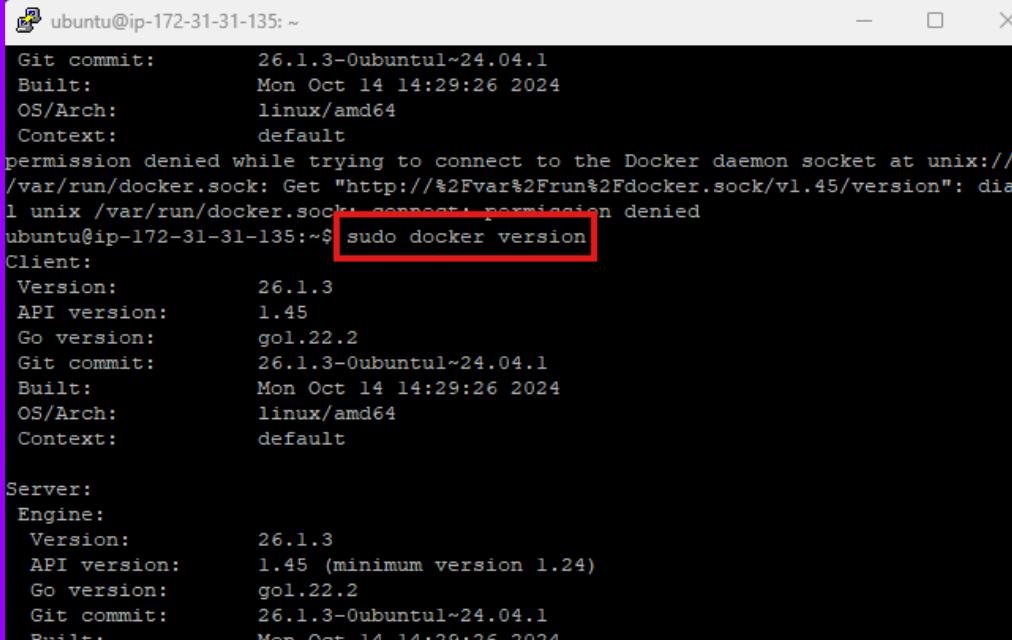
```
ubuntu@ip-172-31-31-135:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-
```

Y después se instala docker.io en el servidor:



```
ubuntu@ip-172-31-31-135: ~
  nts [208 B]
Get:46 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f M
  etadata [356 B]
Fetched 30.5 MB in 5s (5596 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
ubuntu@ip-172-31-31-135:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx
  docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc
```

También se comprueba con sudo docker versión si efectivamente el docker se ha instalado sin problemas:

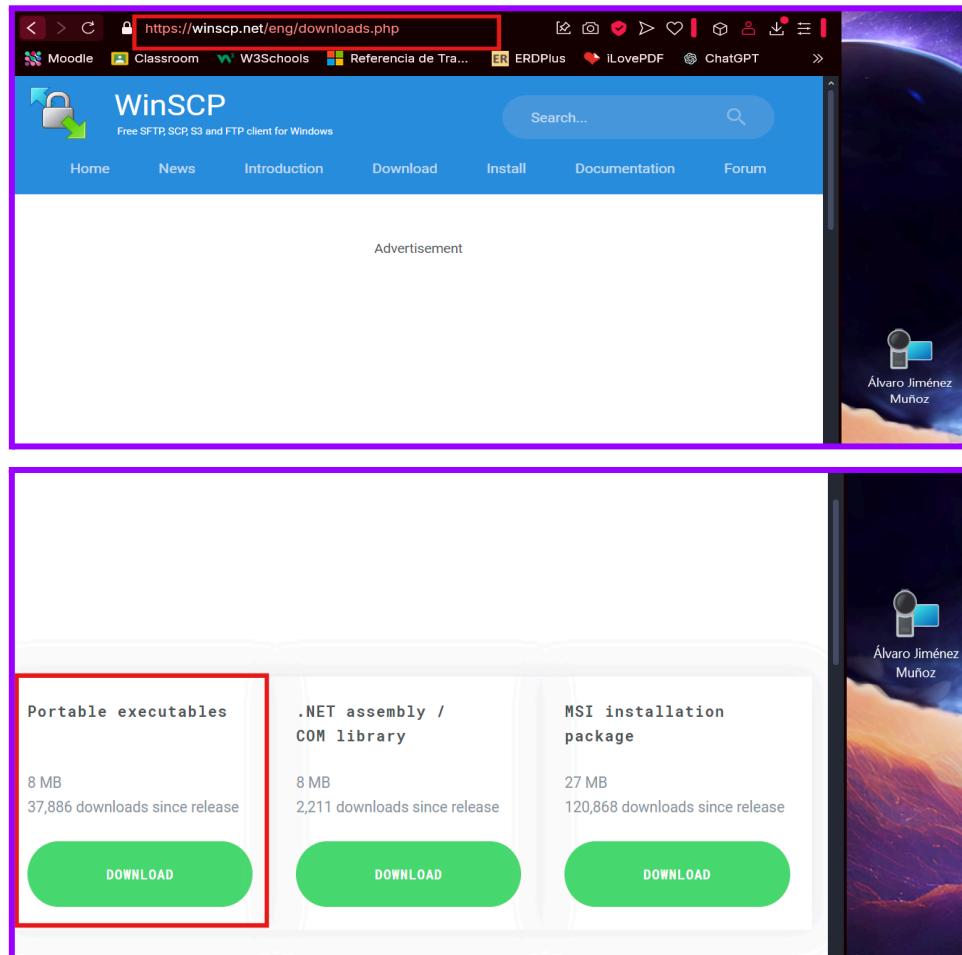


```
ubuntu@ip-172-31-31-135: ~
  Git commit:      26.1.3-0ubuntu1~24.04.1
  Built:          Mon Oct 14 14:29:26 2024
  OS/Arch:        linux/amd64
  Context:        default
  permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.45/version": dial unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-31-135:~$ sudo docker version
Client:
  Version:       26.1.3
  API version:   1.45
  Go version:    go1.22.2
  Git commit:    26.1.3-0ubuntu1~24.04.1
  Built:          Mon Oct 14 14:29:26 2024
  OS/Arch:        linux/amd64
  Context:        default

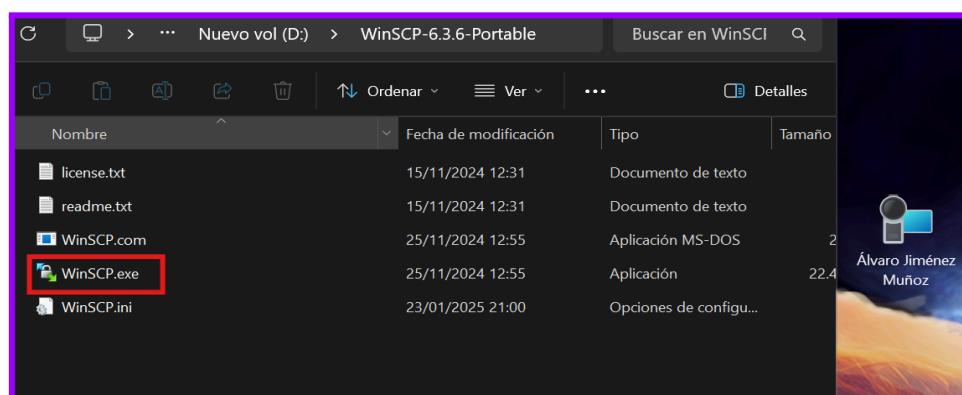
Server:
  Engine:
    Version:       26.1.3
    API version:  1.45 (minimum version 1.24)
    Go version:   go1.22.2
    Git commit:   26.1.3-0ubuntu1~24.04.1
    Built:         Mon Oct 14 14:29:26 2024
```

Ahí se puede ver como se ha instalado sin problemas.

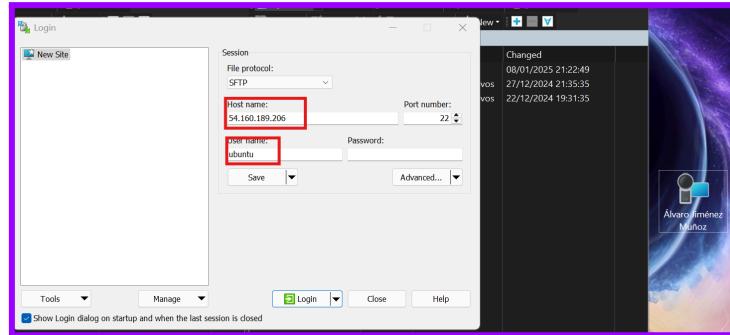
Luego se instala WinSCP, la versión portable también, desde la página web oficial o cualquiera otra:



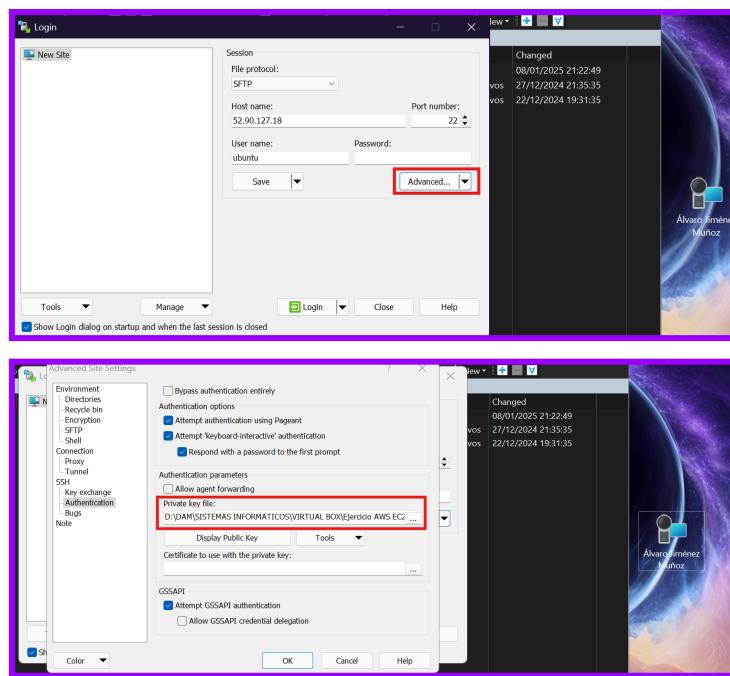
Una vez descargado se ejecuta el siguiente archivo dentro de la carpeta de la descarga:



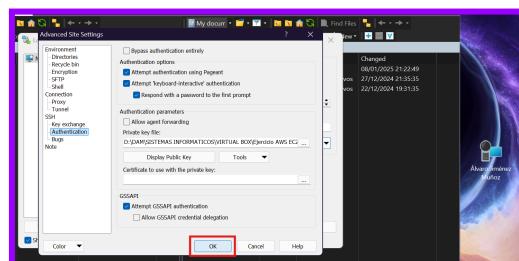
Tras hacerlo, saldrá la siguiente ventana en donde, al igual que se hizo con Putty, se pone la IP de la instancia de EC2 en ejecución, la cual cada vez que se detenga la instancia cambiará, por lo que no va a ser la misma que se puso anteriormente en este caso, además también se pone el username:



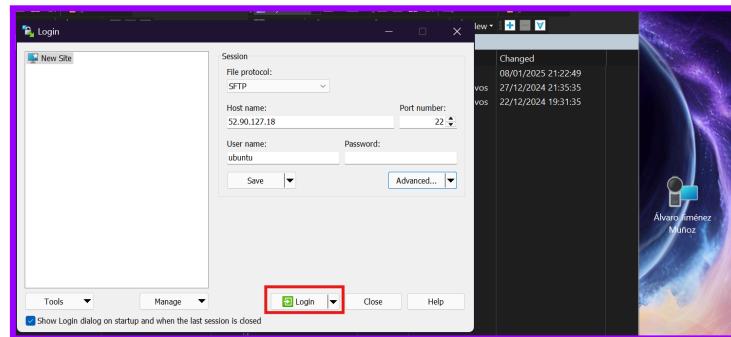
El campo de contraseña se dejará vacío, y en cambio se hará lo siguiente:



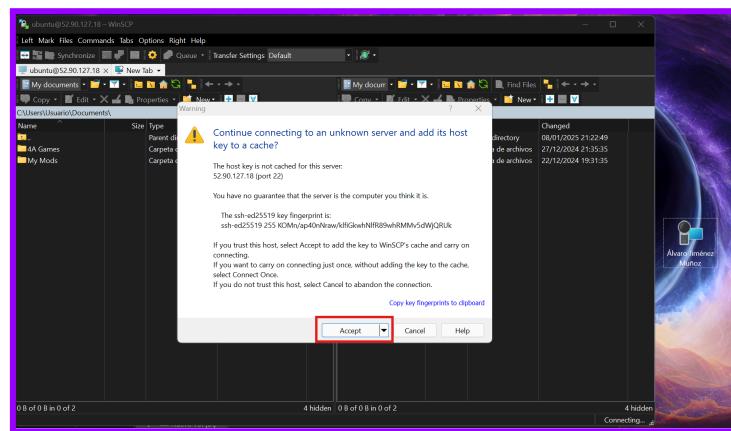
Ahí se pone el par de claves descargadas.



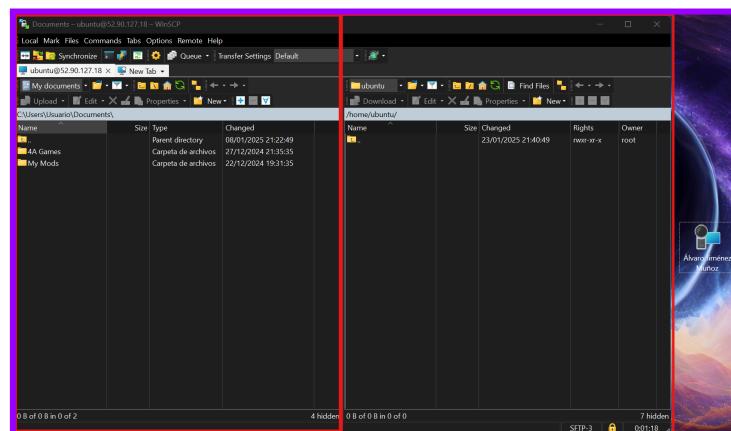
Ya con todo eso, se le da a login:



Al hacerlo dará un aviso, y al igual que antes con Putty se le da aceptar:

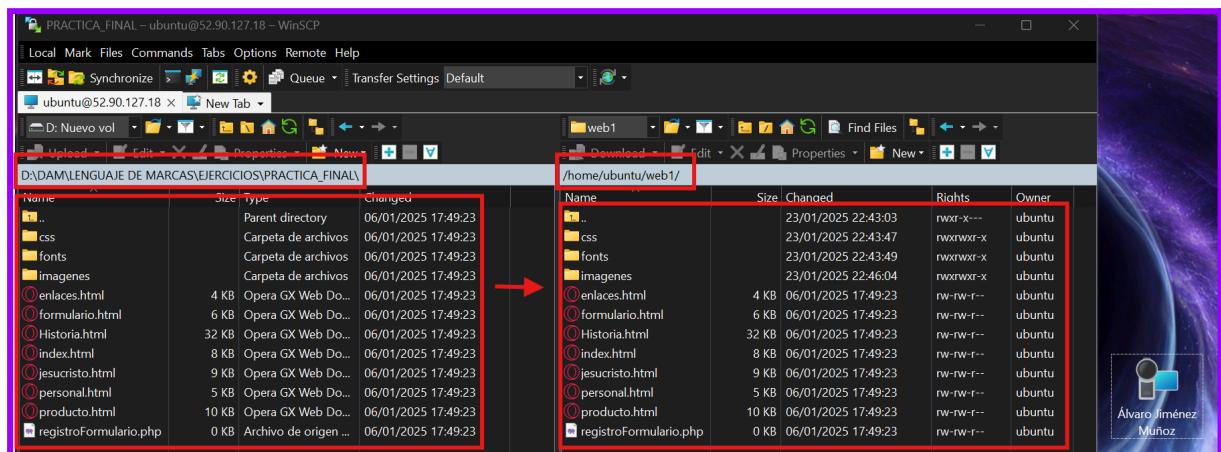
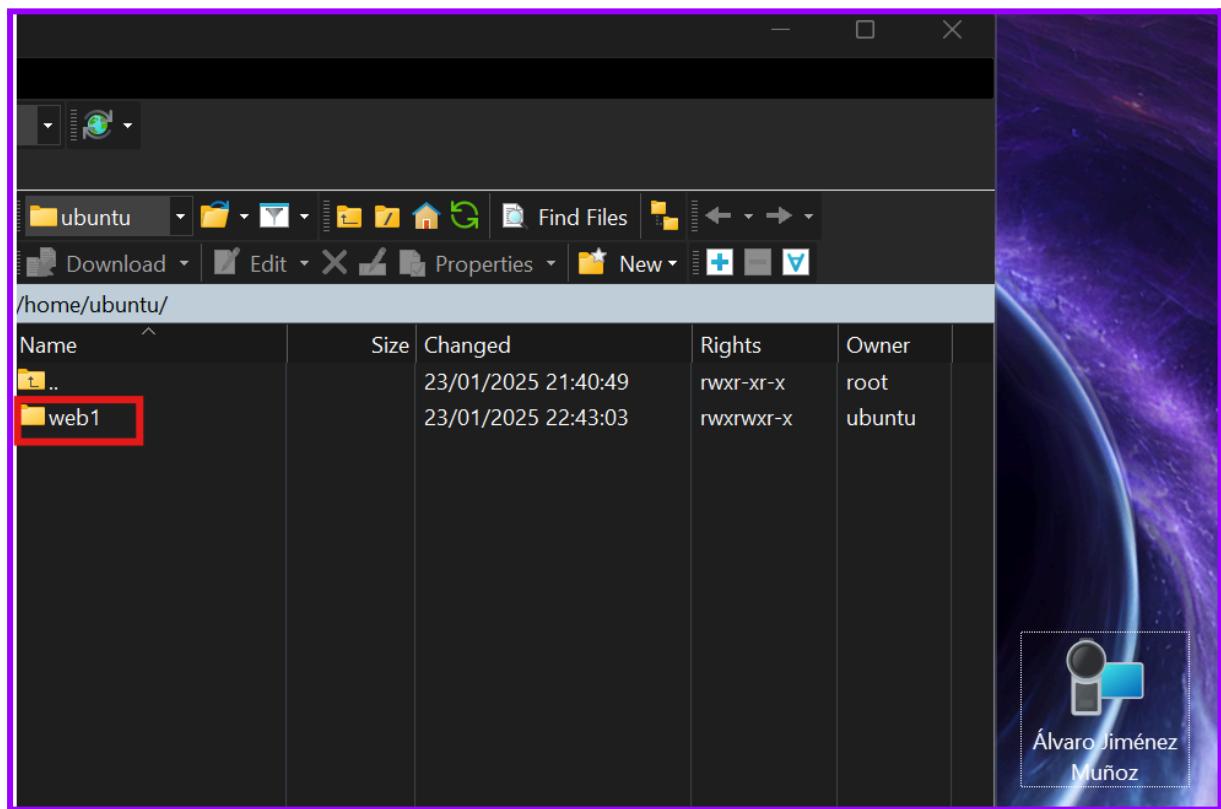


Si después de darle no da ningún problema, se habrá hecho correctamente la conexión con la instancia EC2, y saldrán 2 paneles como aquí se ve:

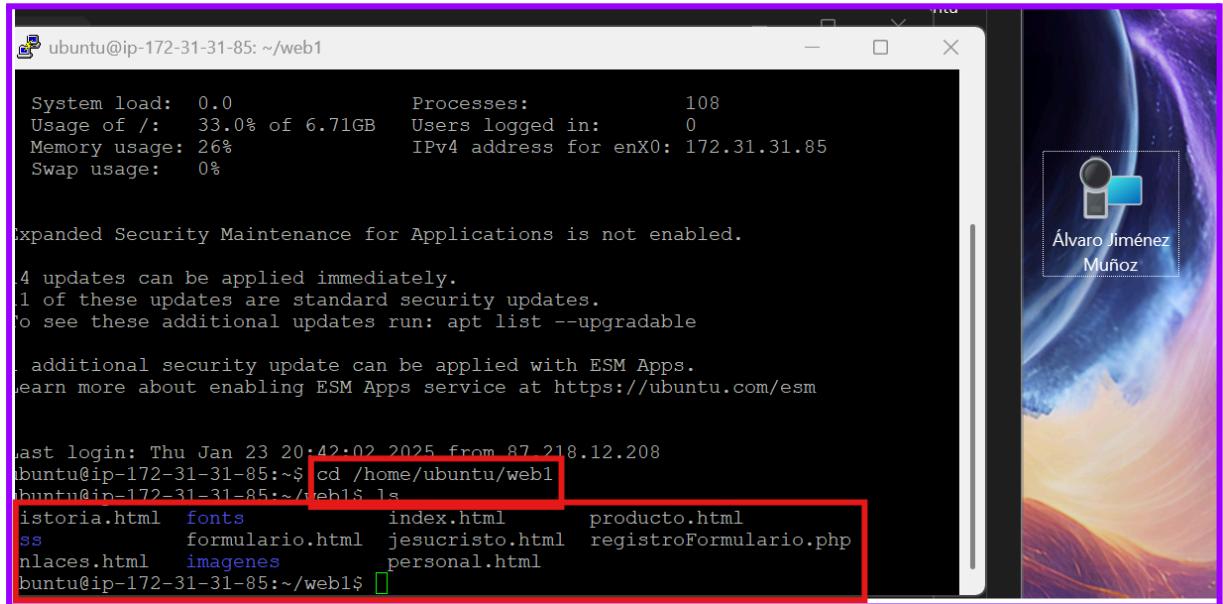


El panel de la izquierda representa la máquina local, mientras que el de la derecha representa a la instancia de AWS, que será donde se subirán los diferentes que componen la página web a lanzar.

Para llevarlo a cabo, en ese mismo panel se navega hasta la carpeta **/home/ubuntu**, que es donde por defecto el panel derecho se sitúa al inicio, y se crea una carpeta llamada **web1**, donde subir esos archivos de la web:



Una vez hecho eso, se abre de nuevo **Putty** y se navega hacia esa carpeta creada para ver si todo esta en orden, haciendo cd y ls:



ubuntu@ip-172-31-31-85: ~/web1

```
System load: 0.0      Processes: 108
Usage of /: 33.0% of 6.71GB  Users logged in: 0
Memory usage: 26%      IPv4 address for enX0: 172.31.31.85
Swap usage: 0%         

expanded Security Maintenance for Applications is not enabled.

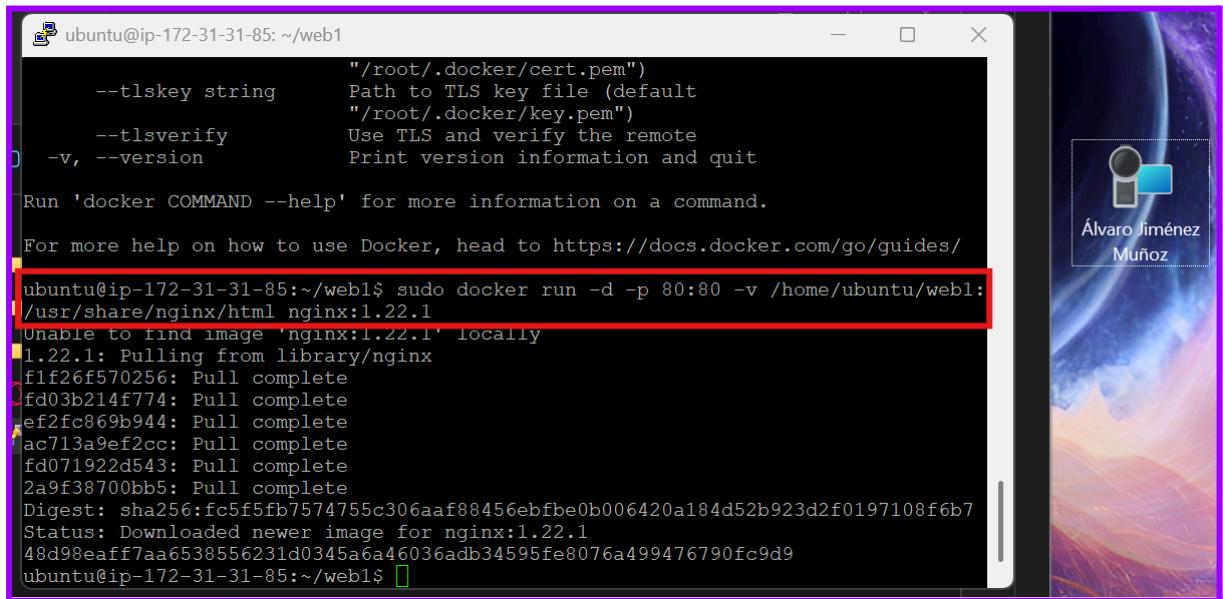
4 updates can be applied immediately.
1 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Thu Jan 23 20:42:02 2025 from 87.218.12.208
ubuntu@ip-172-31-31-85:~$ cd /home/ubuntu/web1
ubuntu@ip-172-31-31-85:~/web1$ ls
istoria.html  fonts      index.html  producto.html
ss            formulario.html jesucristo.html registroFormulario.php
nlaces.html   imagenes   personal.html
ubuntu@ip-172-31-31-85:~/web1$
```

Con el ls, se puede ver como están todos los archivos subidos a la instancia de EC2.

Por lo que, comprobado y hecho ya todo eso, se utiliza docker para publicar la página, usando **nginx:1.22.1** en el puerto 80:

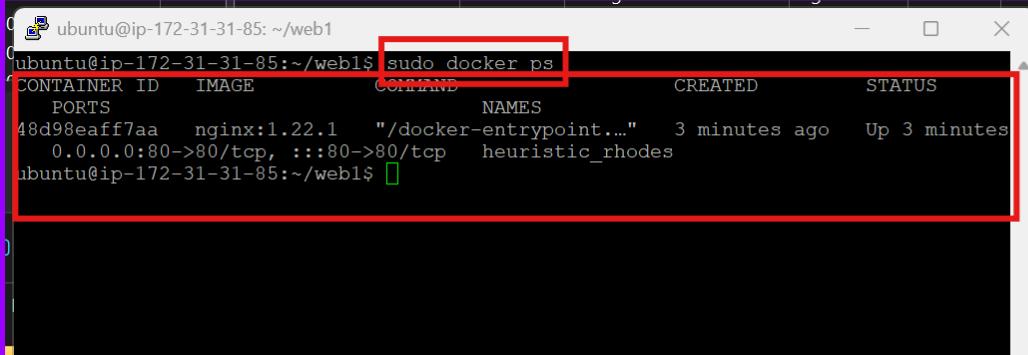


```
ubuntu@ip-172-31-31-85: ~/web1
  --tlskey string      "/root/.docker/cert.pem")
  --tlsverify          Path to TLS key file (default
                       "/root/.docker/key.pem")
  -v, --version         Use TLS and verify the remote
                       Print version information and quit

Run 'docker COMMAND --help' for more information on a command.

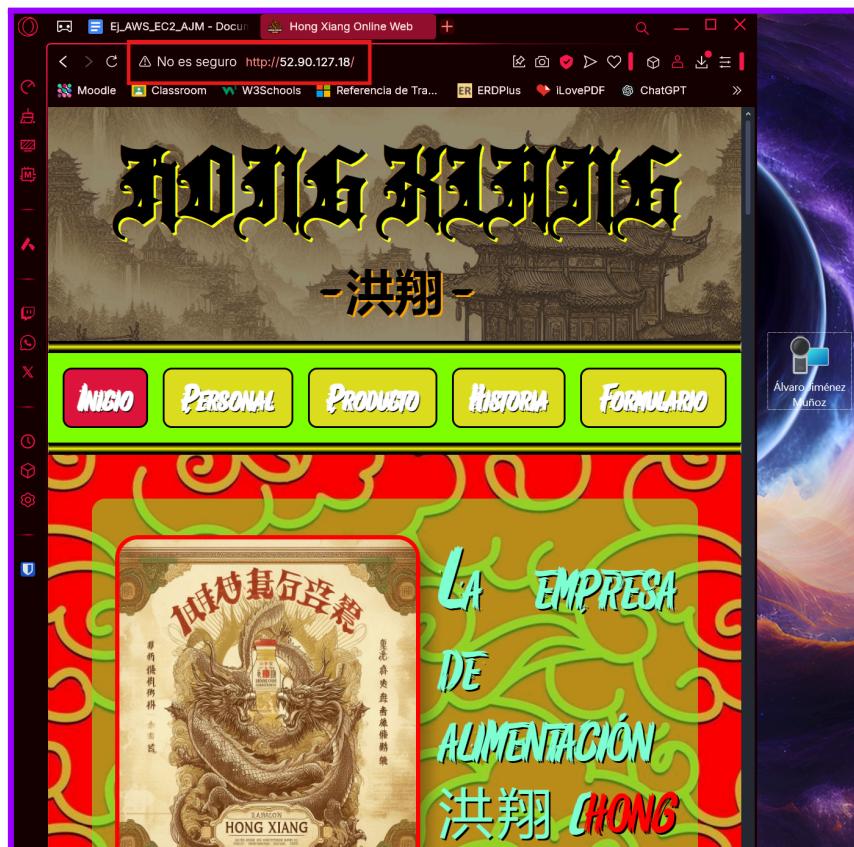
For more help on how to use Docker, head to https://docs.docker.com/go/guides/
ubuntu@ip-172-31-31-85:~/web1$ sudo docker run -d -p 80:80 -v /home/ubuntu/web1:
/usr/share/nginx/html nginx:1.22.1
unable to find image 'nginx:1.22.1' locally
1.22.1: Pulling from library/nginx
f1f26f570256: Pull complete
fd03b214f774: Pull complete
ef2fc869b944: Pull complete
ac713a9ef2cc: Pull complete
fd071922d543: Pull complete
2a9f38700bb5: Pull complete
Digest: sha256:fc5f5fb7574755c306aaf88456ebfbe0b006420a184d52b923d2f0197108f6b7
Status: Downloaded newer image for nginx:1.22.1
48d98eaff7aa6538556231d0345a6a46036adb34595fe8076a499476790fc9d9
ubuntu@ip-172-31-31-85:~/web1$
```

Con `docker ps` se puede ver que el contenedor con `nginx` se ha lanzado correctamente:



```
ubuntu@ip-172-31-31-85:~/web1$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
48d98eaff7aa      nginx:1.22.1       "/docker-entrypoi..."   3 minutes ago     Up 3 minutes
0.0.0.0:80->80/tcp, :::80->80/tcp   heuristic_rhodes
ubuntu@ip-172-31-31-85:~/web1$
```

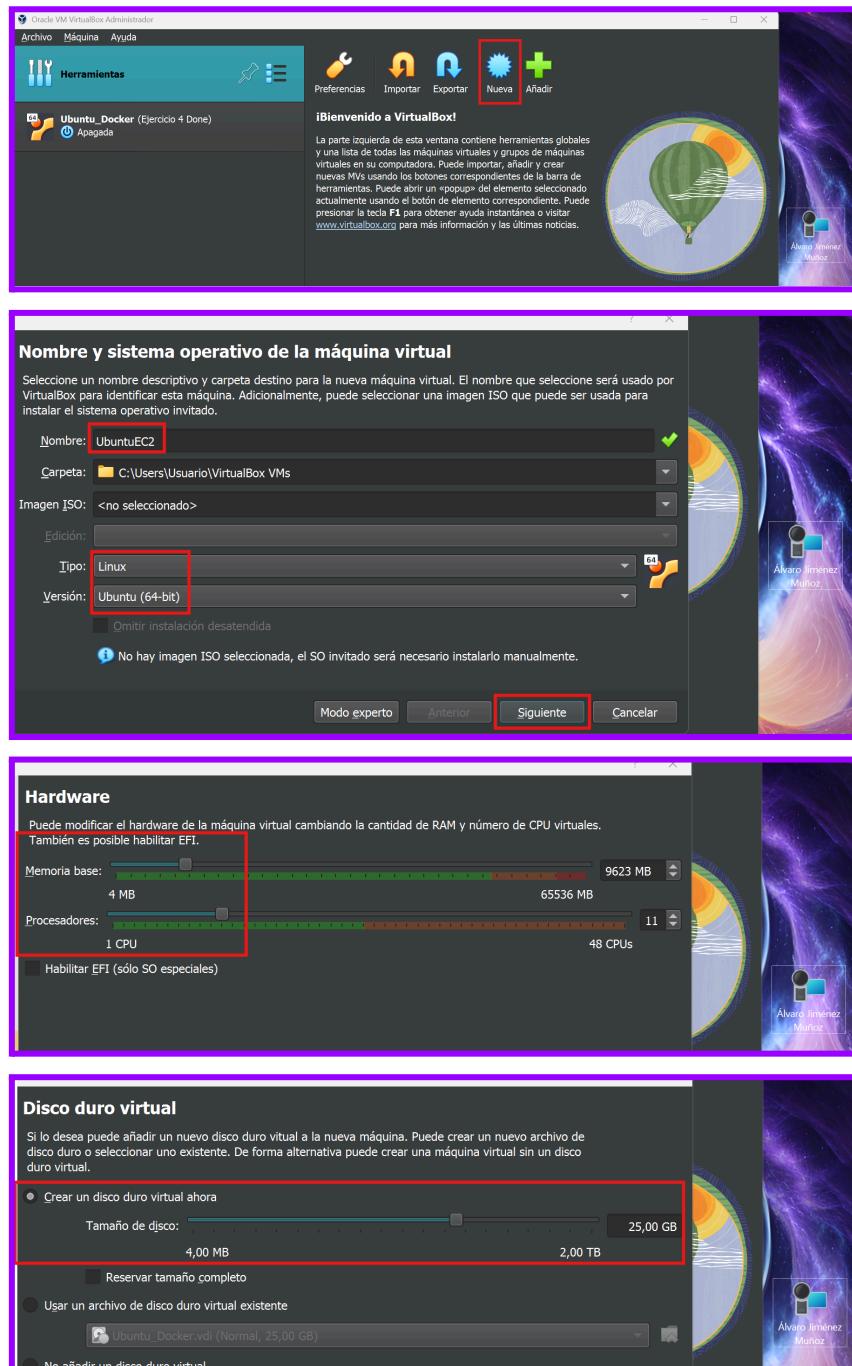
Verificado ya todo, por último, se accede a la página web usando la IP de la instancia EC2:

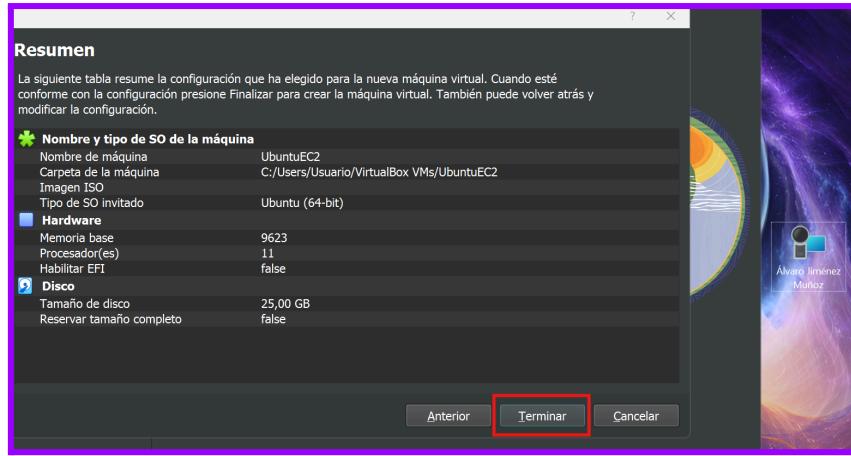


En la imagen se puede ver cómo, a pesar de que indica que la conexión no es segura, ha sido posible acceder a la web desde el navegador.

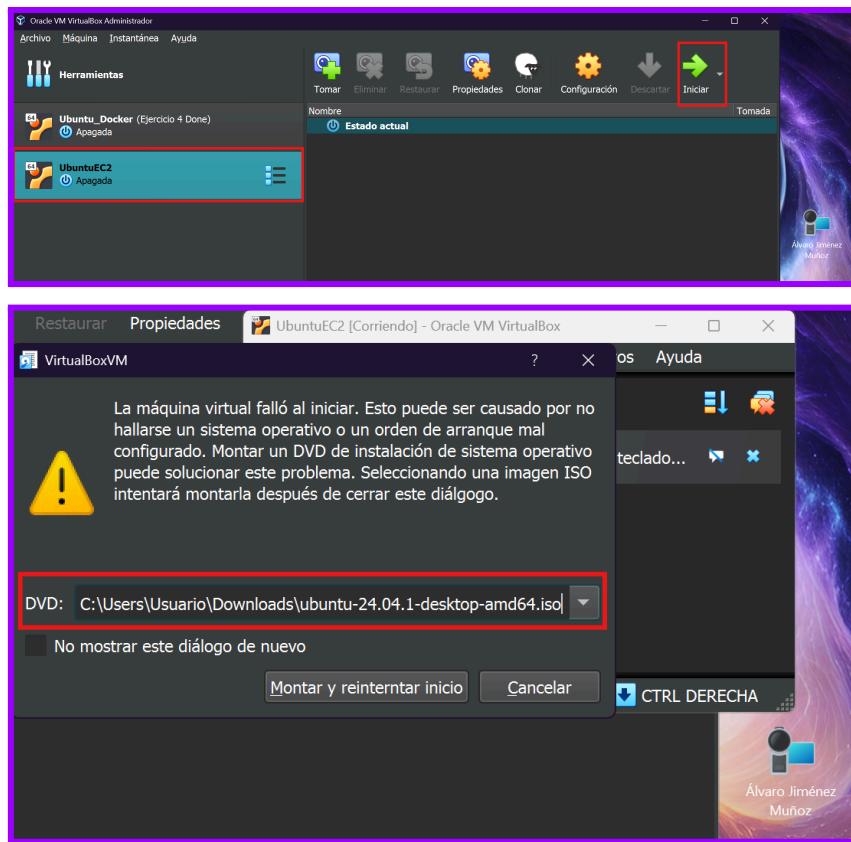
Actividad 3

Lo primero de todo en esta actividad será crear una máquina virtual Ubuntu con VirtualBox y configurarla:

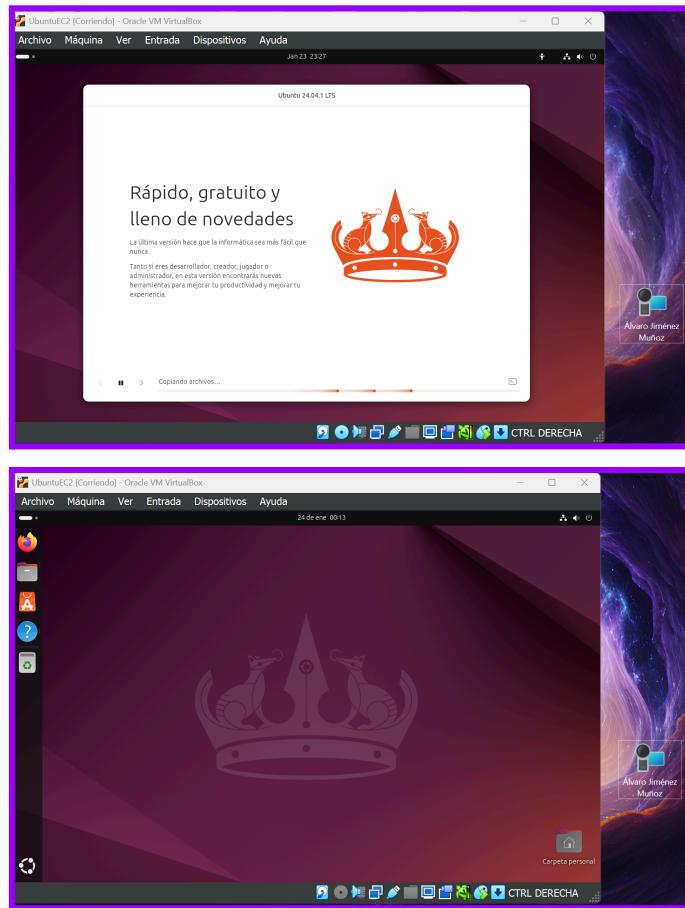




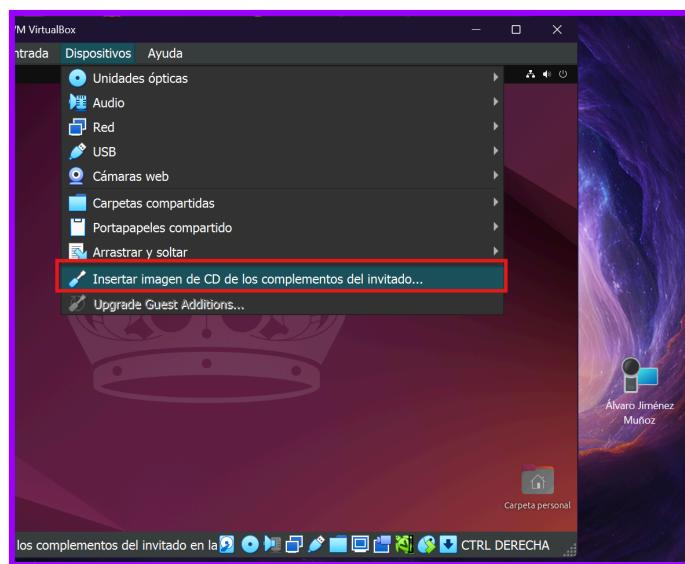
Después se arranca y se selecciona la ISO de ubuntu:

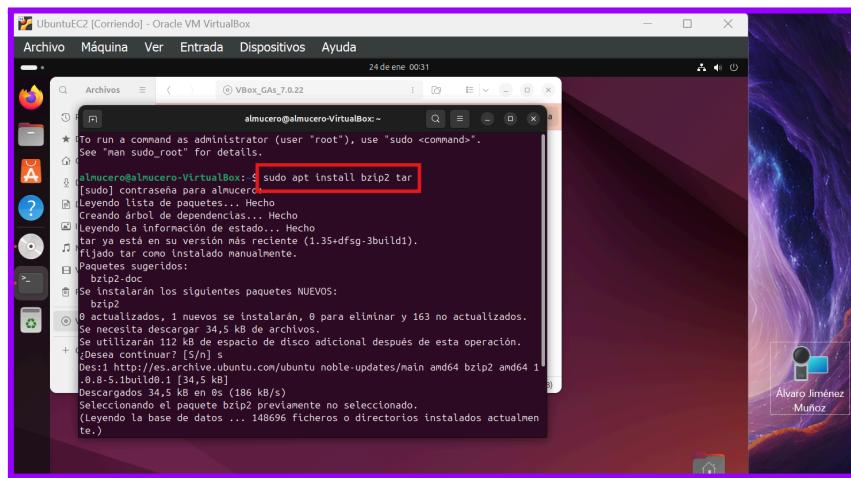
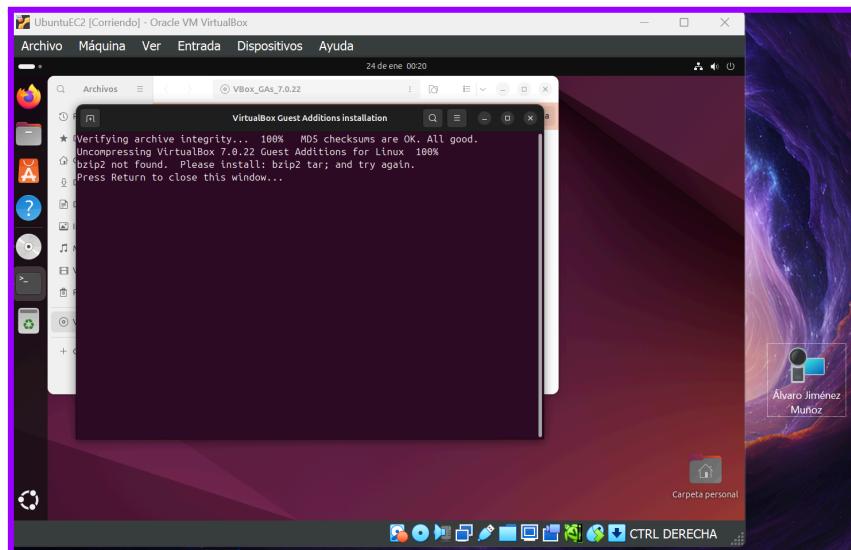
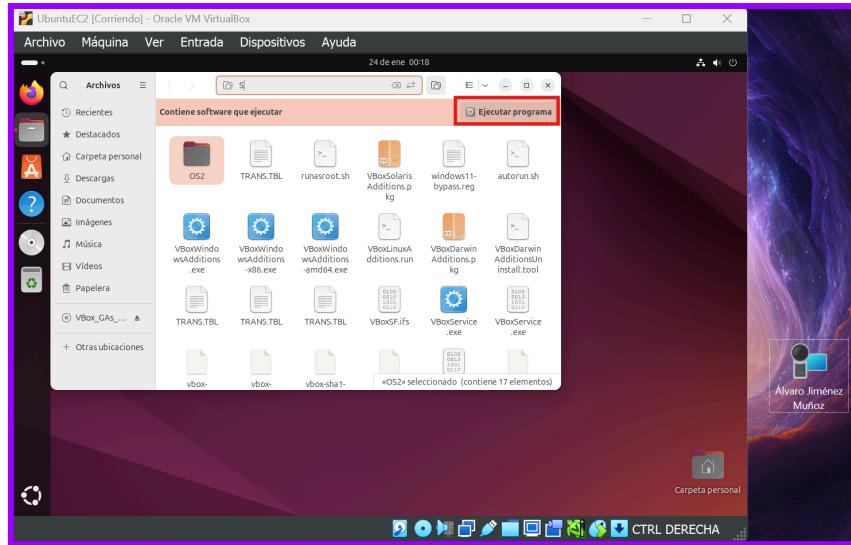


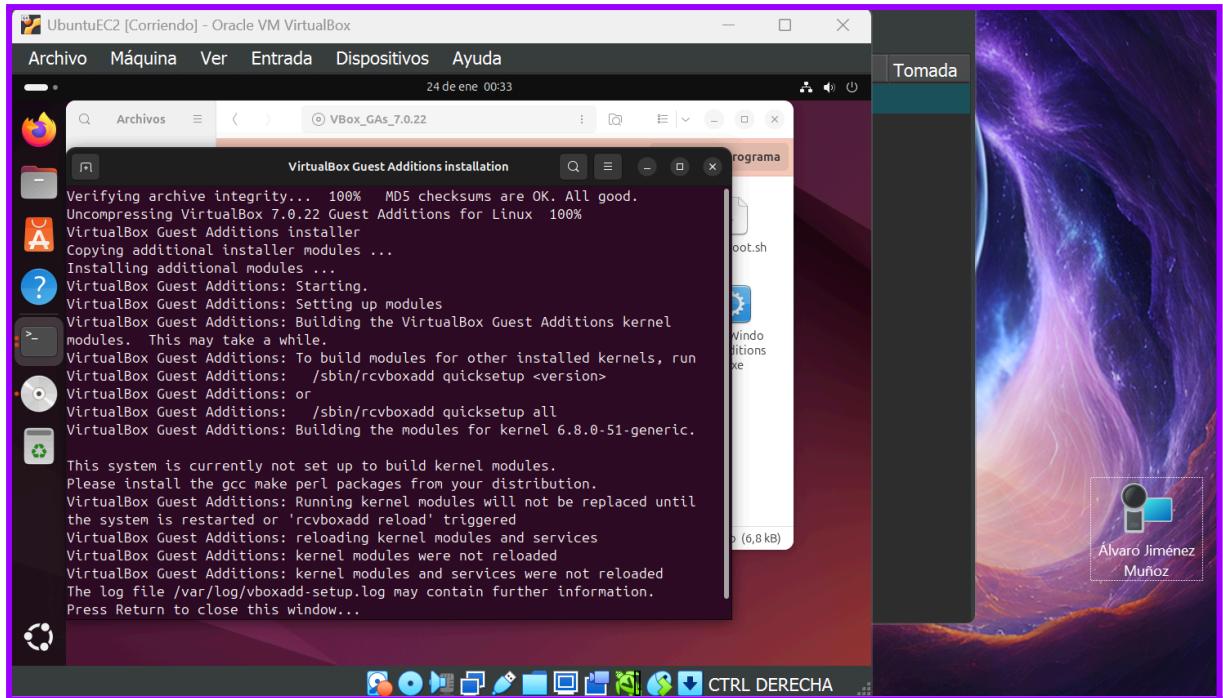
Ya para dejar lista la máquina con **Ubuntu**, se lleva a cabo toda la configuración e instalación inicial de **ubuntu** hasta llegar al escritorio:



Una vez hecho eso, antes de nada, se instalan las **guest additions**:

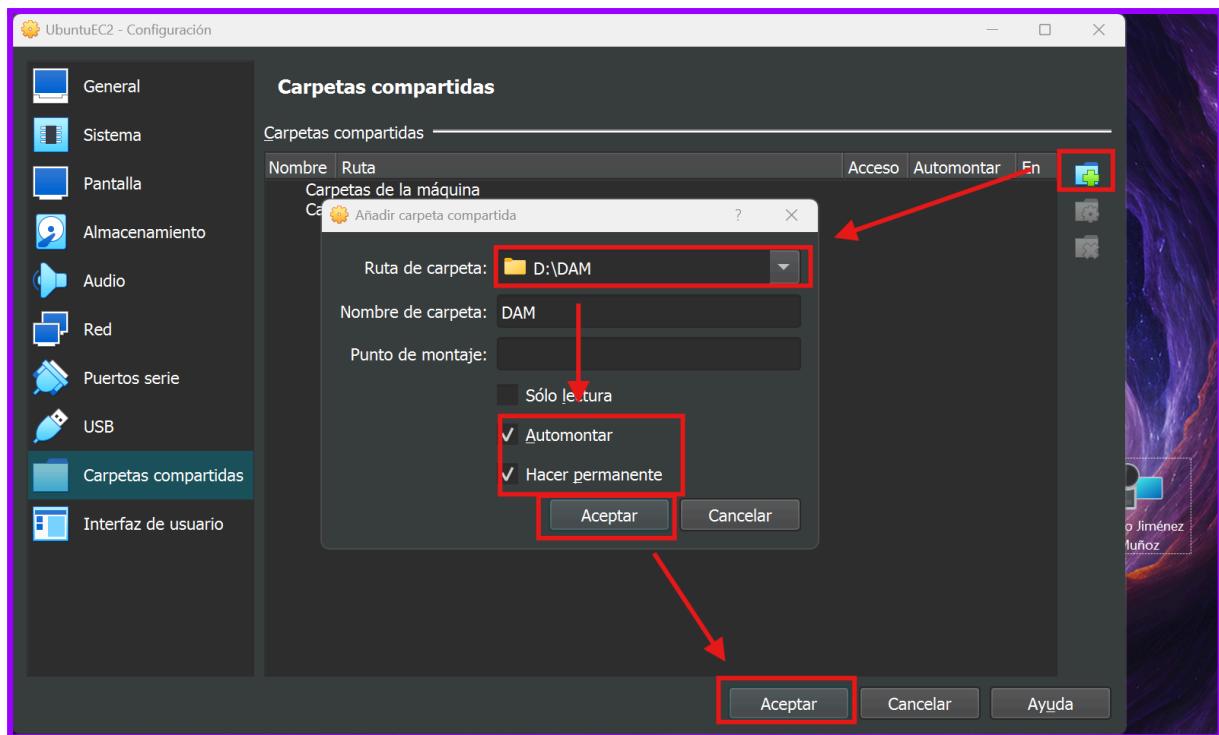




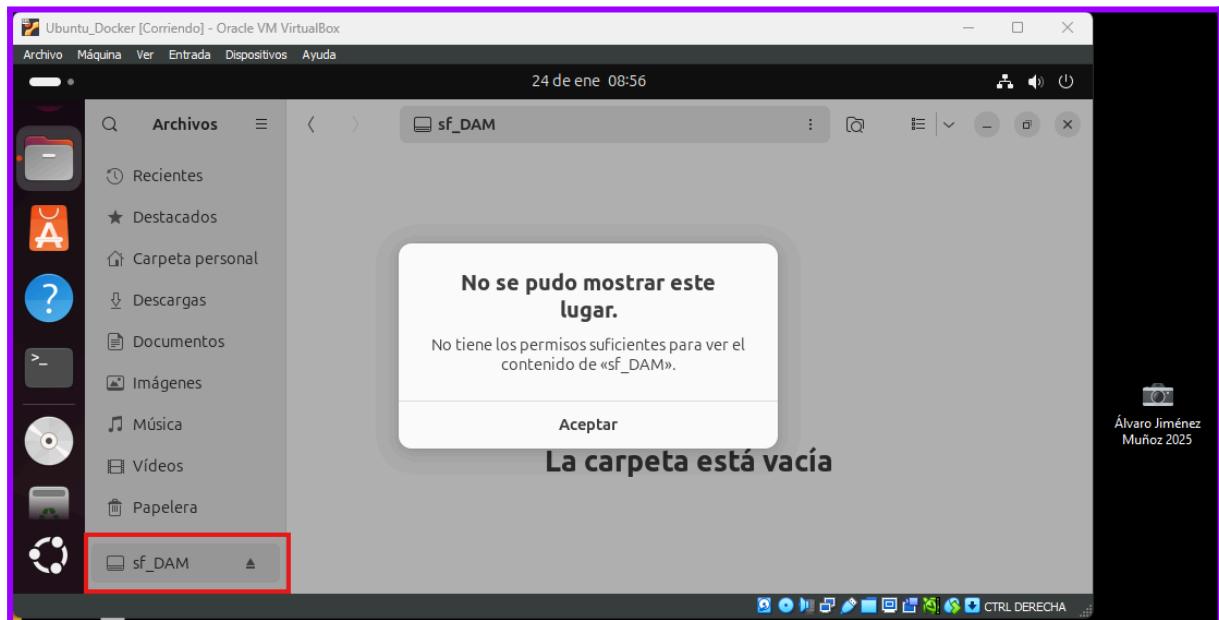


Después se reinicia y se establece una carpeta compartida, en la que se almacenará todo lo que se va a usar, como la web a desplegar y el par de claves de la instancia EC2:

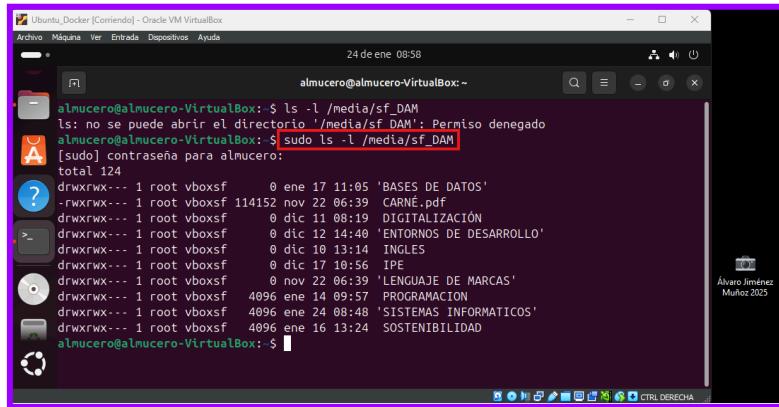




Ya, con eso hecho, la carpeta se habrá compartido entre la máquina local y la virtual, debiendo de aparecer en esta ubicación de la máquina virtual la carpeta:

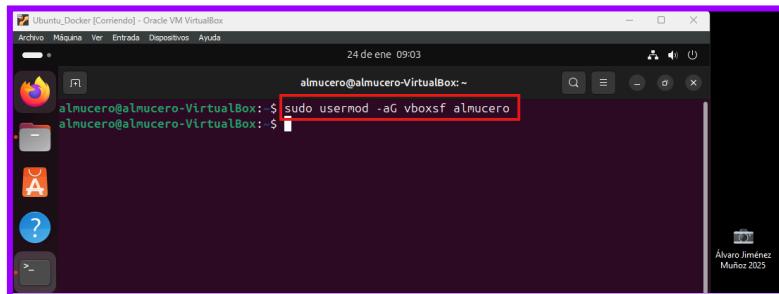


También, como se puede ver, puede llegar a decir que no hay permisos suficientes para acceder a la carpeta, a pesar de que esta aparezca. Para solucionarlo, debe hacerse lo siguiente:



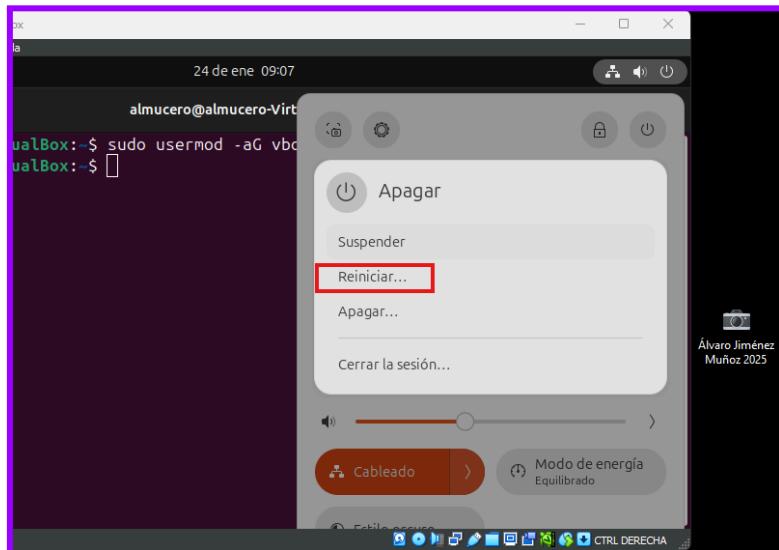
```
Ubuntu_Docker [Conriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
24 de ene 08:58
almucero@almucero-VirtualBox: ~
ls -l /media/sf_DAM
ls: no se puede abrir el directorio '/media/sf_DAM': Permiso denegado
almucero@almucero-VirtualBox: ~
[sudo] contraseña para almucero:
total 124
drwxrwx--- 1 root vboxsf    0 ene 17 11:05 'BASES DE DATOS'
-rw-rwx--- 1 root vboxsf 114152 nov 22 06:39 CARNE.pdf
drwxrwx--- 1 root vboxsf    0 dic 11 08:19 DIGITALIZACIÓN
drwxrwx--- 1 root vboxsf    0 dic 12 14:40 'ENTORNOS DE DESARROLLO'
drwxrwx--- 1 root vboxsf    0 dic 18 13:14 INGLES
drwxrwx--- 1 root vboxsf    0 dic 17 10:56 IPE
drwxrwx--- 1 root vboxsf    0 nov 22 06:39 ' LENGUAJE DE MARCAS '
drwxrwx--- 1 root vboxsf 4096 ene 14 09:57 PROGRAMACION
drwxrwx--- 1 root vboxsf 4096 ene 24 08:48 ' SISTEMAS INFORMATICOS '
drwxrwx--- 1 root vboxsf 4096 ene 16 13:24 SOSTENIBILIDAD
almucero@almucero-VirtualBox: ~
```

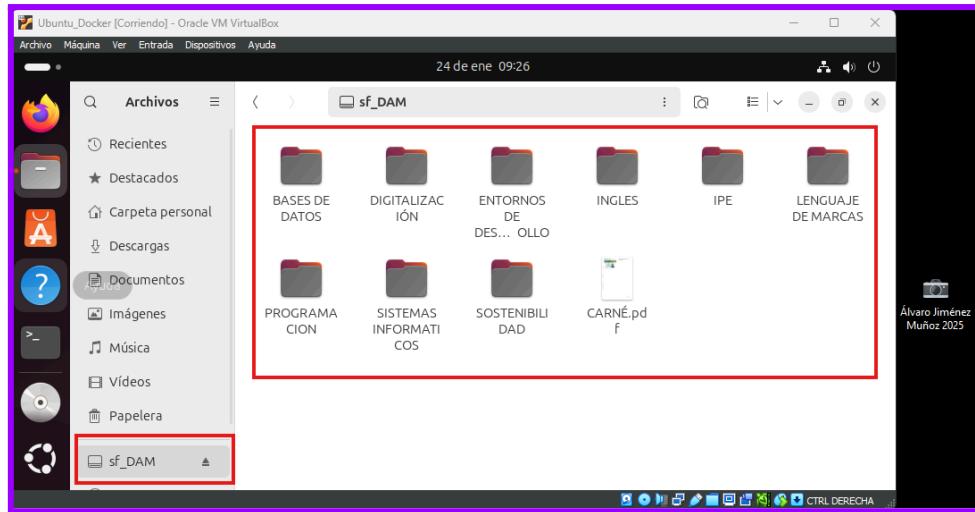
Con ese comando se comprueba antes de nada que la carpeta verdaderamente contiene los archivos que debe contener. Después se agrega el usuario al grupo **vboxsf**:



```
Ubuntu_Docker [Conriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
24 de ene 09:03
almucero@almucero-VirtualBox: ~
[sudo] contraseña para almucero:
almucero@almucero-VirtualBox: ~
```

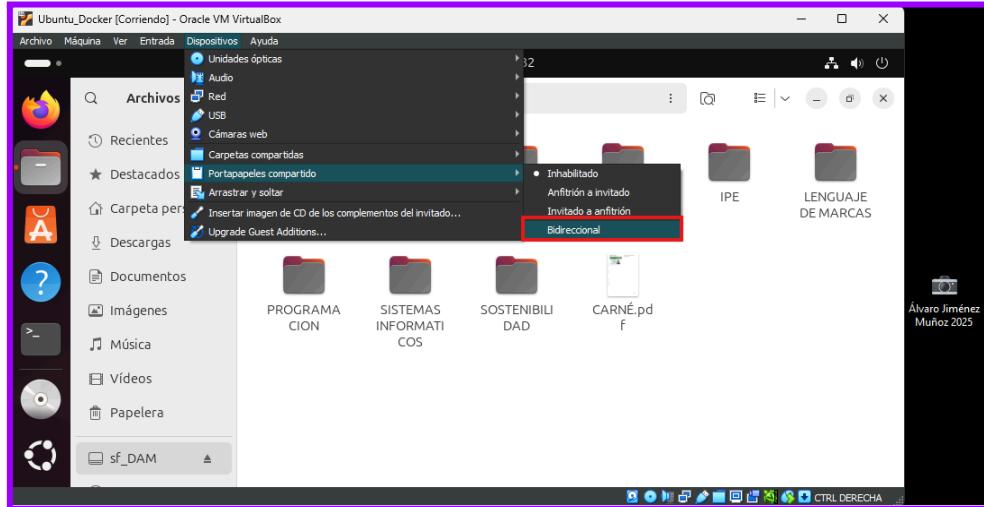
Por último, se reinicia la máquina y ya se podrá acceder a la carpeta:

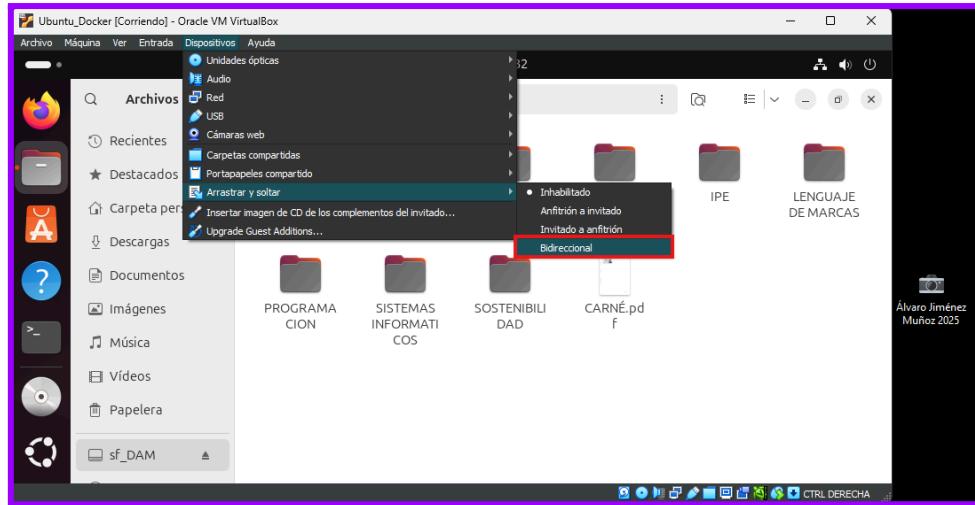




En la imagen se puede ver cómo ahora sí es posible acceder a los contenidos de la carpeta compartida, donde se encontrarán los archivos de la segunda web, el par de claves, etc.

Antes de seguir, también se activa el arrastrar y soltar y el portapapeles compartido bidireccional en la máquina, para poder copiar y pegar la IP de la instancia EC2, entre otras posibles cosas, entre la máquina local y virtual de manera sencilla:





También se deberán actualizar los paquetes de Ubuntu e instalar docker.io en la terminal de la máquina virtual, al igual que se hizo con Putty, y con ello se terminaría la preparación de esta para hacer la actividad:

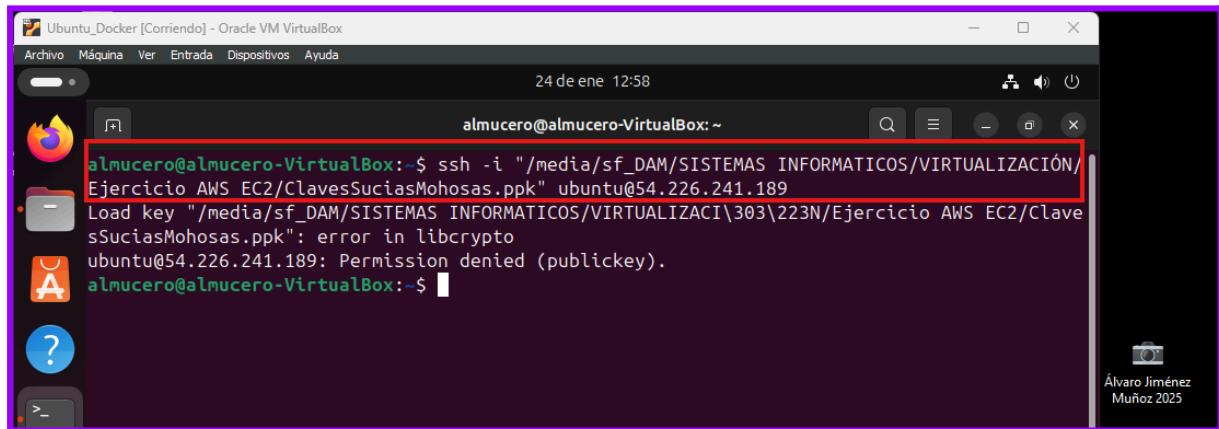
```
Ubuntu_Docker [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
24 de ene 10:04
almucero@almucero-VirtualBox: $ sudo apt update
Obj:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Obj:2 http://archive.ubuntu.com/ubuntu noble InRelease
Obj:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Obj:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 38 paquetes. Ejecute «apt list --upgradable» para verlos.
almucero@almucero-VirtualBox: $
```



```
Ubuntu_Docker [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
24 de ene 10:05
almucero@almucero-VirtualBox: ~
Se pueden actualizar 38 paquetes. Ejecute «apt list --upgradable» para verlos.
almucero@almucero-VirtualBox: ~$ sudo apt install docker.io
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  bridge-utils containerd git git-man liblrror-perl pigz runc ubuntu-fan
Paquetes sugeridos:
  ifupdown aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap docker-buildx
  docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils git-daemon-run
  | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki
  git-svn
Se instalarán los siguientes paquetes NUEVOS:
  bridge-utils containerd docker.io git git-man liblrror-perl pigz runc ubuntu-fan
0 actualizados, 9 nuevos se instalarán, 0 para eliminar y 38 no actualizados.
Se necesita descargar 84,6 MB de archivos.
Se utilizarán 327 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Dsc:1 http://archive.ubuntu.com/ubuntu/noble/universe amd64 pigz amd64 2.8.1-1 f65 6 KB
```

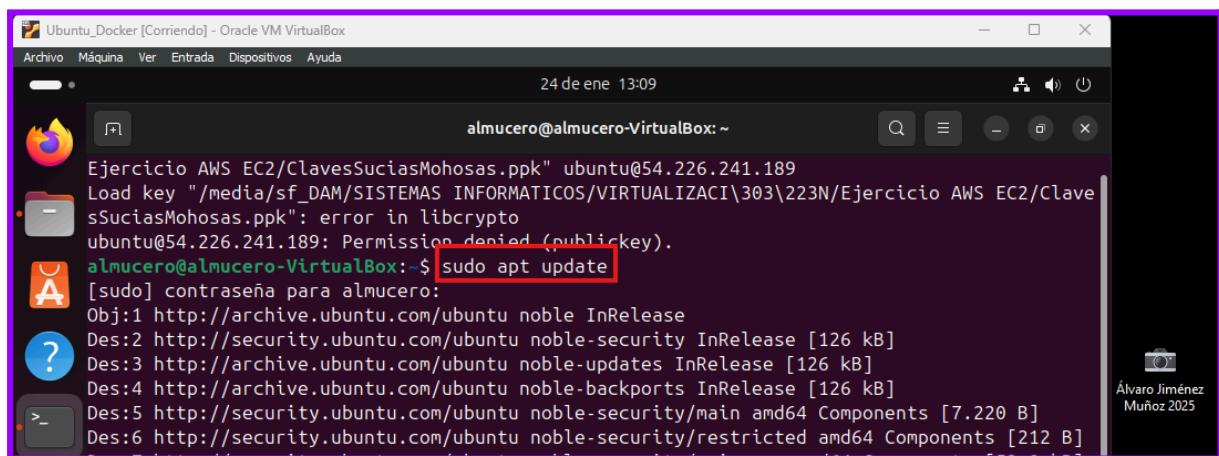
Ya, finalmente, con todo lo anterior hecho, se habrá dejado la VM completamente lista para desplegar una segunda web en la instancia EC2 desde una consola Ubuntu, pudiendo acceder a los archivos que conforman la segunda web, copiar y pegar la IP de la instancia entre los diferentes escritorios, entre otras cosas.

Por lo que, lo primero de la actividad será acceder a la instancia de EC2 desde la terminal de Ubuntu; para ello se ejecuta el siguiente comando en la terminal:



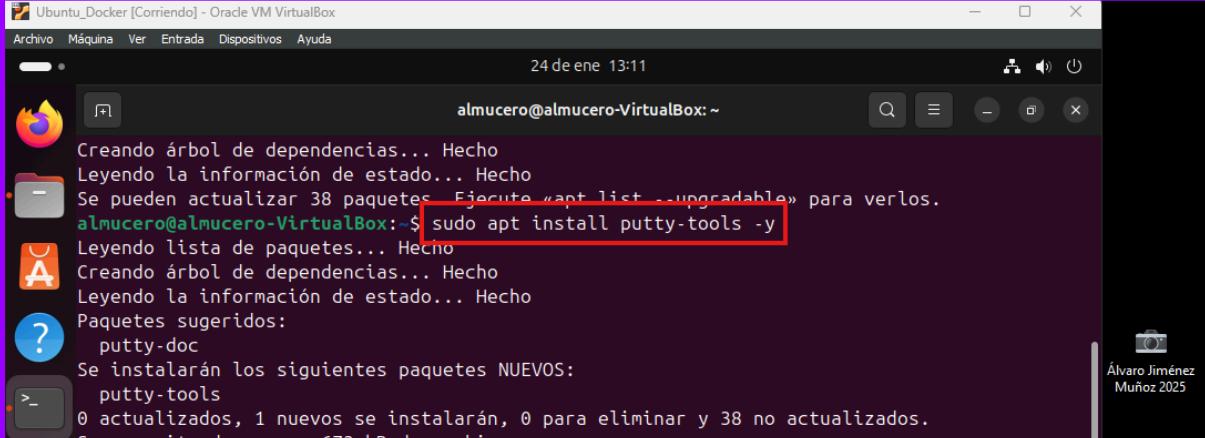
```
Ubuntu_Docker [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
24 de ene 12:58
almucero@almucero-VirtualBox:~ almucero@almucero-VirtualBox:~
almucero@almucero-VirtualBox:~$ ssh -i "/media/sf_DAM/SISTEMAS INFORMATICOS/VIRTUALIZACIÓN/Ejercicio AWS EC2/ClavesSuciasMohosas.ppk" ubuntu@54.226.241.189
Load key "/media/sf_DAM/SISTEMAS INFORMATICOS/VIRTUALIZACIÓN/303/223N/Ejercicio AWS EC2/ClaveSuciasMohosas.ppk": error in libcrypto
ubuntu@54.226.241.189: Permission denied (publickey).
almucero@almucero-VirtualBox:~$
```

En ese comando se especifica la ubicación del par de claves ubicadas dentro de la carpeta compartida en este caso, además de la IP de la instancia EC2. Al hacerlo, dará error, puesto que el formato de la clave es .ppk, y para poder hacer lo que en ese comando se está intentando, debería estar en formato .pem, por lo que se debe de crear una copia de la clave en el formato correcto; eso se hace de la siguiente manera a través de la terminal:



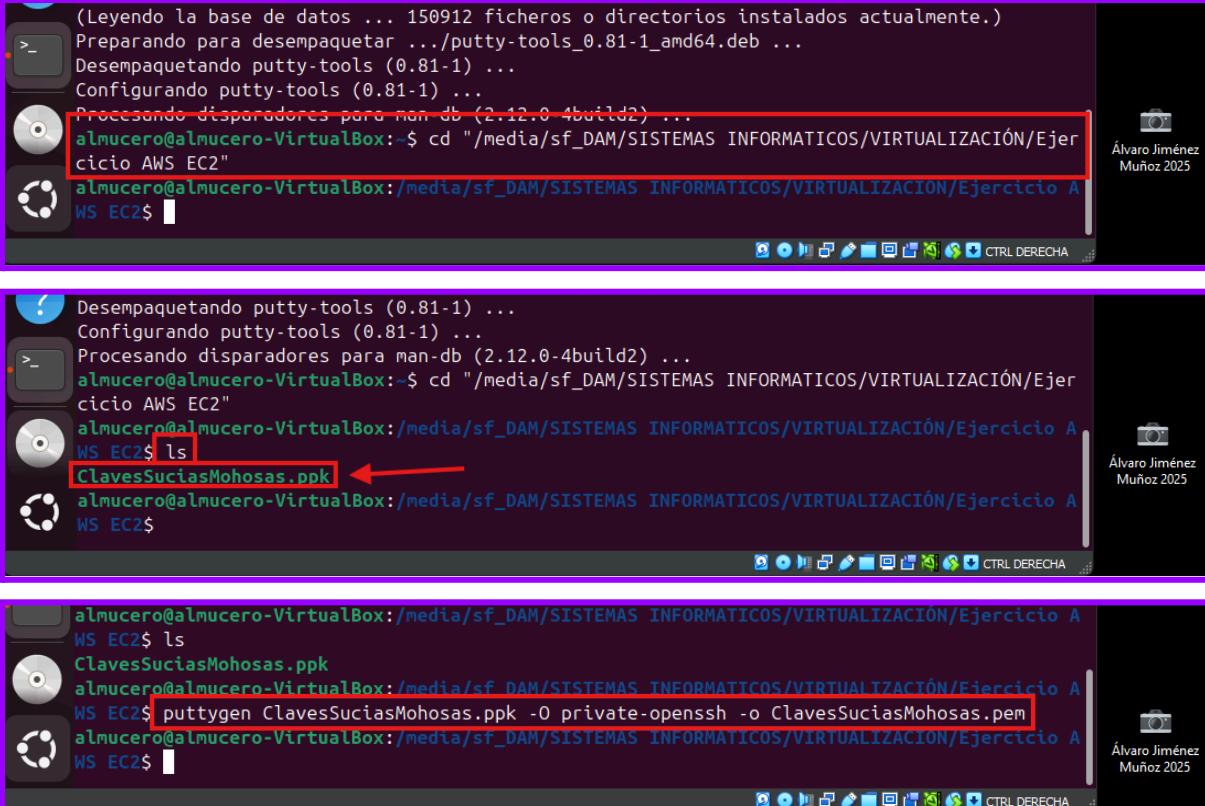
```
Ubuntu_Docker [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
24 de ene 13:09
almucero@almucero-VirtualBox:~ almucero@almucero-VirtualBox:~
almucero@almucero-VirtualBox:~$ Ejercicio AWS EC2/ClavesSuciasMohosas.ppk" ubuntu@54.226.241.189
Load key "/media/sf_DAM/SISTEMAS INFORMATICOS/VIRTUALIZACIÓN/303/223N/Ejercicio AWS EC2/ClaveSuciasMohosas.ppk": error in libcrypto
ubuntu@54.226.241.189: Permission denied (publickey).
almucero@almucero-VirtualBox:~$ sudo apt update
[sudo] contraseña para almucero:
Obj:1 http://archive.ubuntu.com/ubuntu noble InRelease
Des:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Des:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Des:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Des:5 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [7.220 B]
Des:6 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
```

Primero se actualizan los paquetes y luego se instala **puttygen**, una herramienta incluida en el paquete **putty-tools** que permite realizar el cambio de formato de la clave y que se instala así:



```
Ubuntu_Docker [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
24 de ene 13:11
almucero@almucero-VirtualBox: ~
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 38 paquetes. Ejecute «apt list --upgradable» para verlos.
almucero@almucero-VirtualBox: $ sudo apt install putty-tools -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Paquetes sugeridos:
  putty-doc
Se instalarán los siguientes paquetes NUEVOS:
  putty-tools
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 38 no actualizados.
Se necesita descargar 673 kB de archivos.
```

Una vez instalado, se accede a la ubicación de la clave con el comando **cd**, se comprueba que aparezca y se genera la nueva clave en formato **.pem** con **puttygen**:



```
(Leyendo la base de datos ... 150912 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../putty-tools_0.81-1_amd64.deb ...
Desempaquetando putty-tools (0.81-1) ...
Configurando putty-tools (0.81-1) ...
Procesando disparadores para man-db (2.12.0-4build2) ...
almucero@almucero-VirtualBox:~$ cd "/media/sf_DAM/SISTEMAS INFORMATICOS/VIRTUALIZACIÓN/Ejercicio AWS EC2"
almucero@almucero-VirtualBox:/media/sf_DAM/SISTEMAS INFORMATICOS/VIRTUALIZACIÓN/Ejercicio A WS EC2$ 
```

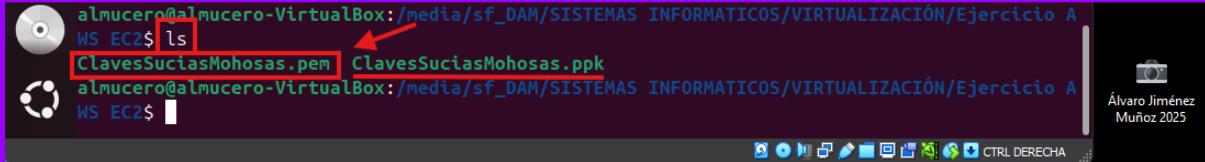


```
Desempaquetando putty-tools (0.81-1) ...
Configurando putty-tools (0.81-1) ...
Procesando disparadores para man-db (2.12.0-4build2) ...
almucero@almucero-VirtualBox:~$ cd "/media/sf_DAM/SISTEMAS INFORMATICOS/VIRTUALIZACIÓN/Ejercicio AWS EC2"
almucero@almucero-VirtualBox:/media/sf_DAM/SISTEMAS INFORMATICOS/VIRTUALIZACIÓN/Ejercicio A WS EC2$ ls
ClavesSuciasMohosas.ppk 
```



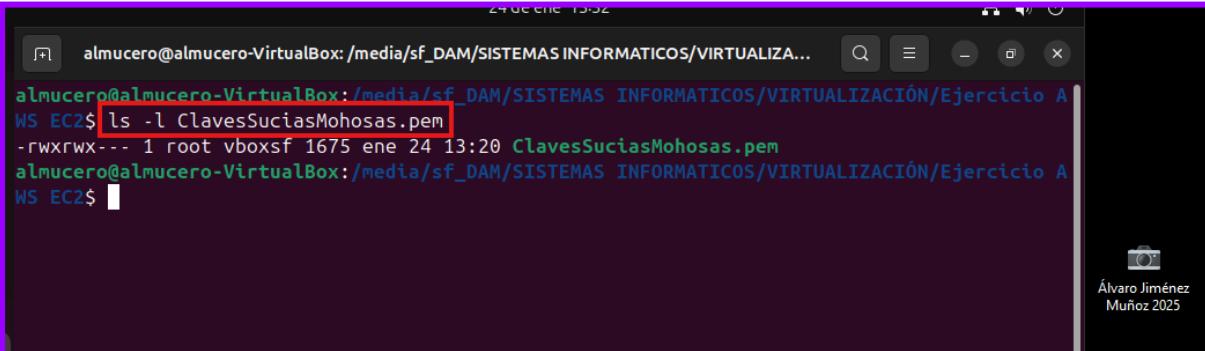
```
almucero@almucero-VirtualBox:/media/sf_DAM/SISTEMAS INFORMATICOS/VIRTUALIZACIÓN/Ejercicio A WS EC2$ ls
ClavesSuciasMohosas.ppk
almucero@almucero-VirtualBox:/media/sf_DAM/SISTEMAS INFORMATICOS/VIRTUALIZACIÓN/Ejercicio A WS EC2$ puttygen ClavesSuciasMohosas.ppk -o private-openssh -o ClavesSuciasMohosas.pem
almucero@almucero-VirtualBox:/media/sf_DAM/SISTEMAS INFORMATICOS/VIRTUALIZACIÓN/Ejercicio A WS EC2$ 
```

Con ese comando se habrá creado la clave .pem; de todas formas se comprueba si verdaderamente se ha creado por si acaso:



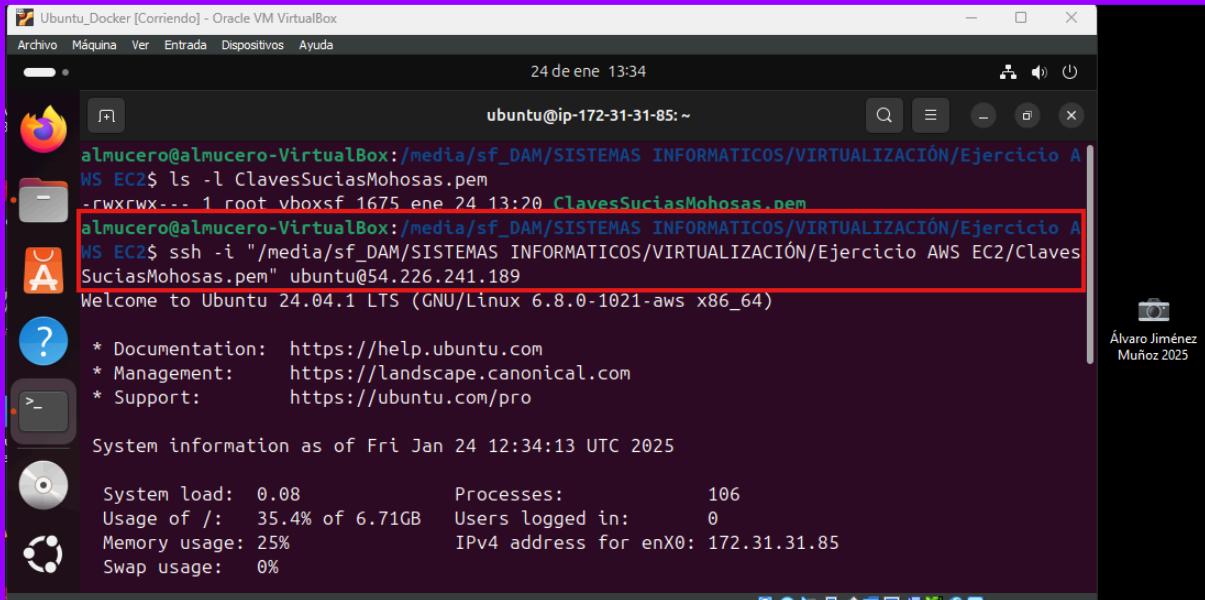
```
almucero@almucero-VirtualBox:~/media/sf_DAM/SISTEMAS INFORMATICOS/VIRTUALIZACIÓN/Ejercicio A
WS EC2$ ls
ClavesSuciasMohosas.pem ClavesSuciasMohosas.ppk
almucero@almucero-VirtualBox:~/media/sf_DAM/SISTEMAS INFORMATICOS/VIRTUALIZACIÓN/Ejercicio A
WS EC2$
```

Habiendo verificado, se mira también si tiene los permisos correctos para conectarse a la instancia con el siguiente comando:



```
24 de ene 13:32
+ almucero@almucero-VirtualBox:~/media/sf_DAM/SISTEMAS INFORMATICOS/VIRTUALIZACIÓN/Ejercicio A
WS EC2$ ls -l ClavesSuciasMohosas.pem
-rwxrwx--- 1 root vboxsf 1675 ene 24 13:20 ClavesSuciasMohosas.pem
almucero@almucero-VirtualBox:~/media/sf_DAM/SISTEMAS INFORMATICOS/VIRTUALIZACIÓN/Ejercicio A
WS EC2$
```

Después se intenta la conexión a la instancia de EC2 de nuevo de la misma forma que antes, pero con la clave .pem esta vez:

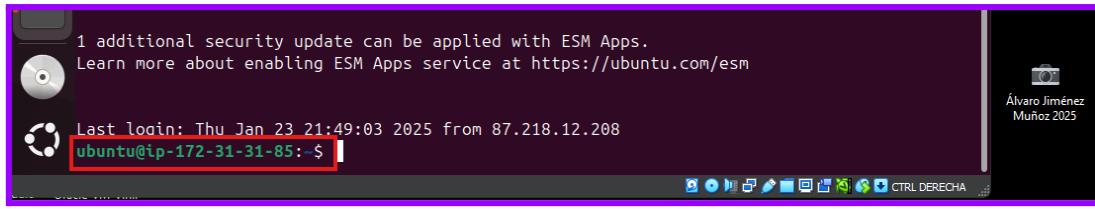


```
Ubuntu_Docker [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
24 de ene 13:34
ubuntu@ip-172-31-31-85: ~
+ almucero@almucero-VirtualBox:~/media/sf_DAM/SISTEMAS INFORMATICOS/VIRTUALIZACIÓN/Ejercicio A
WS EC2$ ls -l ClavesSuciasMohosas.pem
-rwxrwx--- 1 root vboxsf 1675 ene 24 13:20 ClavesSuciasMohosas.pem
+ almucero@almucero-VirtualBox:~/media/sf_DAM/SISTEMAS INFORMATICOS/VIRTUALIZACIÓN/Ejercicio A
WS EC2$ ssh -i "/media/sf_DAM/SISTEMAS INFORMATICOS/VIRTUALIZACIÓN/Ejercicio AWS EC2/ClavesSuciasMohosas.pem" ubuntu@54.226.241.189
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

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

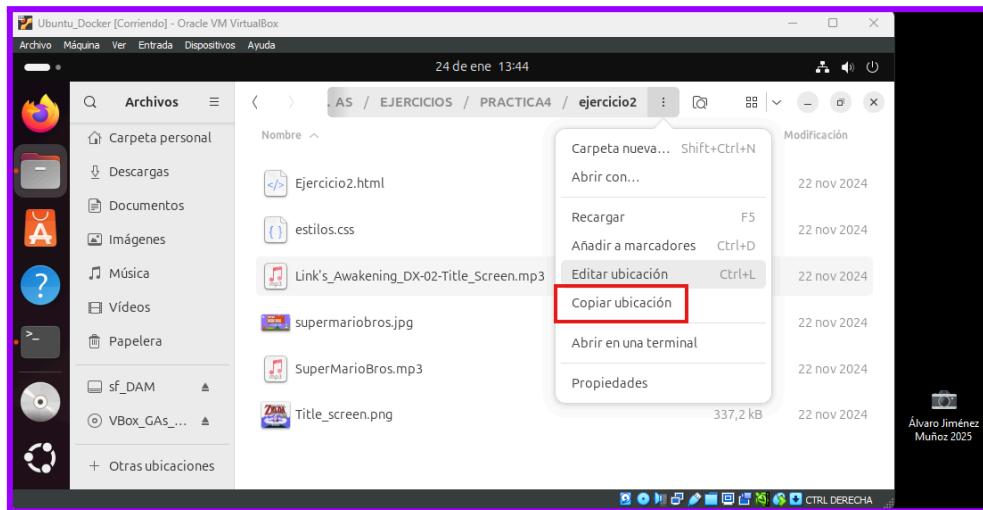
 System information as of Fri Jan 24 12:34:13 UTC 2025

 System load:  0.08           Processes:          106
 Usage of /:   35.4% of 6.71GB  Users logged in:     0
 Memory usage: 25%            IPv4 address for enX0: 172.31.31.85
 Swap usage:   0%
```

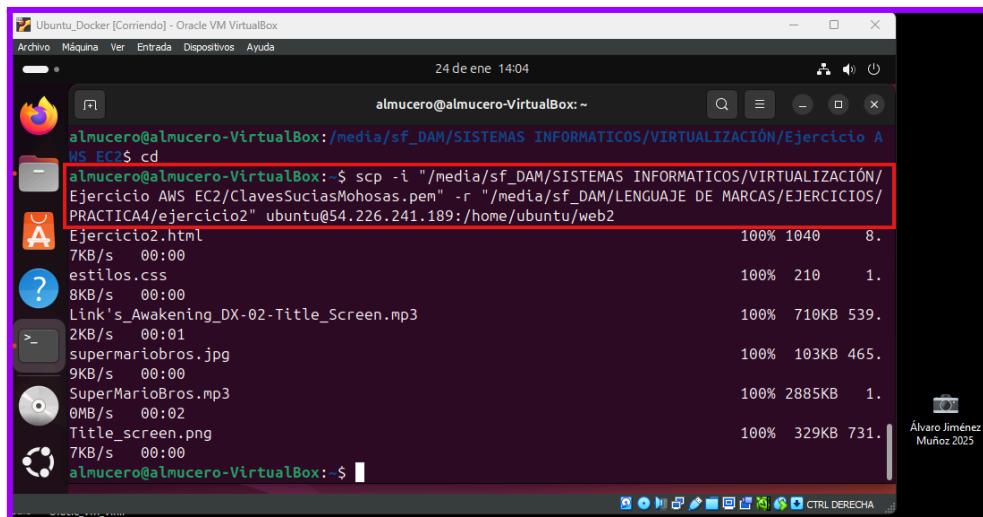


Aquí puede verse cómo ahora sí ha sido posible conectarse y no ha dado error.

Lo siguiente es ubicar y copiar la dirección de otra página web a desplegar dentro de la carpeta compartida:

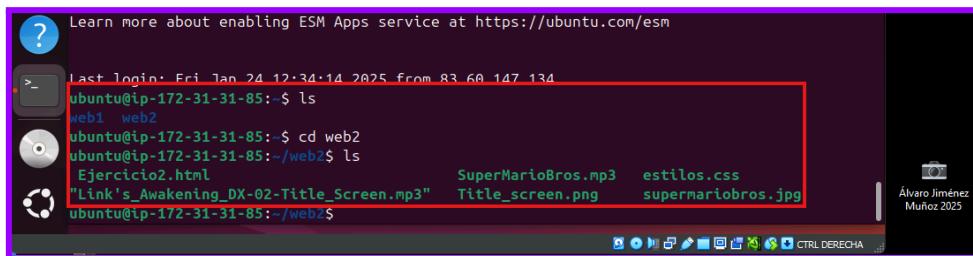


Copiada la dirección, se vuelve a la terminal y se escribe el siguiente comando, especificando dicha ubicación, dirección de la clave, IP de la instancia EC2 y el lugar dentro de la instancia a donde se copiarán los archivos que componen la web:



Accediendo de nuevo a la instancia, se puede ver cómo se ha creado el directorio **web2** junto con todos sus contenidos:

(debe haber un archivo **index.html** para hacer los pasos siguientes)



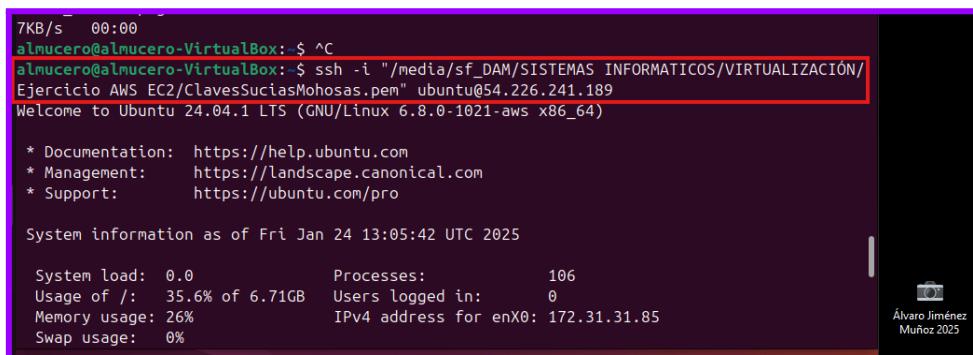
```
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

ubuntu@ip-172-31-31-85:~$ ls
web1  web2

ubuntu@ip-172-31-31-85:~$ cd web2
ubuntu@ip-172-31-31-85:~/web2$ ls
Ejercicio2.html      SuperMarioBros.mp3    estilos.css
"Link's_Awakening_DX-02-Title_Screen.mp3" Title_screen.png  supernariobros.jpg
ubuntu@ip-172-31-31-85:~/web2$
```

Después, para ejecutar un contenedor Docker con **nginx:perl** para ejecutar la página web en el puerto **8080**, se debe de hacer lo siguiente:

Acceder de nuevo a la instancia EC2:



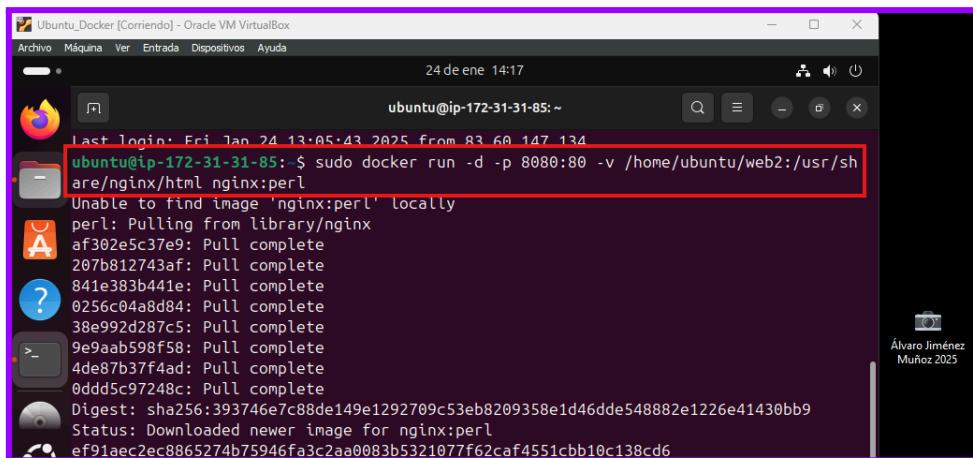
```
7KB/s  00:00
almucero@almucero-VirtualBox:~$ ^C
almucero@almucero-VirtualBox:~$ ssh -i "/media/sf_DAM/SISTEMAS INFORMATICOS/VIRTUALIZACIÓN/Ejercicio AWS EC2/ClavesSuciasMohosas.pem" ubuntu@54.226.241.189
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

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

System information as of Fri Jan 24 13:05:42 UTC 2025

 System load:  0.0          Processes:           106
 Usage of /:   35.6% of 6.71GB   Users logged in:     0
 Memory usage: 26%           IPv4 address for enX0: 172.31.31.85
 Swap usage:   0%
```

Y ahí dentro ejecutar el siguiente comando:



```
Ubuntu_Docker [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
24 de ene 14:17
ubuntu@ip-172-31-31-85:~$ sudo docker run -d -p 8080:80 -v /home/ubuntu/web2:/usr/share/nginx/html nginx:perl
Unable to find image 'nginx:perl' locally
perl: Pulling from library/nginx
af302e5c37e9: Pull complete
207b812743af: Pull complete
841e383b441e: Pull complete
0256c04a8d84: Pull complete
38e992d287c5: Pull complete
9e9aab598f58: Pull complete
4de87b37ff4ad: Pull complete
0ddd5c97248c: Pull complete
Digest: sha256:393746e7c88de149e1292709c53eb8209358e1d46dde548882e1226e41430bb9
Status: Downloaded newer image for nginx:perl
ef91aec2ec8865274b75946fa3c2aa0083b5321077f62caf4551ccb10c138cd6
```

Con eso hecho, la página ya se debería haber lanzado, usando docker ps se puede comprobar:

```

4de87b37f4ad: Pull complete
0ddd5c97248c: Pull complete
Digest: sha256:393746e7c88de149e1292709c53eb8209358e1d46dde548882e1226e41430bb9
Status: Downloaded newer image for nginx:perl
ef91aec2ec8865274b75946fa3c2aa083b5321077f62caf4551ccb10c138cd6
ubuntu@ip-172-31-31-85:~$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
ef91aec2ec88 nginx:perl "/docker-entrypoint..." 51 seconds ago Up 50 seconds
0.0.0.0:8080->80/tcp, :::8080->80/tcp compassionate_rosalind
ubuntu@ip-172-31-31-85:~$ 

```

Finalmente, se deben editar las reglas de entrada de la instancia para que permita el tráfico en el puerto 8080:

The screenshots show the AWS CloudFront console interface for managing security groups and ingress rules.

Screenshot 1: Security Groups

- The top section shows the security group details for instance **i-03b335facd0cc074b**. The **Seguridad** tab is selected.
- A red box highlights the security group entry: **sg-06d686efd8edcda8b (launch-wizard-4)**.

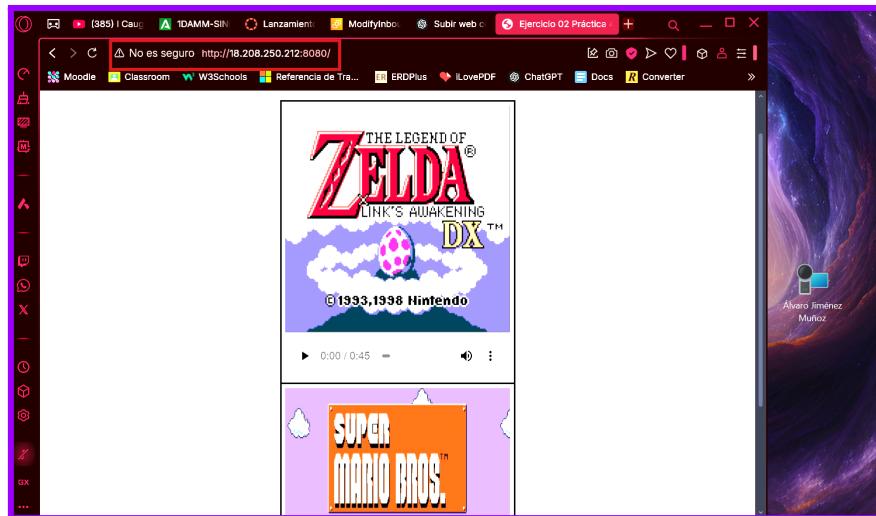
Screenshot 2: Ingress Rules

- The bottom section shows the **Reglas de entrada** (Ingress Rules) tab.
- A red box highlights the **Editar reglas de entrada** (Edit Ingress Rules) button.
- The table lists four existing rules:

Name	ID de la regla del grupo	Versión de IP	Tipo
-	sgr-085eda2a001007e29	IPv4	TCP personalizado
-	sgr-002991fe431810d52	IPv4	SSH
-	sgr-058b0b2d7bf4be621	IPv4	HTTPS
-	sgr-0c546f9c063962b3a	IPv4	HTTP

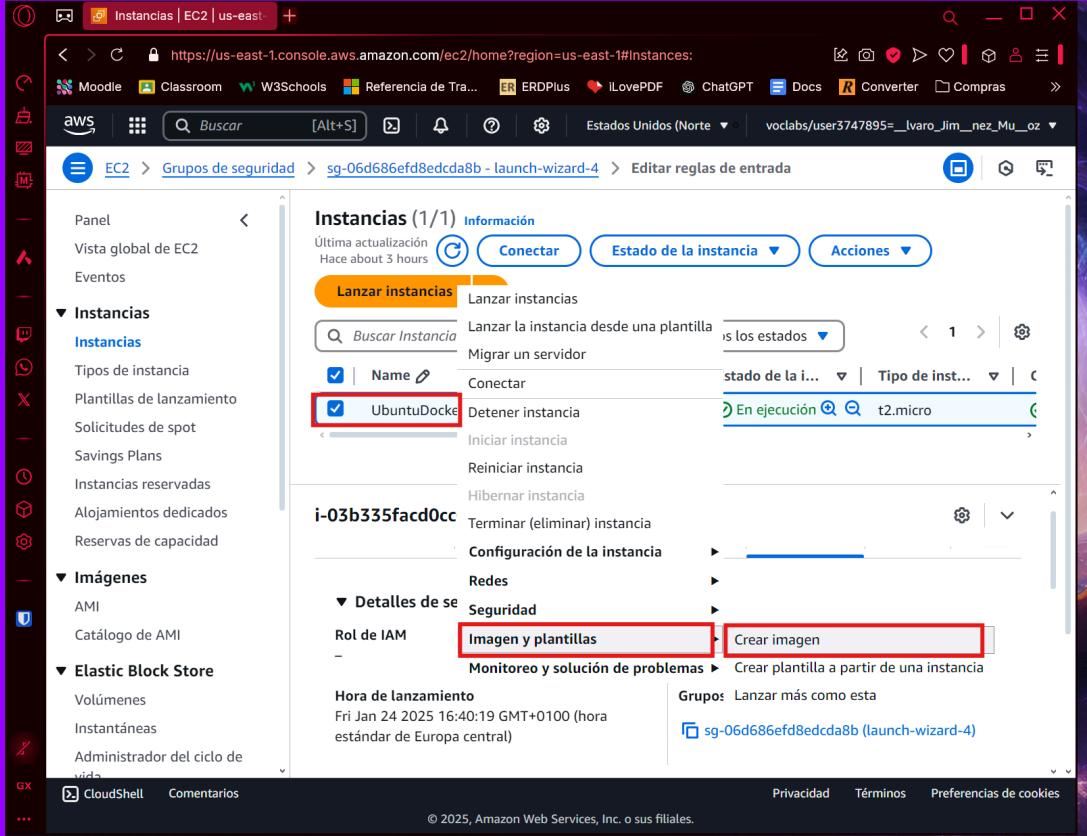
The screenshots show the configuration of a firewall rule. The top one is a general rule creation screen with fields for port range (80), origin type (Personalized), and destination (0.0.0.0/0). The bottom one is a detailed rule entry for port 8080, where the port range (8080), origin type (Anywhere-IPv4), and destination (0.0.0.0/0) are all highlighted with red boxes.

Ahora sí se podrá acceder a la web desde el navegador, usando la IP de la instancia seguida de :8080:

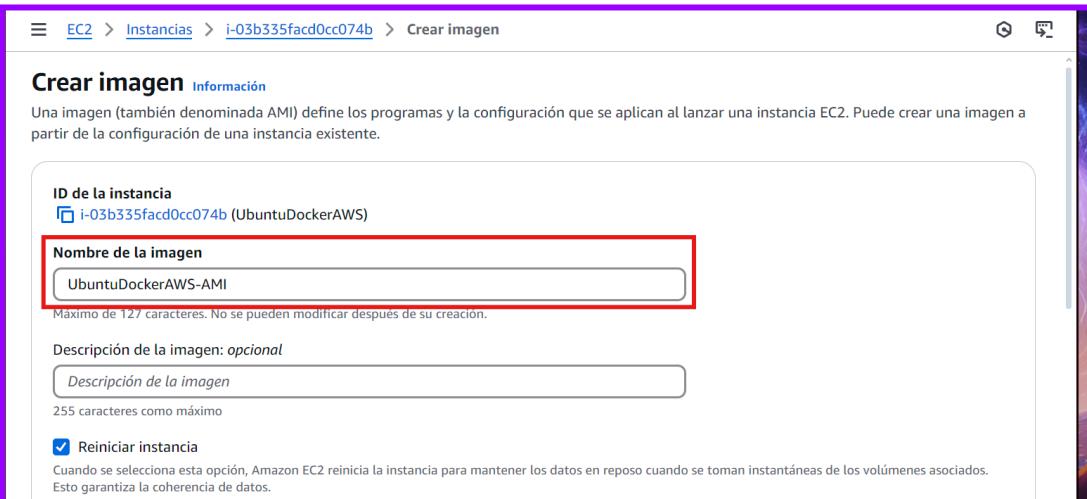


Actividad 4

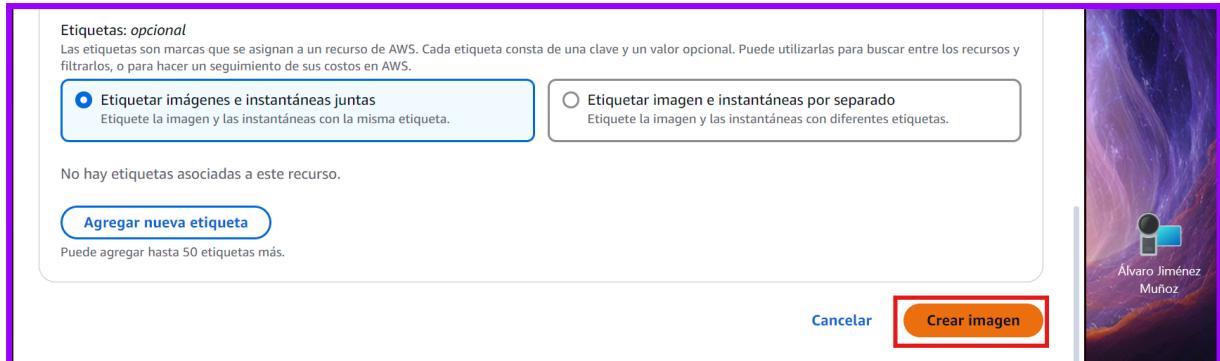
Para crear una nueva **AMI** en base a la instancia usada anteriormente, “UbuntuDockerAWS”, se debe de hacer lo siguiente en la página web de **Amazon EC2**:



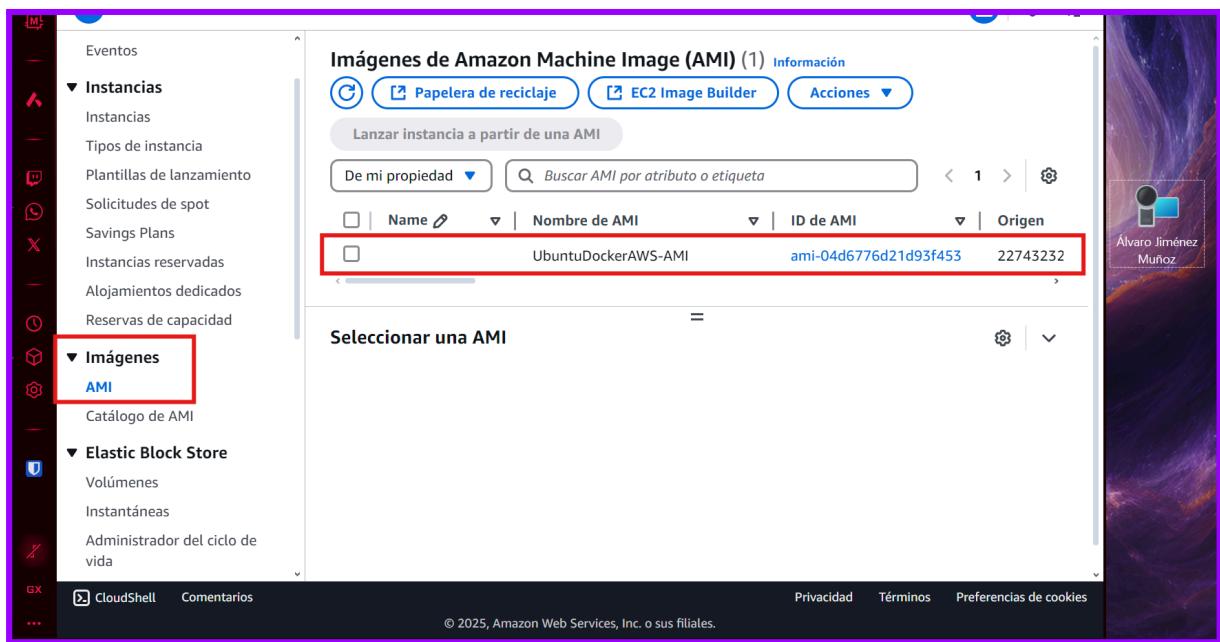
The screenshot shows the AWS EC2 Instances page. On the left sidebar, under 'Instancias', the 'Instancias' section is selected. In the main content area, the 'Lanzar instancias' section is highlighted. A checkbox labeled 'UbuntuDocker' is checked and highlighted with a red box. Below it, another checkbox labeled 'UbuntuDocke' is also checked. To the right, there's a table showing an instance named 'i-03b335facd0cc'. Under the 'Imagen y plantillas' tab, the 'Crear imagen' button is highlighted with a red box.



The screenshot shows the 'Crear imagen' (Create Image) wizard. The first step, 'Información' (Information), is displayed. It shows the instance ID 'i-03b335facd0cc074b (UbuntuDockerAWS)' and a text input field for the 'Nombre de la imagen' (Image name) containing 'UbuntuDockerAWS-AMI', which is also highlighted with a red box. Below the input field is a note: 'Maximo de 127 caracteres. No se pueden modificar después de su creación.' (Maximum of 127 characters. Cannot be modified after creation.)

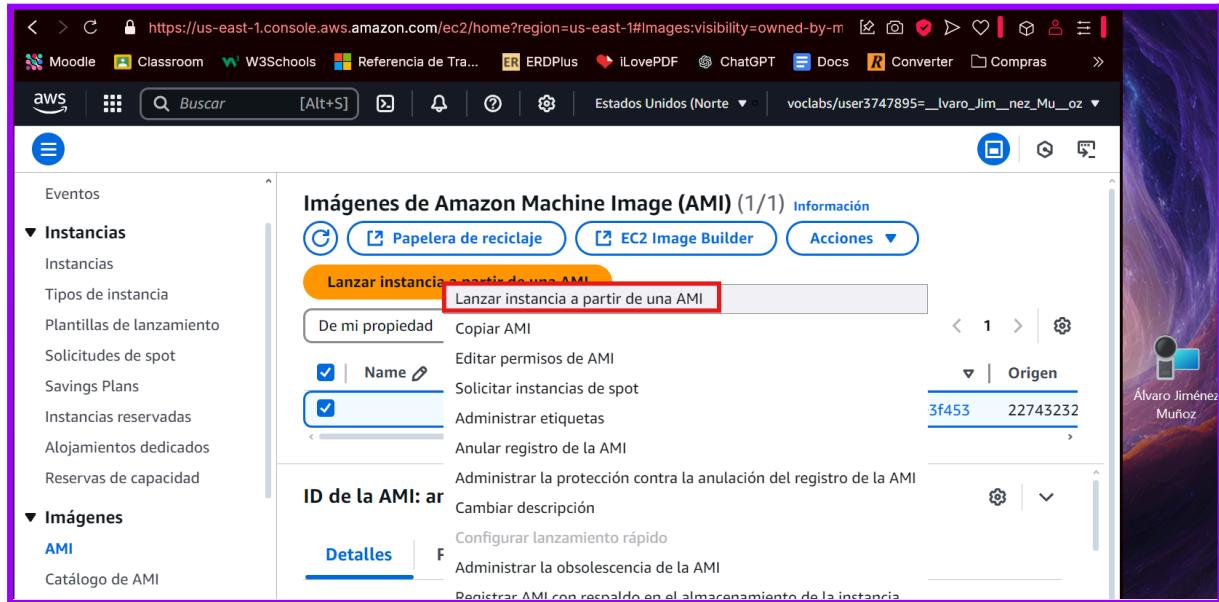


Con eso ya se habría creado la AMI y en esta pestaña se puede verificar su creación:



Actividad 5

Para crear una nueva instancia EC2 llamada “UbuntuDockerComposeAWS” con las mismas características de la anterior se debe de hacer lo siguiente con la AMI anteriormente creada:



Ahora se hacen las configuraciones necesarias:

The image contains two screenshots of the AWS EC2 'Launch instance' wizard.

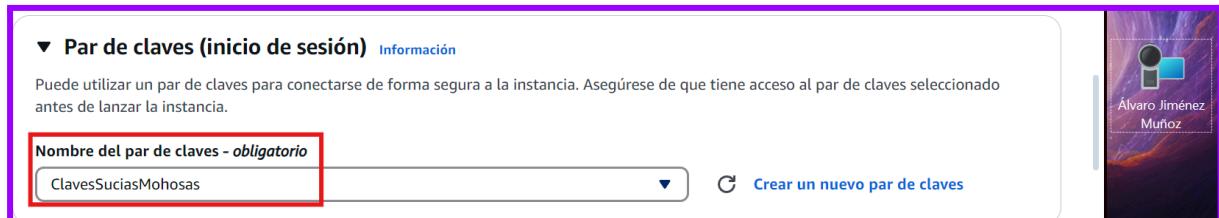
Screenshot 1: Nombre y etiquetas

This step shows the 'Nombre y etiquetas' section. A red box highlights the 'Nombre' input field, which contains 'UbuntuDockerComposeAWS'. To the right, there's a 'Agregar etiquetas adicionales' link.

Screenshot 2: Tipos de instancia

This step shows the 'Tipo de instancia' section. A red box highlights the 't2.micro' option in the 'Tipo de instancia' dropdown. Below it, a box contains detailed information about the t2.micro instance type, including its family (t2), vCPU count (1), memory (1 GiB), generation (true), and base prices for various operating systems. To the right, there are buttons for 'Todas las generaciones' and 'Comparar tipos de instancias', and a note at the bottom stating 'Se aplican costos adicionales a las AMI con software preinstalado'.

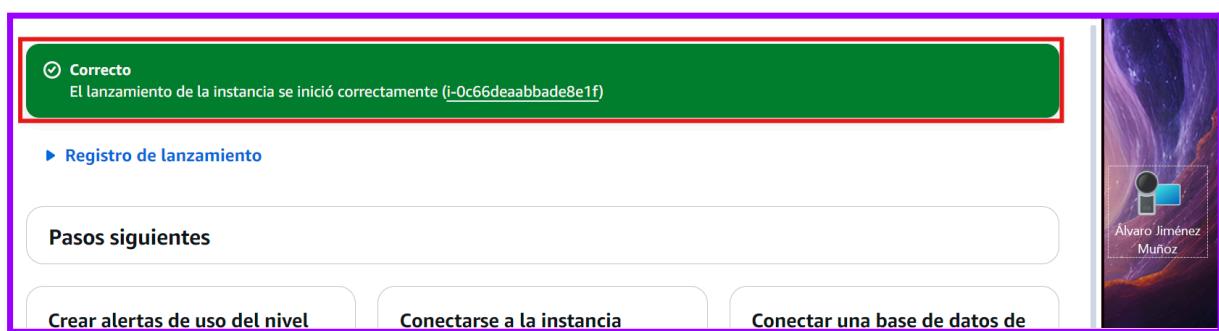
Se usa el mismo par de claves:



Y se configura el grupo de seguridad:

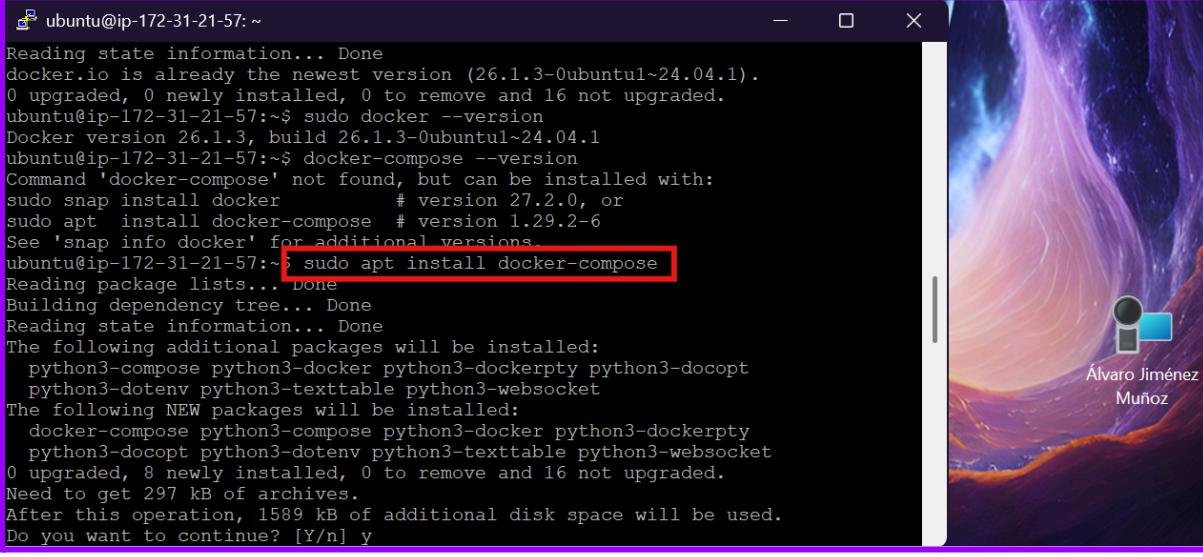


Por último se lanza la instancia:



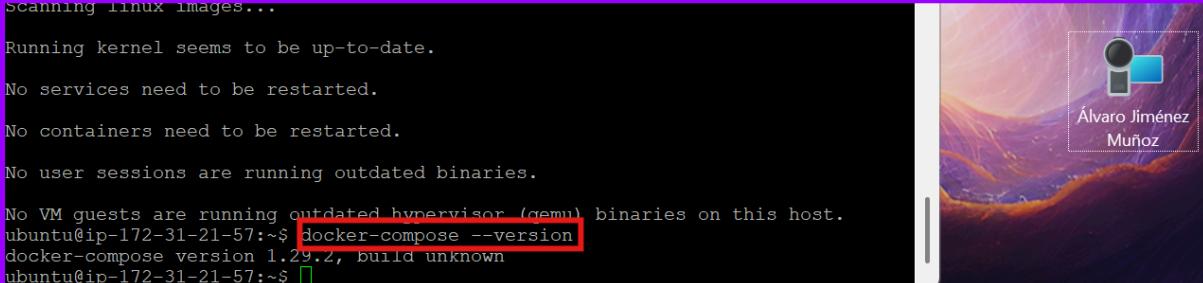
Para desplegar un wordpress con **Docker Compose** con mariadb en el puerto **9000**, se debe, antes de nada hacer la conexión con **Putty**, actualizar paquetes, instalar **docker.io** y todo lo que anteriormente se hizo de la exacta misma forma, por lo que no se explicará esa parte.

Habiendo hecho ya todo lo mencionado, se instala **Docker Compose** de la siguiente forma desde la terminal de Putty:



```
ubuntu@ip-172-31-21-57:~$ Reading state information... Done
docker.io is already the newest version (26.1.3-0ubuntu1~24.04.1).
0 upgraded, 0 newly installed, 0 to remove and 16 not upgraded.
ubuntu@ip-172-31-21-57:~$ sudo docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
ubuntu@ip-172-31-21-57:~$ docker-compose --version
Command 'docker-compose' not found, but can be installed with:
sudo snap install docker            # version 27.2.0, or
sudo apt install docker-compose     # version 1.29.2-6
See 'snap info docker' for additional versions.
ubuntu@ip-172-31-21-57:~$ sudo apt install docker-compose
Reading package lists... done
Building dependency tree... done
Reading state information... done
The following additional packages will be installed:
  python3-compose python3-docker python3-dockerpty python3-docopt
  python3-dotenv python3-texttable python3-websocket
The following NEW packages will be installed:
  docker-compose python3-compose python3-docker python3-dockerpty
  python3-docopt python3-dotenv python3-texttable python3-websocket
0 upgraded, 8 newly installed, 0 to remove and 16 not upgraded.
Need to get 297 kB of archives.
After this operation, 1589 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

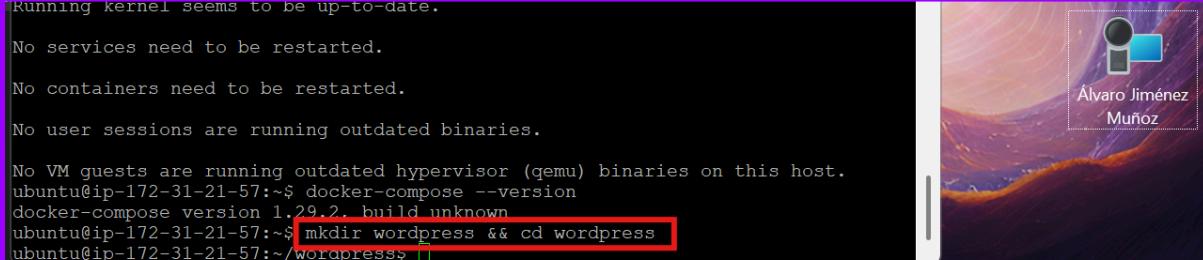
Se verifica la instalación:



```
Scanning for Linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-21-57:~$ docker-compose --version
docker-compose version 1.29.2, build unknown
ubuntu@ip-172-31-21-57:~$
```

Y después, se crea y se accede un directorio en el que se almacenará el archivo **docker-compose.yml** que almacenará al **wordpress**:



```
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-21-57:~$ docker-compose --version
docker-compose version 1.29.2, build unknown
ubuntu@ip-172-31-21-57:~$ mkdir wordpress && cd wordpress
ubuntu@ip-172-31-21-57:~/wordpress$
```

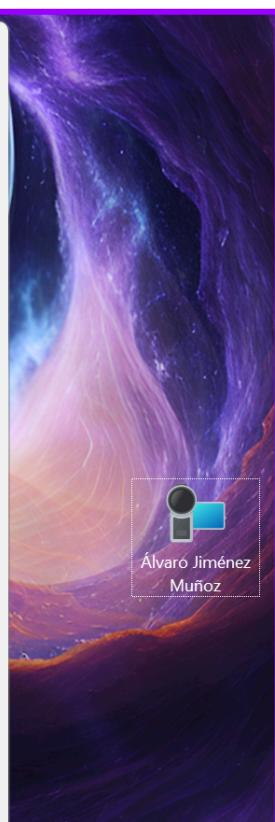
Estando dentro se crea dicho archivo:

```
No services need to be restarted.  
No containers need to be restarted.  
No user sessions are running outdated binaries.  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
ubuntu@ip-172-31-21-57:~$ docker-compose --version  
docker-compose version 1.29.2, build unknown  
ubuntu@ip-172-31-21-57:~$ mkdir wordpress && cd wordpress  
ubuntu@ip-172-31-21-57:~/wordpress$ nano docker-compose.yml  
ubuntu@ip-172-31-21-57:~/wordpress$
```



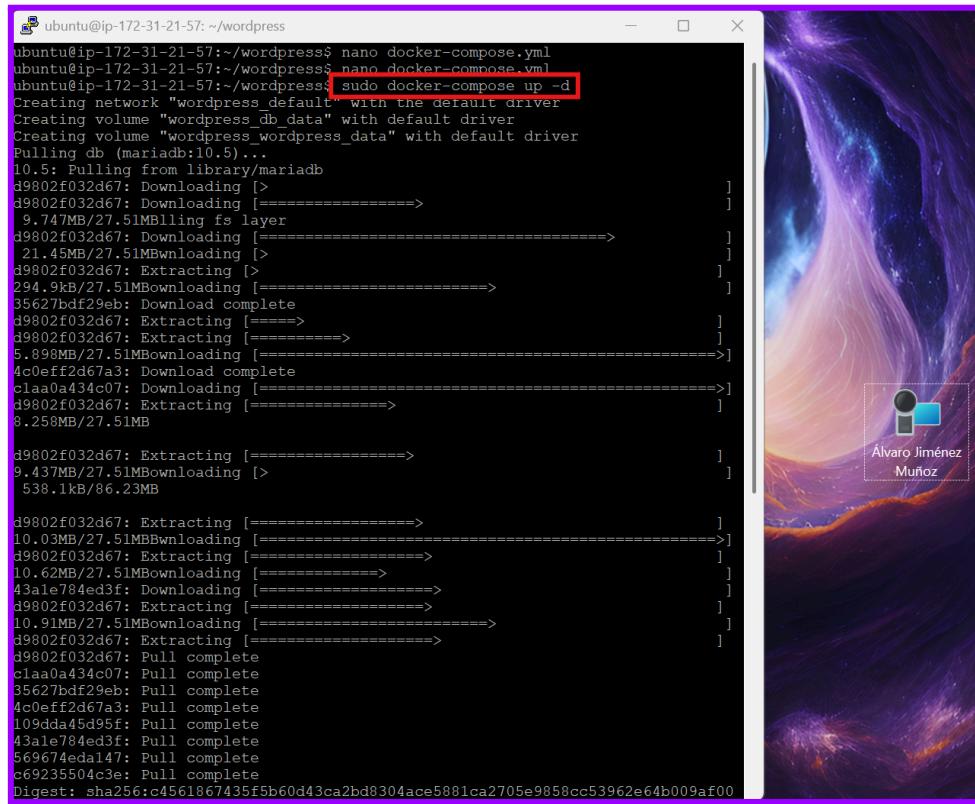
Al acceder se edita su contenido añadiendo la siguiente información:

```
ubuntu@ip-172-31-21-57:~/wordpress$ nano docker-compose.yml *  
version: '3.3'  
  
services:  
  db:  
    image: mariadb:10.5  
    container_name: mariadb  
    restart: always  
    environment:  
      MYSQL_ROOT_PASSWORD: rootpassword  
      MYSQL_DATABASE: wordpress  
      MYSQL_USER: wpuser  
      MYSQL_PASSWORD: wppassword  
    volumes:  
      - db_data:/var/lib/mysql  
  
  wordpress:  
    image: wordpress:latest  
    container_name: wordpress  
    restart: always  
    ports:  
      - "9000:80"  
    environment:  
      WORDPRESS_DB_HOST: db  
      WORDPRESS_DB_USER: wpuser  
      WORDPRESS_DB_PASSWORD: wppassword  
      WORDPRESS_DB_NAME: wordpress  
    depends_on:  
      - db  
    volumes:  
      - wordpress_data:/var/www/html  
  
volumes:  
  db_data:  
  wordpress_data:
```



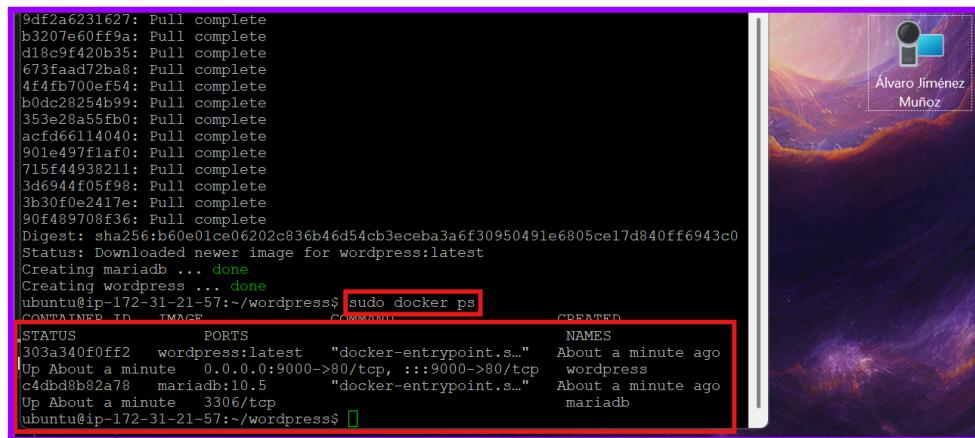
Después se guarda el archivo y se cierra.

Con eso hecho, se inicia el servicio con **Docker Compose**, con el siguiente comando en **Putty**, que se encargará de descargar las imágenes de **Wordpress** y **MariaDB**, levantando ambos servicios:



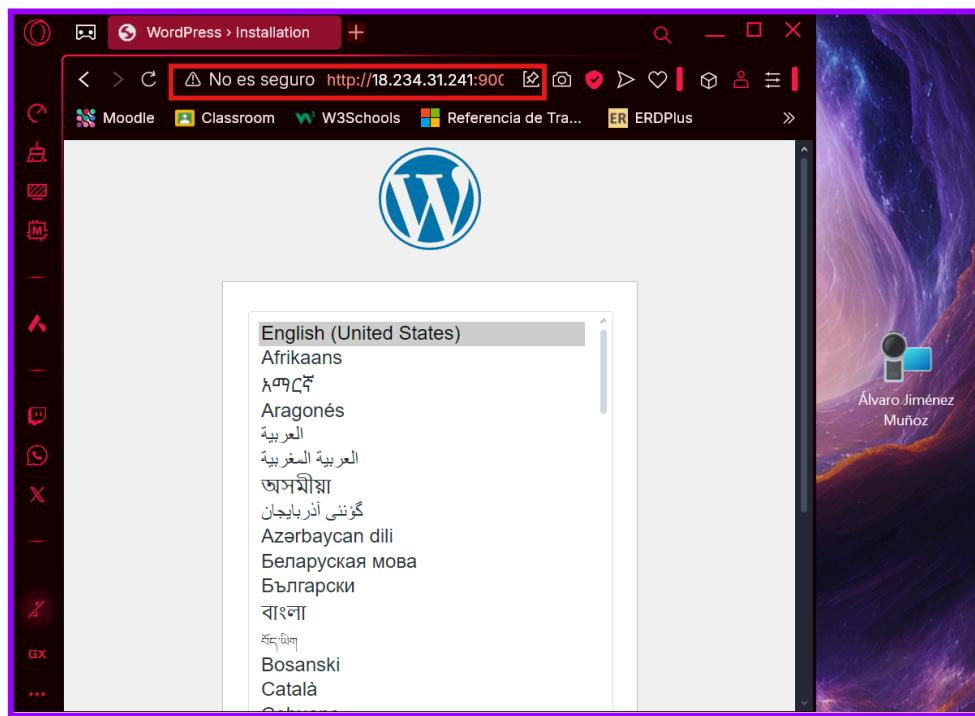
```
ubuntu@ip-172-31-21-57:~/wordpress$ nano docker-compose.yml
ubuntu@ip-172-31-21-57:~/wordpress$ nano docker-compose.yml
ubuntu@ip-172-31-21-57:~/wordpress$ sudo docker-compose up -d
Creating network "wordpress_default" with the default driver
Creating volume "wordpress_db_data" with default driver
Creating volume "wordpress_wordpress_data" with default driver
Pulling db (mariadb:10.5)...
10.5: Pulling from library/mariadb
d9802f032d67: Downloading [>=====]
9.747MB/27.51MB
d9802f032d67: Downloading [>=====]
21.45MB/27.51MB
d9802f032d67: Extracting [>=====]
294.9KB/27.51MB
d9802f032d67: Downloading [>=====]
35627bdf29eb: Download complete
d9802f032d67: Extracting [>=====]
d9802f032d67: Extracting [>=====]
5.898MB/27.51MB
d9802f032d67: Downloading [>=====]
4c0eff2d67a3: Download complete
claa0a434c07: Downloading [>=====]
d9802f032d67: Extracting [>=====]
8.258MB/27.51MB
d9802f032d67: Extracting [>=====]
9.437MB/27.51MB
d9802f032d67: Downloading [>=====]
538.1kB/86.23MB
d9802f032d67: Extracting [>=====]
10.03MB/27.51MB
d9802f032d67: Extracting [>=====]
10.62MB/27.51MB
d9802f032d67: Downloading [>=====]
43a1e784ed3f: Downloading [>=====]
d9802f032d67: Extracting [>=====]
10.91MB/27.51MB
d9802f032d67: Extracting [>=====]
d9802f032d67: Full complete
claa0a434c07: Full complete
35627bdf29eb: Full complete
4c0eff2d67a3: Full complete
109dd45d95t: Full complete
43a1e784ed3f: Full complete
569674eda147: Full complete
c69235504c3e: Full complete
Digest: sha256:c4561867435f5b60d43ca2bd8304ace5881ca2705e9858cc53962e64b009af00
```

Con **docker ps** se comprueba que ha funcionado:

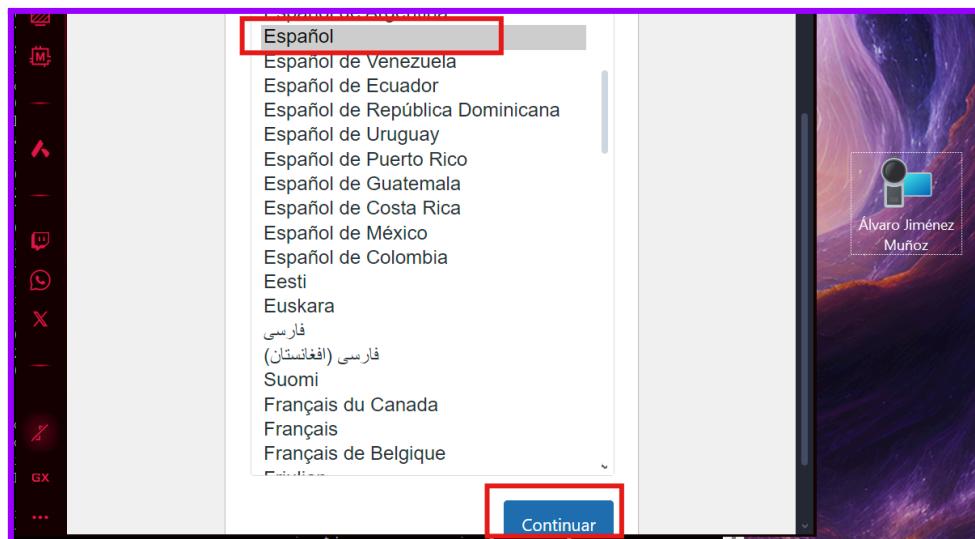


```
9df2a6231627: Pull complete
b3207e60ff9a: Pull complete
d18c9f420b35: Pull complete
673faad72ba8: Pull complete
4f4fb700ef54: Pull complete
b0dc28254b99: Pull complete
353e28a55fb0: Pull complete
acfd66114040: Pull complete
901e497f1af0: Pull complete
715f44938211: Full complete
3d6944f05f98: Full complete
3b30f0e2417e: Full complete
90f489708f36: Full complete
Digest: sha256:b60e01ce06202c836b46d54cb3eceba3a6f30950491e6805ce17d840ff6943c0
Status: Downloaded newer image for wordpress:latest
Creating mariadb ... done
Creating wordpress ... done
ubuntu@ip-172-31-21-57:~/wordpress$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
STATUS              PORTS              NAMES
303a340f0ff2        wordpress:latest   "docker-entrypoint.s..."   About a minute ago   Up    About a minute ago   0.0.0.0:9000->80/tcp, :::9000->80/tcp   wordpress
c4dbd8b82a78        mariadb:10.5      "docker-entrypoint.s..."   About a minute ago   Up    About a minute ago   3306/tcp                           mariadb
```

Ahora toca intentar despegar **wordpress** en el navegador, usando la IP de instancia EC2 seguida de :9000:



Al hacerlo, si ha ido bien saldrá eso en el navegador y lo que quedaría por hacer sería seguir los pasos que salen por pantalla para configurar wordpress:

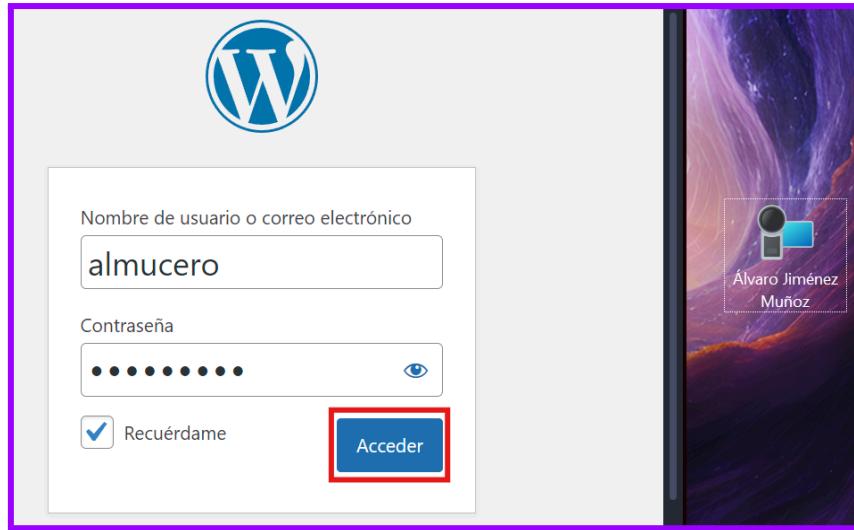


The screenshot shows the first step of a WordPress installation wizard. It includes fields for 'Título del sitio' (UbuntuDockerComposeAWS), 'Nombre de usuario' (almucero), and 'Contraseña' (a password consisting of six dots). A note below the user name field specifies allowed characters: alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @. A strength indicator bar at the bottom right shows the password as 'Muy débil'.

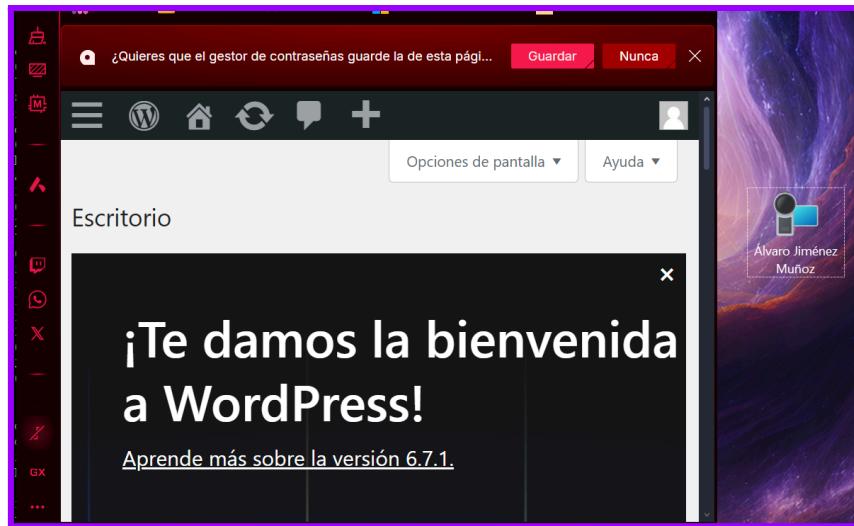
This screenshot shows the second step of the installation wizard. It includes a checkbox for confirming the use of a weak password, which is checked. Below it is a field for 'Tu correo electrónico' (ajim mun901@g.educaand.es) with a note to check the email address. Under 'Visibilidad en los motores de búsqueda', there is a checkbox for requesting search engines not to index the site, which is unchecked. A note states that search engines will handle this request. At the bottom is a prominent blue 'Instalar WordPress' button.

Por último se accede:

The final screenshot shows the completion of the WordPress installation. A message box displays '¡Lo lograste!' (You did it!). Below it, a note says 'WordPress ya está instalado. ¡Gracias, y que lo disfrutes!' (WordPress is installed. Thank you, and enjoy it!). It shows the previously entered user information ('Nombre de usuario': almucero, 'Contraseña': La contraseña que has elegido). At the bottom is a red-bordered 'Acceder' (Access) button.



Y así quedaría:



BIBLIOGRAFÍA

<https://docs.aws.amazon.com/ec2/index.html>

https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/CreatingEBSSnapshots.html

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html>

<https://chatgpt.com/>

<https://www.google.com/>

<https://www.youtube.com/>

[Apuntes](#)