



HERRAMIENTA EC2 DE AWS

Sistema Informáticos (1º DAM)



Amazon
EC2

Antonio Salces Alcaraz (1º DAM)
C.P.I.F.P. Alan Turing
18/01/2025

ÍNDICE

<u>1.</u>	<u>CREAR INSTANCIA EC2</u>	<u>1</u>
<u>2.</u>	<u>ACCEDER A LA NUEVA INSTANCIA CON PUTTY Y OPERAR CON ELLA</u>	<u>5</u>
2.1.	ACCEDER CON PUTTY	5
2.2.	ACTUALIZAR PAQUETES Y DESCARGAR DOCKER.IO	7
2.3.	PASAR WEB ESTÁTICA CON “WINSCP”	8
2.4.	ARRANCAR CONTENEDOR CON “NGINX” PARA PODER VER LA WEB EN INTERNET.	10
<u>3.</u>	<u>TRABAJAR CON LA INSTANCIA DESDE LA TERMINAL DE LINUX</u>	<u>12</u>
3.1.	CONVERTIR CLAVE .PPK A .PEM	12
3.2.	ACCEDER A LA INSTANCIA MEDIANTE SSH	13
3.3.	ENVIAR PÁGINA WEB MEDIANTE “SCP”	13
3.4.	DESCOMPIMIR Y ARRANCAR CONTENEDOR	13
<u>4.</u>	<u>CREAR AMI</u>	<u>16</u>
4.1.	COMO CREAR UNA AMI DE UNA INSTANCIA	16
<u>5.</u>	<u>LANZAR INSTANCIA CON LA NUEVA AMI Y TRABAJAR CON DOCKER-COMPOSE Y WORDPRESS.</u>	<u>18</u>
5.1.	LANZAR INSTANCIA CON LA AMI	18
5.2.	INSTALAR “DOCKER-COMPOSE” Y DESPLEGAR WORDPRESS EN LA NUEVA INSTANCIA	18

1. Crear instancia EC2

En este primer ejercicio, tenemos que crear una instancia en EC2 con las siguientes características:

- **Nombre:** UbuntuDockerAWS
- **Plataforma:** Ubuntu
- **Tipo de instancia:** t2.micro
- **Acceso por clave:** vockey.ppk / vockey.pem
- **Configuración del firewall:** permitir SSH (22), HTTP (80 y 8080) y HTTPS (443)
- **Almacenamiento:** 8 GB (por defecto)

En la pestaña “Nombre y etiquetas”, elegiremos el nombre.

Nombre y etiquetas Información

Nombre

[Agregar etiquetas adicionales](#)



En la pestaña “Imágenes de aplicaciones y sistemas operativos (Imagen de máquina de Amazon)”, elegiremos el Sistema Operativo.

▼ Imágenes de aplicaciones y sistemas operativos (Imagen de máquina de Amazon) Información

Una AMI es una plantilla que contiene la configuración de software (sistema operativo, servidor de aplicaciones y aplicaciones) necesaria para lanzar la instancia. Busque o examine las AMI si no ve lo que busca a continuación.

Recientes

Inicio rápido

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Wir

M

Buscar más AMI

Inclusión de AMI de AWS, Marketplace y la comunidad



En la pestaña “Tipo de instancia”, elegiremos el tipo de máquina que queremos dentro de las opciones de Amazon.

▼ **Tipo de instancia** [Información](#) | [Obtener asesoramiento](#)

Tipo de instancia

t2.micro Apto para la capa gratuita

Familia: t2 1 vCPU 1 GiB Memoria

Generación actual: true

Bajo demanda Windows base precios: 0.0162 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

☐ Todas las generaciones

[Comparar tipos de instancias](#)

Se aplican costos adicionales a las AMI con software preinstalado



En la pestaña “Par de claves (inicio de sesión)”, elegiremos la clave a utilizar.

▼ **Par de claves (inicio de sesión)** [Información](#)

Puede utilizar un par de claves para conectarse de forma segura a la instancia. Asegúrese de que tiene acceso al par de claves seleccionado antes de lanzar la instancia.

Nombre del par de claves - obligatorio

vockey.ppk

[Crear un nuevo par de claves](#)



En la pestaña “Configurar almacenamiento”, elegiremos el almacenamiento que queramos.

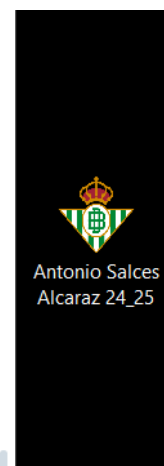
▼ **Configurar almacenamiento** [Información](#) [Avanzado](#)

1x 8 GiB gp3 Volumen raíz 3000 IOPS

(Sin cifrar)

[Agregar un nuevo volumen](#)

La AMI seleccionada contiene más volúmenes de almacén de instancias de los que permite la instancia. Solo se podrá obtener acceso desde la instancia a los primeros 0 volúmenes de almacén de instancias de la AMI



En la pestaña “Configuraciones de red”, podremos activar SSH, HTTP y HTTPS activando las tres casillas.

▼ Configuraciones de red Información

Editar

Red | Información

vpc-0f99debec65150384

Subred | Información

Sin preferencias (subred predeterminada en cualquier zona de disponibilidad)

Asignar automáticamente la IP pública | Información

Habilitar

Se aplican cargos adicionales cuando no se cumplen los límites del nivel gratuito

Firewall (grupos de seguridad) | Información

Un grupo de seguridad es un conjunto de reglas de firewall que controlan el tráfico de la instancia. Agregue reglas para permitir que un tráfico específico llegue a la instancia.

☒ Crear grupo de seguridad

☐ Seleccionar un grupo de seguridad existente

Crearemos un nuevo grupo de seguridad denominado "launch-wizard-5" con las siguientes reglas:

☒ Permitir el tráfico de SSH desde
Ayuda a establecer conexión con la instancia

Cualquier lugar

0.0.0.0/0

☒ Permitir el tráfico de HTTPS desde Internet
Para configurar un punto de enlace, por ejemplo, al crear un servidor web

☒ Permitir el tráfico de HTTP desde Internet

Antonio Salces Alcaraz 24_25

Papelera de reciclaje

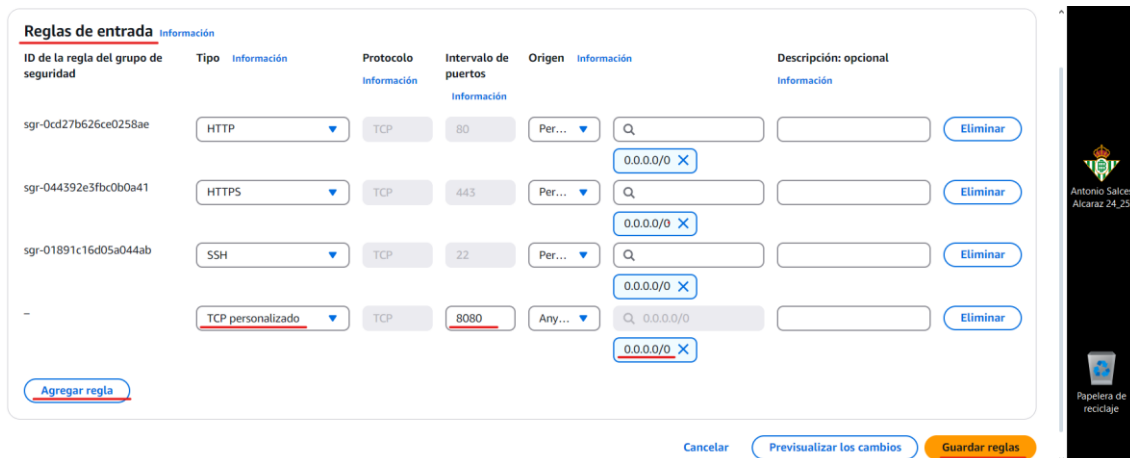
Para activar el puerto 8080 en la instancia, tenemos primero que saber el grupo de seguridad de la misma. Para ello, iremos a la pantalla donde se ven nuestras instancias, y buscaremos “Nombre del grupo de seguridad”.

<u>Nombre del grupo de seguridad</u> ▼	Nombre de... ▼	Hora de
launch-wizard-7	vockey.ppk	2025/0
default	vockey.ppk	2025/0

Ahora nos dirigiremos a “Red y seguridad > Security Groups”, seleccionamos el grupo de seguridad de la instancia, y pulsaremos sobre “Acciones > Editar reglas de entrada”.



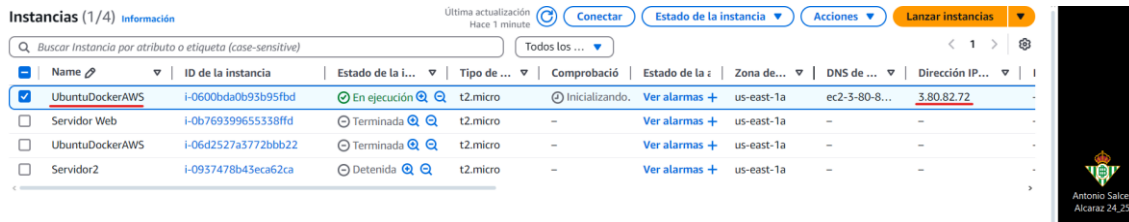
Tendremos que hacer click sobre “Agregar regla” y, en el menú desplegable, seleccionar TCP personalizado, pondremos entonces el puerto deseado, y pondremos las IPs válidas para conectarnos a ese puerto. Si ponemos “0.0.0.0/0”, podremos conectarnos desde cualquier IP. Cuando lo hayamos puesto todo, clickaremos sobre “Guardar regla”.



2. Acceder a la nueva instancia con PuTTY y operar con ella

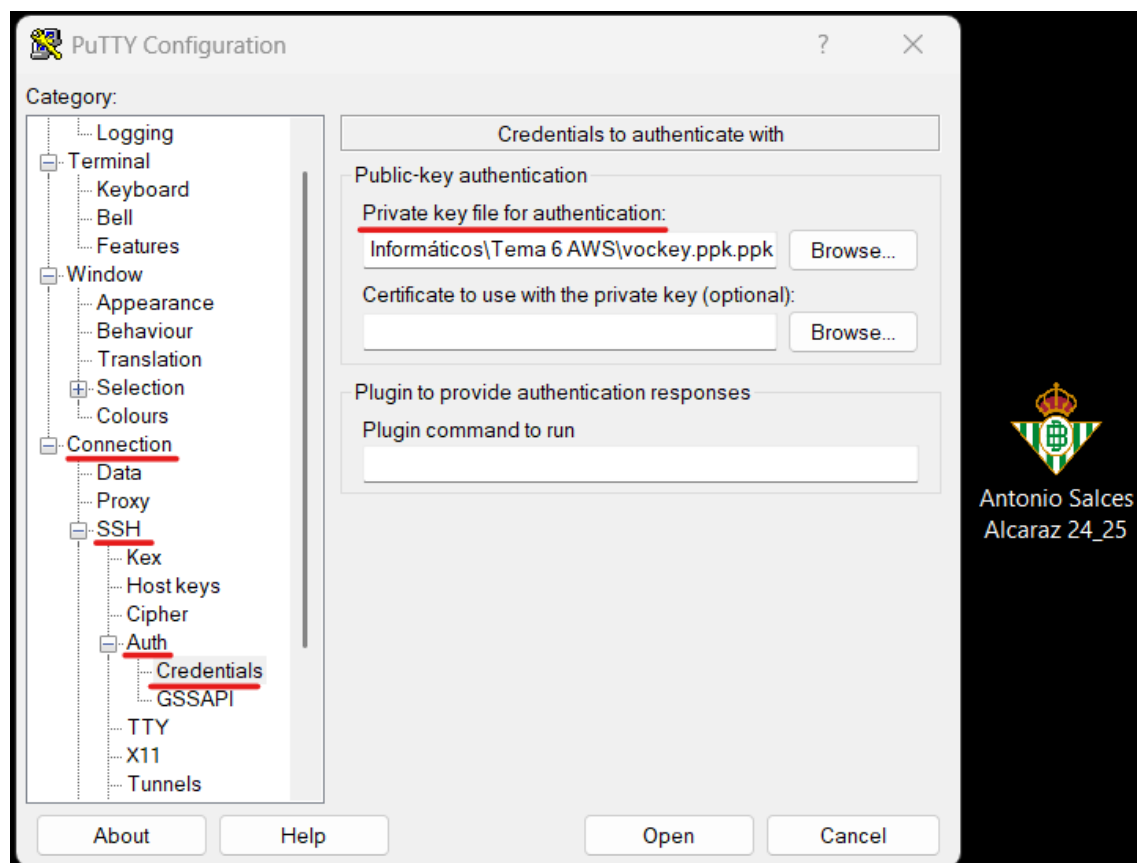
2.1. Acceder con PuTTY

En primer lugar, debemos de saber la IP que tiene nuestra instancia en el momento de conectarnos mediante “PuTTY”. Para ellos, vamos a nuestras instancias, y buscamos la IP de la instancia.

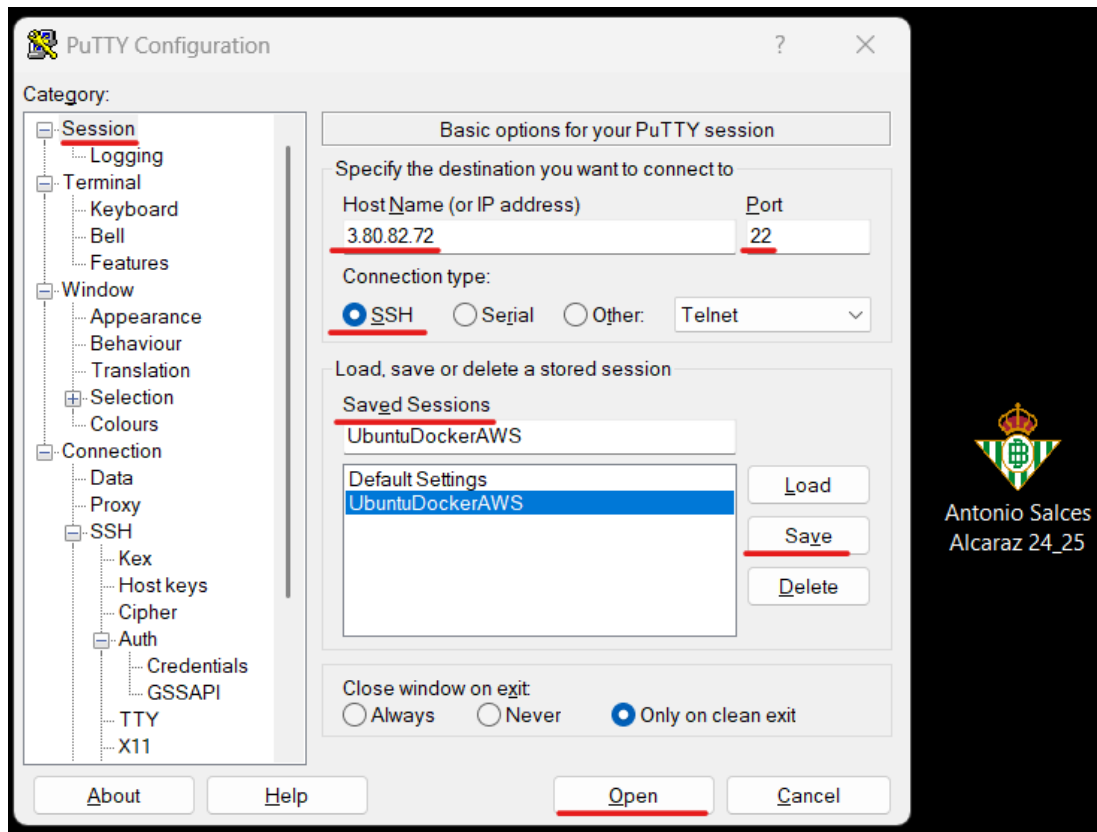


Name	ID de la instancia	Estado de la i...	Tipo de ...	Comprobació	Estado de la i	Zona de...	DNS de ...	Dirección IP...
UbuntuDockerAWS	i-0600bda0b93b95fbd	En ejecución	t2.micro	Inicializando	Ver alarmas +	us-east-1a	ec2-3-80-8...	3.80.82.72
Servidor Web	i-0b769399655338ff	Terminada	t2.micro	-	Ver alarmas +	us-east-1a	-	-
UbuntuDockerAWS	i-06d2527a3772bbb22	Terminada	t2.micro	-	Ver alarmas +	us-east-1a	-	-
Servidor2	i-0937478b43eca62ca	Detenida	t2.micro	-	Ver alarmas +	us-east-1a	-	-

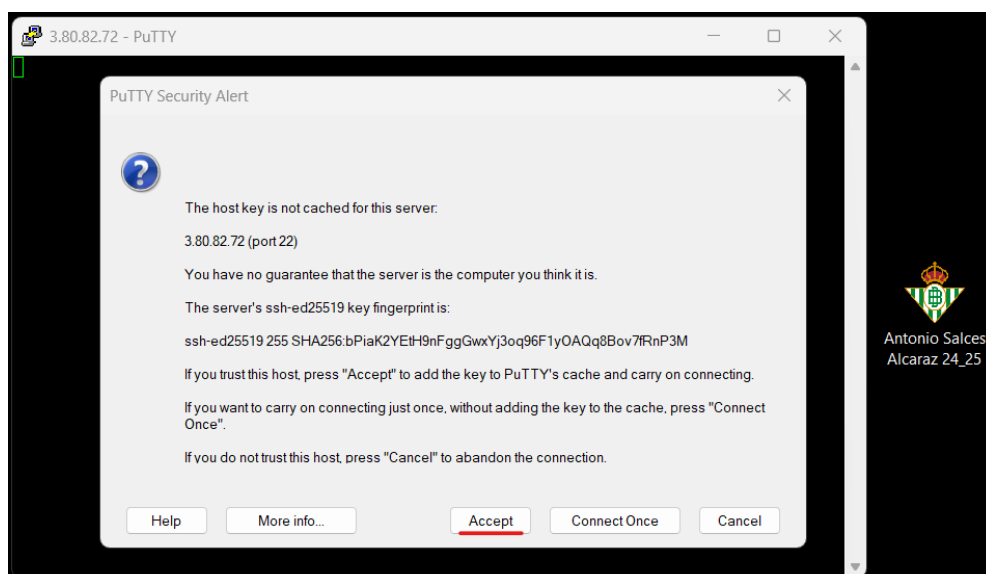
Ahora abrimos “PuTTY”. Iremos a la sección “Connection > SSH > Auth > Credentials”, y una vez dentro, pondremos la clave “vockey.ppk” dentro de “Private key file for authentication”.



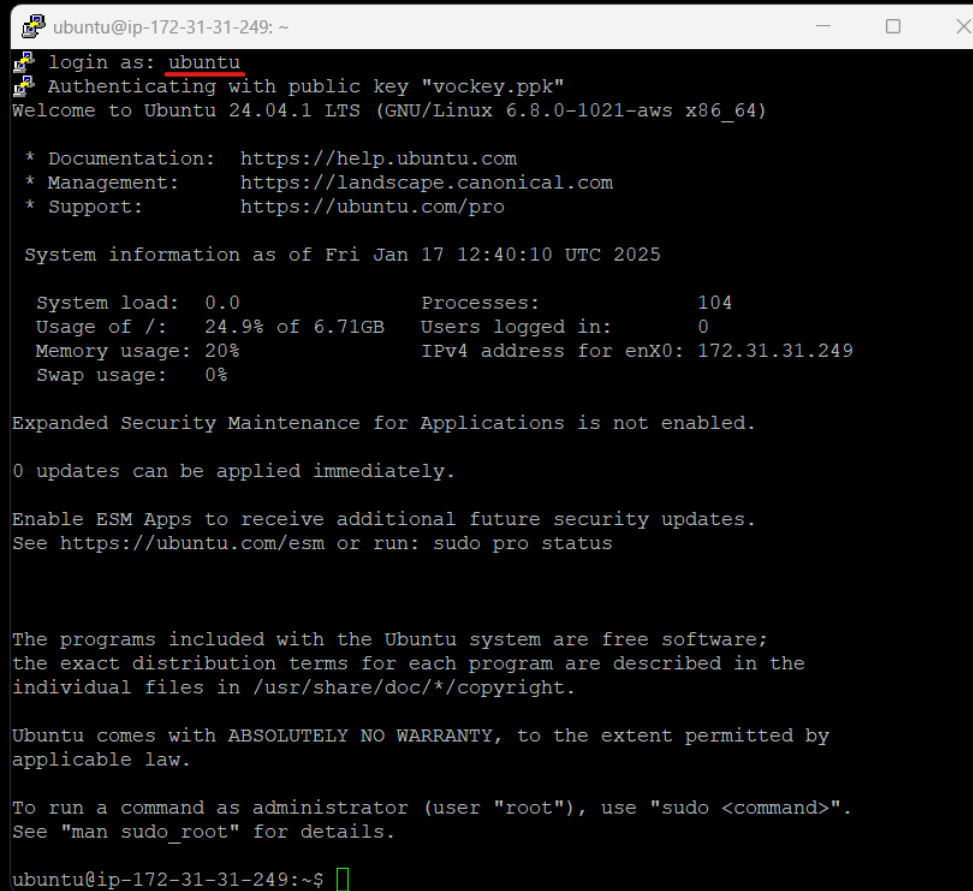
Ahora vamos a la pestaña “Session”. Primero pondremos la IP dentro de “Host Name (or IP address)”, y el puerto 22 dentro de “Port”. Seleccionaremos SSH dentro de “Connection type”. Para guardar la configuración para futuras conexiones, pondremos el nombre deseado de “Saved Sessions” y clickamos sobre “Save”. Una vez hayamos guardado la configuración, haremos click en “Open” para acceder a la instancia.



Cuando entremos por primera en cualquier instancia, nos saltará un aviso de seguridad. Clickando sobre “Accept”, podremos acceder a la instancia.



Una vez accedemos a la instancia, tendremos que loguearnos como "ubuntu", y PuTTY se encargará de enviar las claves para poder acceder. Como podemos ver en la imagen, ya podemos acceder y utilizar comando en la máquina.



```

ubuntu@ip-172-31-31-249: ~
login as: ubuntu
* Authenticating with public key "vockey.ppk"
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 17 12:40:10 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: 172.31.31.249
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 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-172-31-31-249:~$

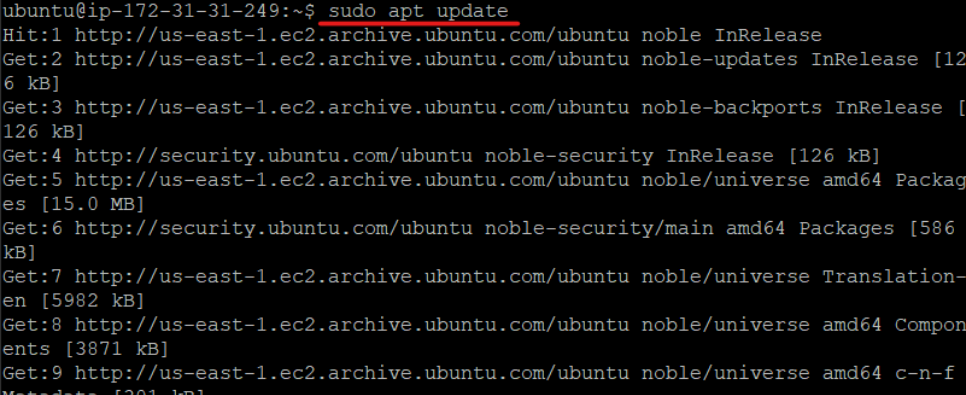
```



Antonio Salces
Alcaraz 24_25

2.2. Actualizar paquetes y descargar docker.io

Primero, debemos de actualizar los paquetes con "***sudo apt update***"



```

ubuntu@ip-172-31-31-249:~$ 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://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [586 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]

```



Antonio Salces
Alcaraz 24_25

Ahora utilizaremos “***sudo apt install <paquete>***” para instalar “docker.io”.

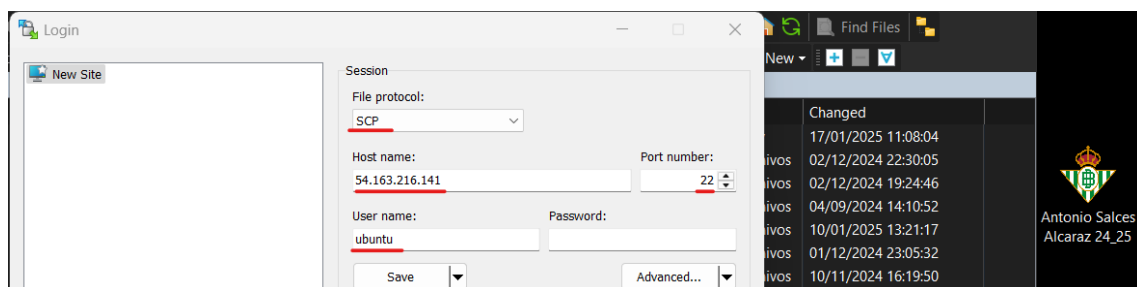
```
ubuntu@ip-172-31-31-249:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  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
  ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 1 not upgraded.
Need to get 80.1 MB of archives.
After this operation, 304 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz a
md64 2.8-1 [65.6 kB]
```



Antonio Salces
Alcaraz 24_25

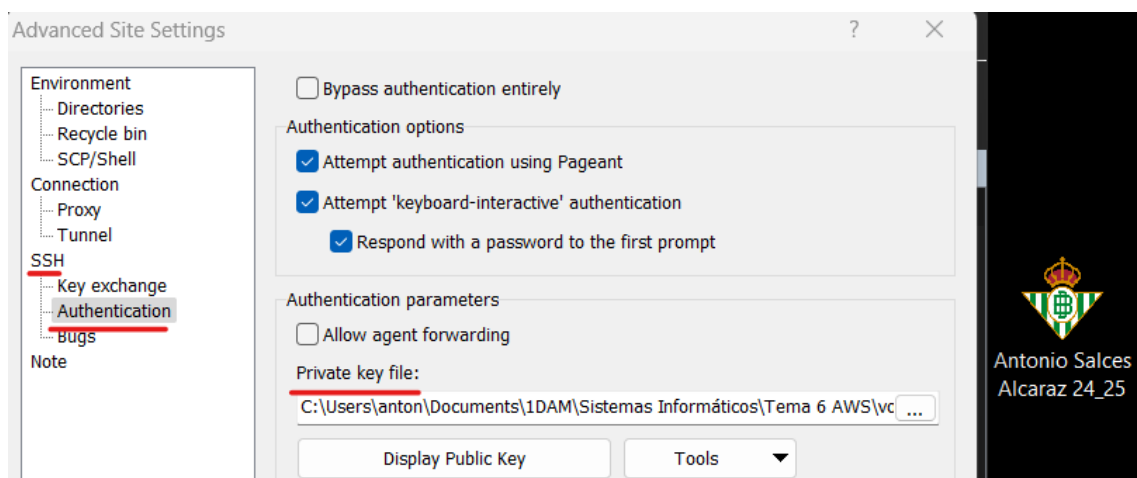
2.3. Pasar web estática con “WinSCP”

Abriremos “WinSCP”. En “File protocol” elegiremos “SCP”, en “Host name” pondremos la IP, y en “User name” pondremos “Ubuntu”. Ahora nos iremos a la sección “Advanced...”.



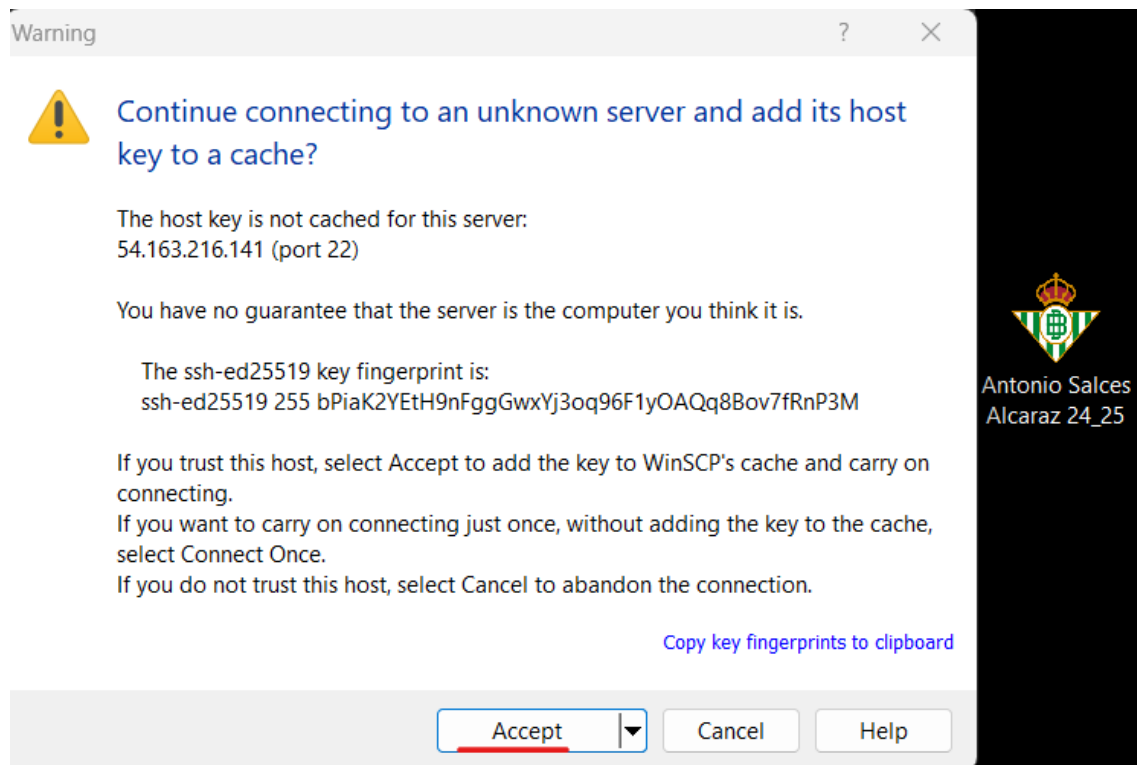
Antonio Salces
Alcaraz 24_25

Nos dirigiremos a “SSH > Authentication”, y en “Private key file:” pondremos la clave generada anteriormente. Tras colocarla, podremos conectarnos.

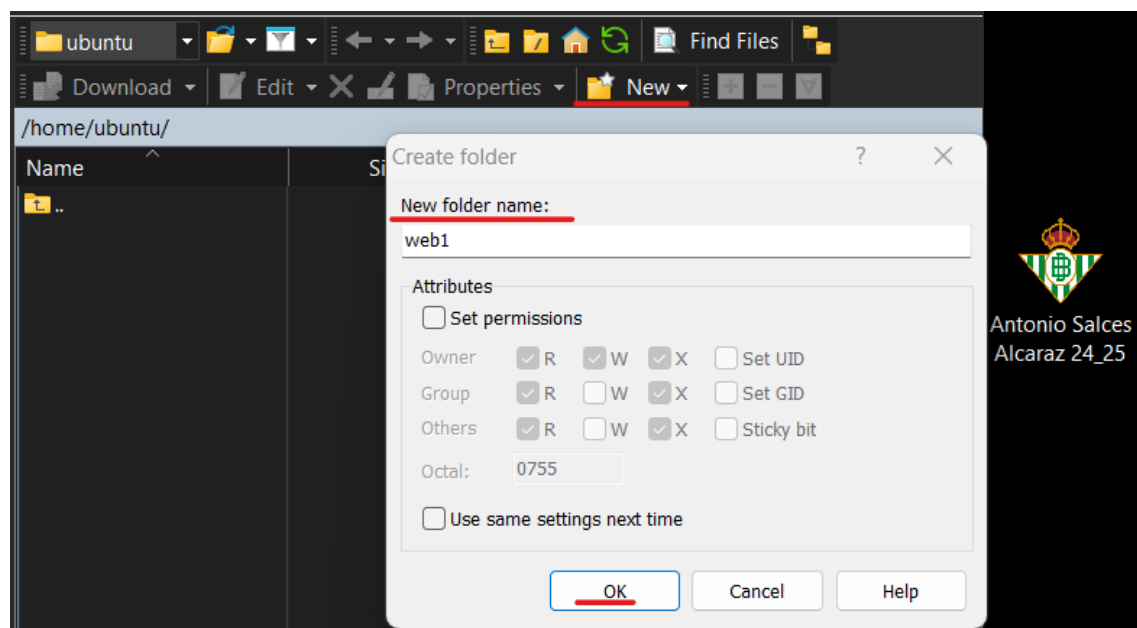


Antonio Salces
Alcaraz 24_25

Al igual que “PuTTY”, nos dará un warning de seguridad. Simplemente tenemos que darle a “Accept”.



Ahora clickaremos sobre el botón señalado para crear una nueva carpeta dentro de “/home/ubuntu”, y en “New folder name:” pondremos el nombre deseado, y daremos a “OK”.



Ahora nos dirigiremos a la carpeta de nuestro S.O. que tenga los archivos de la página web a publicar, entonces, los seleccionaremos y los pegaremos a `/home/ubuntu/web1` de nuestra instancia.

C:\1DAM\Lenguajes de marcas y sistemas de gestión de información\proyectoFinal\				/home/ubuntu/web1/				
Name	Size	Type	Changed	Name	Size	Changed	Rights	Owner
.		Parent directory	09/12/2024 20:35:22	.		17/01/2025 19:28:17	rw-r--r--	ubuntu
audio		Carpeta de archivos	06/12/2024 17:44:54	audio		17/01/2025 19:29:12	rw-r--r--	ubuntu
css		Carpeta de archivos	08/12/2024 13:15:50	css		17/01/2025 19:29:15	rw-r--r--	ubuntu
fonts		Carpeta de archivos	06/12/2024 18:46:59	fonts		17/01/2025 19:29:18	rw-r--r--	ubuntu
img		Carpeta de archivos	06/12/2024 21:14:48	img		17/01/2025 19:29:30	rw-r--r--	ubuntu
video		Carpeta de archivos	06/12/2024 18:21:29	video		17/01/2025 19:30:07	rw-r--r--	ubuntu
discografia.html	6 KB	Opera GX Web Do...	08/12/2024 13:57:23	discografia.html	6 KB	08/12/2024 12:57:23	rw-r--r--	ubuntu
ENLACES.txt	1 KB	Documento de tex...	09/12/2024 13:41:22	ENLACES.txt	1 KB	09/12/2024 12:41:22	rw-r--r--	ubuntu
formulario.html	5 KB	Opera GX Web Do...	08/12/2024 14:51:10	formulario.html	5 KB	08/12/2024 13:51:10	rw-r--r--	ubuntu
index.html	5 KB	Opera GX Web Do...	06/12/2024 21:40:56	index.html	5 KB	06/12/2024 20:40:56	rw-r--r--	ubuntu
integrantes.html	8 KB	Opera GX Web Do...	06/12/2024 21:43:05	integrantes.html	8 KB	06/12/2024 20:43:05	rw-r--r--	ubuntu
videosCoreos.html	4 KB	Opera GX Web Do...	06/12/2024 21:43:03	videosCoreos.html	4 KB	06/12/2024 20:43:03	rw-r--r--	ubuntu



Antonio Salces
Alcaraz 24_25

2.4. Arrancar contenedor con “nginx” para poder ver la web en Internet.

En primer lugar, haremos **`sudo docker pull <imagen>`** para descargar **`nginx:1.22.1`**. Una vez lo hayamos descargado, utilizaremos el comando **`sudo docker run [-d] [-p <puerto>] [-v <ruta de ubuntu:ruta de nginx>] <imagen>`** para arrancar el contenedor y enlazar la carpeta que contiene nuestra web, a la carpeta publica del contenedor. Por último, utilizaremos **`docker ps`** para comprobar que el contenedor está funcionando.

```
ubuntu@ip-172-31-31-249:~$ sudo docker pull nginx:1.22.1
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:fc5f5fb7574755c306aaf88456ebf0b006420a184d52b923d2f0197108f6b7
Status: Downloaded newer image for nginx:1.22.1
docker.io/library/nginx:1.22.1
ubuntu@ip-172-31-31-249:~$ docker run -d -p 80:80 -v /home/ubuntu/web1:/usr/share/nginx/html nginx:1.22.1
docker: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping": dial unix /var/run/docker.sock: connect: permission denied.
See 'docker run --help'.
ubuntu@ip-172-31-31-249:~$ sudo docker run -d -p 80:80 -v /home/ubuntu/web1:/usr/share/nginx/html nginx:1.22.1
2a4b5ffdb40fc6ce3b671be81c3f99cc23d2fc8ec24b372a6d76d5908d6a495
ubuntu@ip-172-31-31-249:~$ docker ps
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/containers/json": dial unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-31-249:~$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
2a4b5ffdb40   nginx:1.22.1   "/docker-entrypoint..." 15 seconds ago Up 14 seconds
0.0.0.0:80->80/tcp, :::80->80/tcp   goofy_dewdney
```



Antonio Salces
Alcaraz 24_25

Ahora, si buscamos la IP de nuestra instancia en cualquier buscador, podremos ver la página web que hemos subido a Ubuntu.



3. Trabajar con la instancia desde la terminal de Linux

3.1. Convertir clave .ppk a .pem

En primer lugar, debemos de convertir el archivo “.ppk” en “.pem” para conectarnos en la terminal de Ubuntu. Para ello, actualizaremos el sistema con “**apt update**” como primer paso.

```

usuario@usuario:~$ sudo apt update
Des:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Obj:2 http://es.archive.ubuntu.com/ubuntu noble InRelease
Des:3 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Des:4 https://download.docker.com/linux/ubuntu noble InRelease [48,8 kB]
Des:5 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Des:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [586 kB]
Des:7 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [775 kB]
Des:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [114 kB]
Des:9 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [18,2 kB]
Des:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [7.240 B]
Des:11 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
]

```

Tras actualizar los paquetes y las dependencias, instalaremos el paquete “putty-tools” con “**apt install <paquete>**”.

```

usuario@usuario:~$ sudo apt install putty-tools
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  bridge-utils python3-netifaces ubuntu-fan
Utilice «sudo apt autoremove» para eliminarlos.
Paquetes sugeridos:
  putty-doc
Se instalarán los siguientes paquetes NUEVOS:
  putty-tools
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 13 no actualizados.
Se necesita descargar 673 kB de archivos.
Se utilizarán 3.745 kB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu noble/universe amd64 putty-tools amd64 0.81-1 [673 kB]
Descargados 673 kB en 2s (341 kB/s)
Seleccionando el paquete putty-tools previamente no seleccionado.
(Leyendo la base de datos ... 151030 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) ...
usuario@usuario:~$

```

Tras instalar el paquete mencionado, con el comando “**puttygen <archivo de origen> -O private-openssh -o <archivo de destino>**” podremos convertir la antigua clave .ppk en una nueva .pem. Además, le tendremos que dar pocos permisos porque Ubuntu tiene unas directivas de seguridad estrictas con SSH. Por lo tanto, tendremos que ejecutar el comando “**chmod 600 <archivo>**” para que no de problemas con los permisos.

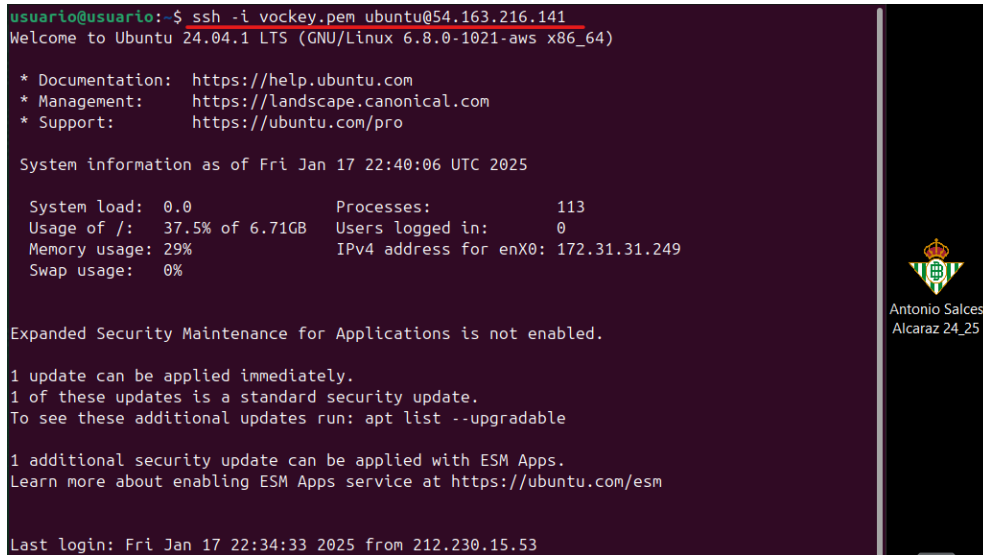
```

usuario@usuario:~$ puttygen vockey.ppk.ppk -O private-openssh -o vockey.pem
usuario@usuario:~$ chmod 600 vockey.pem
usuario@usuario:~$

```


3.2. Acceder a la instancia mediante SSH

Para acceder a la instancia, debemos utilizar el comando **"ssh -i clave <usuario>@<IP>"**.



```

usuario@usuario:~$ ssh -i vockey.pem ubuntu@54.163.216.141
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 17 22:40:06 UTC 2025

System load:  0.0               Processes:    113
Usage of /:   37.5% of 6.71GB   Users logged in: 0
Memory usage: 29%              IPv4 address for enX0: 172.31.31.249
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

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

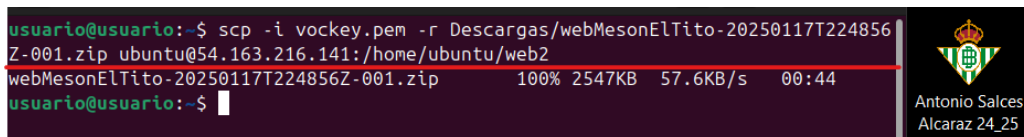
1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Fri Jan 17 22:34:33 2025 from 212.230.15.53

```

3.3. Enviar página web mediante "SCP"

Para enviar la web mediante SCP, debemos utilizar **"scp -i clave <-r ruta local> <usuario@IP>:<ruta>"**. Como podemos ver en la imagen, envío un .zip a la máquina, por lo que luego tendremos que descomprimirlo.



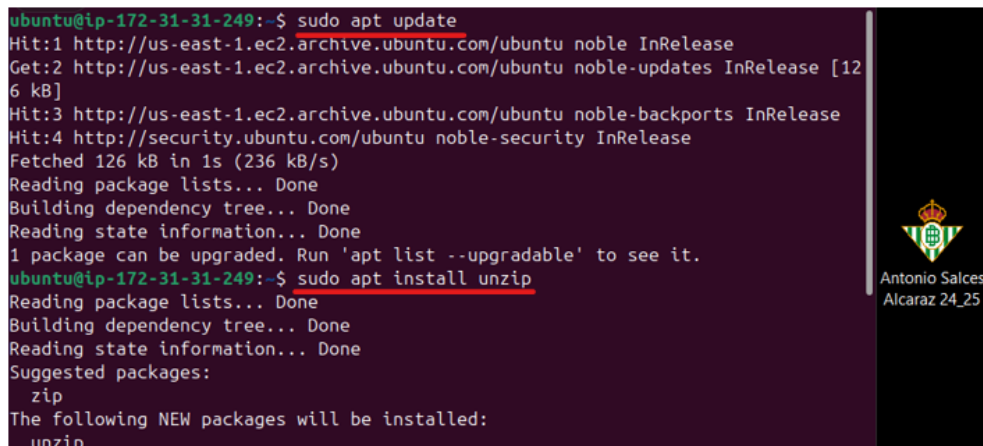
```

usuario@usuario:~$ scp -i vockey.pem -r Descargas/webMesonElTito-20250117T224856Z-001.zip ubuntu@54.163.216.141:/home/ubuntu/web2
webMesonElTito-20250117T224856Z-001.zip      100% 2547KB  57.6KB/s   00:44
usuario@usuario:~$

```

3.4. Descomprimir y arrancar contenedor

Primero tenemos que instalar el paquete "unzip", por lo que ejecutaremos **"apt update"** y, tras ello, **"apt install <paquete>"**.




```

ubuntu@ip-172-31-31-249:~$ 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]
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Fetched 126 kB in 1s (236 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-172-31-31-249:~$ sudo apt install unzip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  zip
The following NEW packages will be installed:
  unzip

```

Tras instalar el paquete “unzip”, ejecutaremos “**unzip <archivo>**” para descomprimirlo.


```
ubuntu@ip-172-31-31-249:~$ unzip web2
Archive:  web2
  inflating: webMesonElTito/tasty/contact.html
  inflating: webMesonElTito/tasty/index.html
  inflating: webMesonElTito/tasty/css/style.css
  inflating: webMesonElTito/tasty/css/bootstrap.css.map
  inflating: webMesonElTito/tasty/css/bootstrap-datetimepicker.min.css
  inflating: webMesonElTito/___MACOSX/tasty/._index.html
  inflating: webMesonElTito/tasty/css/flexslider.css
  inflating: webMesonElTito/___MACOSX/._tasty
  inflating: webMesonElTito/___MACOSX/tasty/._js
  inflating: webMesonElTito/___MACOSX/tasty/._css
  inflating: webMesonElTito/___MACOSX/tasty/._images
  inflating: webMesonElTito/___MACOSX/tasty/._fonts
  inflating: webMesonElTito/___MACOSX/tasty/._README.txt
  inflating: webMesonElTito/___MACOSX/tasty/._sass
  inflating: webMesonElTito/tasty/about.html
  inflating: webMesonElTito/tasty/css/bootstrap.css
  inflating: webMesonElTito/tasty/gallery.html
```



Antonio Salces
Alcaraz 24_25

Ahora haremos un “**ls**” por los archivos del archivo descomprimido para comprobar que está todo.


```
ubuntu@ip-172-31-31-249:~$ ls webMesonElTito/tasty/
README.txt  contact.html  fonts      images      js          reservation.html
about.html  css          gallery.html  index.html  menu.html  sass
```



Antonio Salces
Alcaraz 24_25

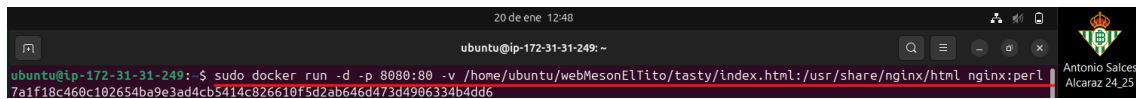
Necesitamos hacer un contenedor con “nginx:perl”, por lo que ejecutaremos el comando “**docker pull <paquete>**”.

```
ubuntu@ip-172-31-31-249:~$ sudo docker pull nginx:perl
perl: Pulling from library/nginx
af302e5c37e9: Pull complete
207b812743af: Pull complete
841e383b441e: Pull complete
0256c04a8d84: Pull complete
38e992d287c5: Pull complete
9e9aab598f58: Pull complete
4de87b37f4ad: Pull complete
0ddd5c97248c: Pull complete
Digest: sha256:393746e7c88de149e1292709c53eb8209358e1d46dde548882e1226e41430bb9
Status: Downloaded newer image for nginx:perl
docker.io/library/nginx:perl
ubuntu@ip-172-31-31-249:~$
```

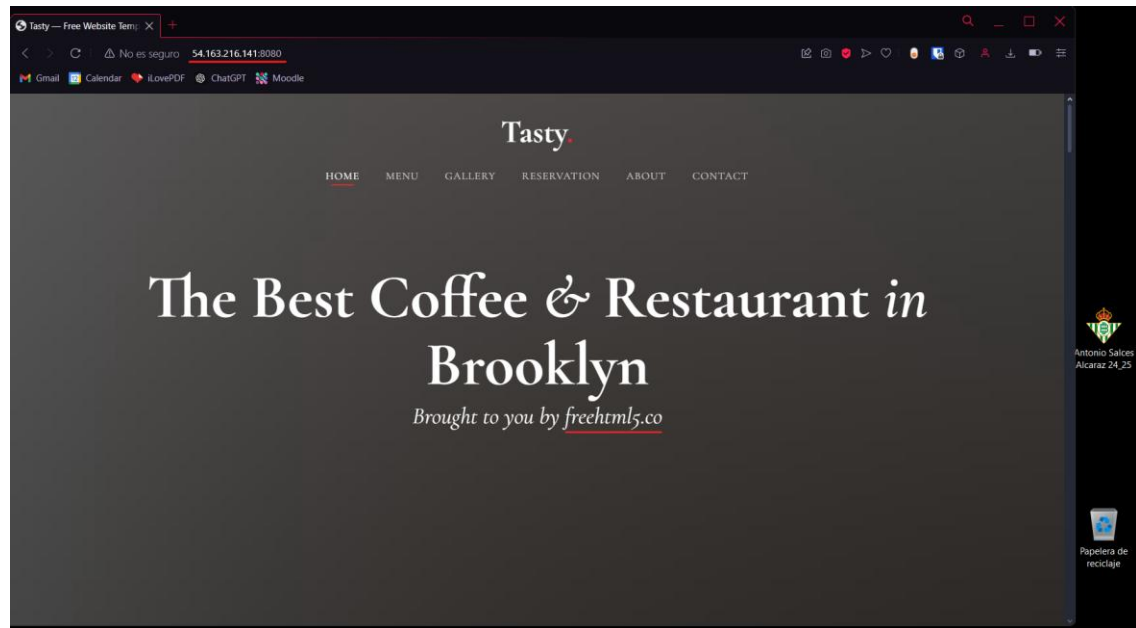


Antonio Salces
Alcaraz 24_25

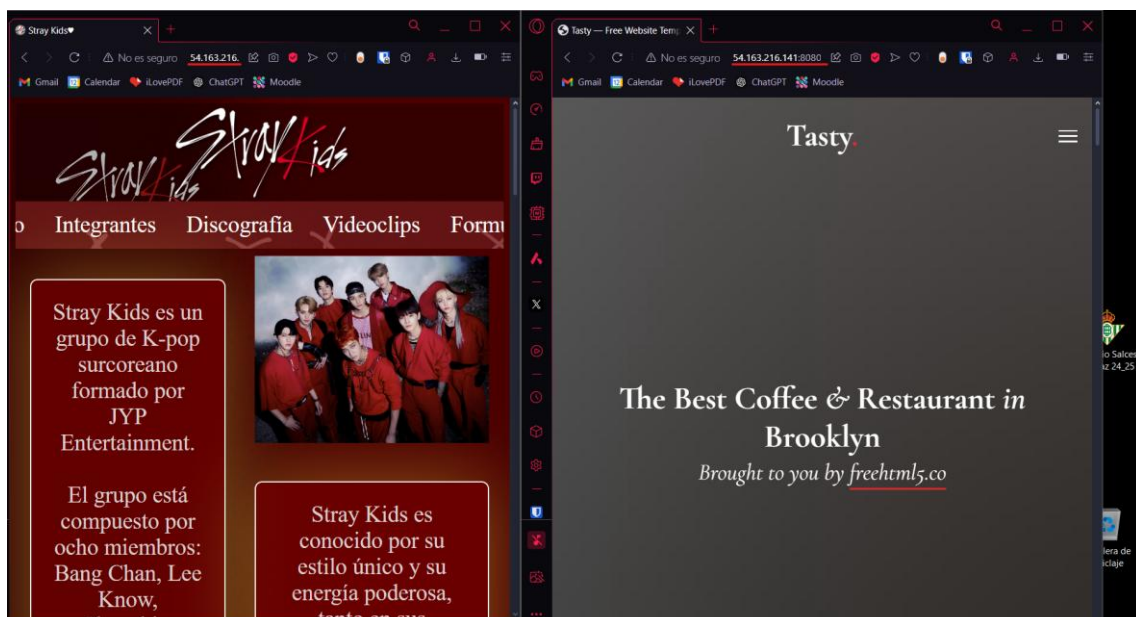
Ahora solo tendremos que arrancar el contenedor con **“*docker run [-d] [-p <puerto> [-v <ruta local>:<ruta al contenedor>] imagen*”**.



Tras arrancar el contenedor, si accedemos mediante el navegador a la IP por el puerto “8080”, podremos ver la nueva web.



Y aquí podremos ver como en la misma instancia, podemos tener dos páginas web en dos puertos diferentes.

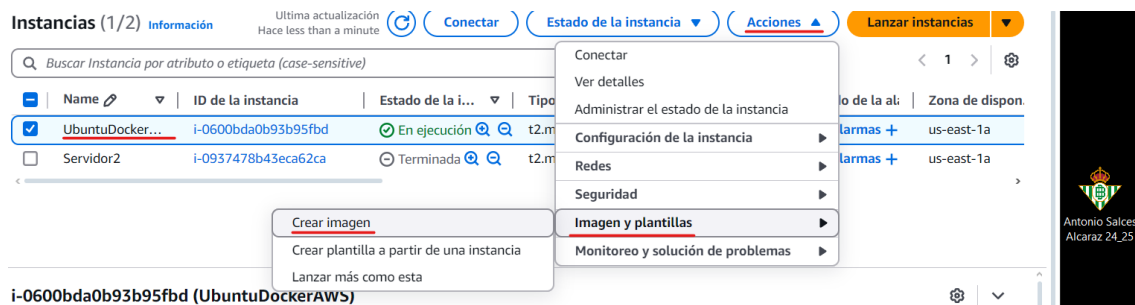


4. Crear AMI

Una AMI es una plantilla preconfigurada que proporciona la información necesaria para lanzar instancias EC2 en AWS. Está incluye información sobre el S.O., software preinstalado, y permisos.

4.1. Como crear una AMI de una instancia

En primer lugar, debemos de ir a la pantalla donde tenemos todas nuestras instancias. Marcaremos la instancia de la que queremos hacer la AMI, clickaremos sobre “Acciones”, nos iremos a “Imagen y plantillas”, y seleccionaremos “Crear imagen”.



Ahora pondremos el nombre a la imagen y una descripción si así lo quisiéramos. Marcaremos la opción “Reiniciar instancia” para que no haya problemas con los datos mientras se está creando la AMI.

Crear imagen Información

Una imagen (también denominada AMI) define los programas y la configuración que se aplican al lanzar una instancia EC2. Puede crear una imagen a partir de la configuración de una instancia existente.

ID de la instancia
 i-0600bda0b93b95fbd (UbuntuDockerAWS)

Nombre de la imagen

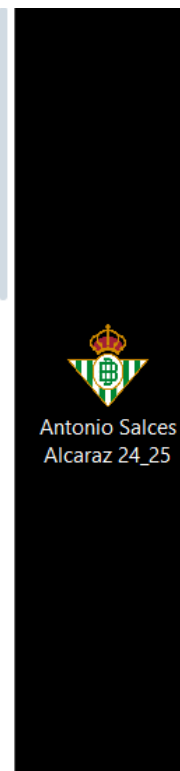
Máximo de 127 caracteres. No se pueden modificar después de su creación.

Descripción de la imagen: *opcional*

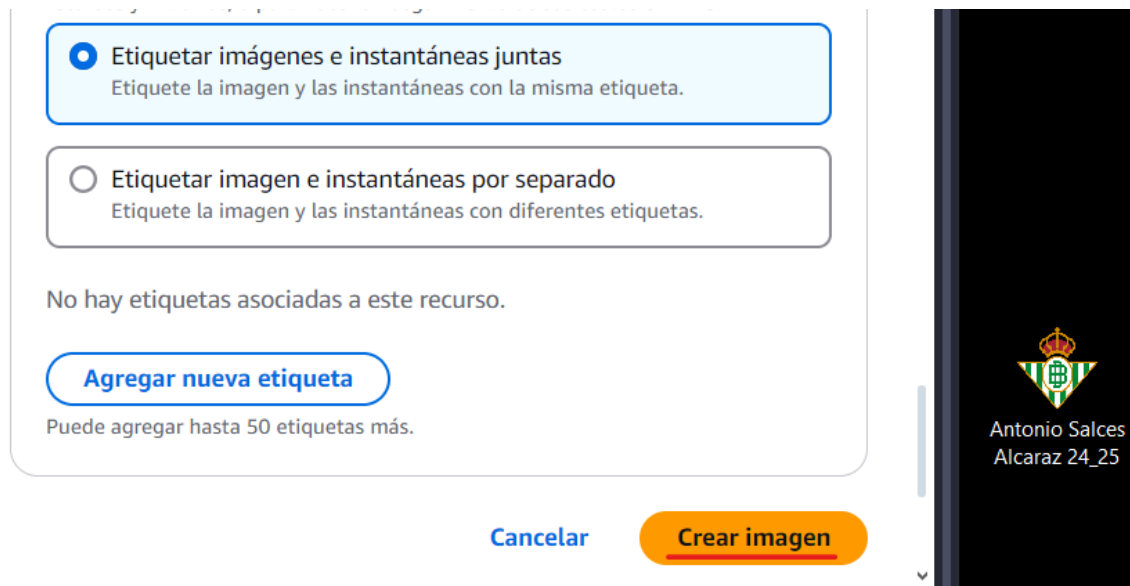
255 caracteres como máximo

☒ **Reiniciar instancia**

Cuando se selecciona esta opción, Amazon EC2 reinicia la instancia para mantener los datos en reposo cuando se toman instantáneas de los volúmenes asociados. Esto garantiza la coherencia de datos.



Ahora bajaremos al final de la pantalla y clickaremos sobre “Crear imagen”.



☒ Etiquetar imágenes e instantáneas juntas
Etiquete la imagen y las instantáneas con la misma etiqueta.

☐ Etiquetar imagen e instantáneas por separado
Etiquete la imagen y las instantáneas con diferentes etiquetas.

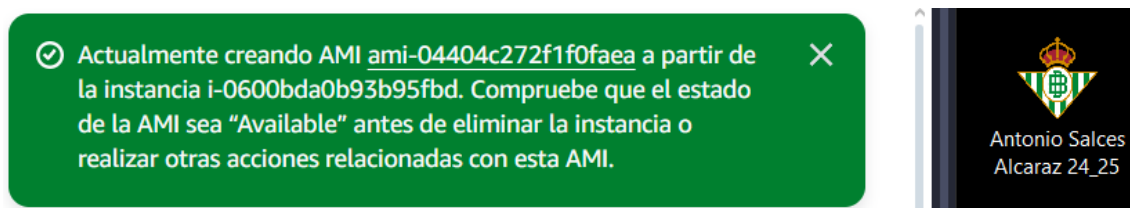
No hay etiquetas asociadas a este recurso.

Agregar nueva etiqueta
Puede agregar hasta 50 etiquetas más.

[Cancelar](#) [Crear imagen](#)

Antonio Salces Alcaraz 24_25

Y como podemos ver, nos sale un mensaje de que la AMI se está creando.



✓ Actualmente creando AMI [ami-04404c272f1f0faea](#) a partir de la instancia [i-0600bda0b93b95fbd](#). Compruebe que el estado de la AMI sea “Available” antes de eliminar la instancia o realizar otras acciones relacionadas con esta AMI. ✕

Antonio Salces Alcaraz 24_25

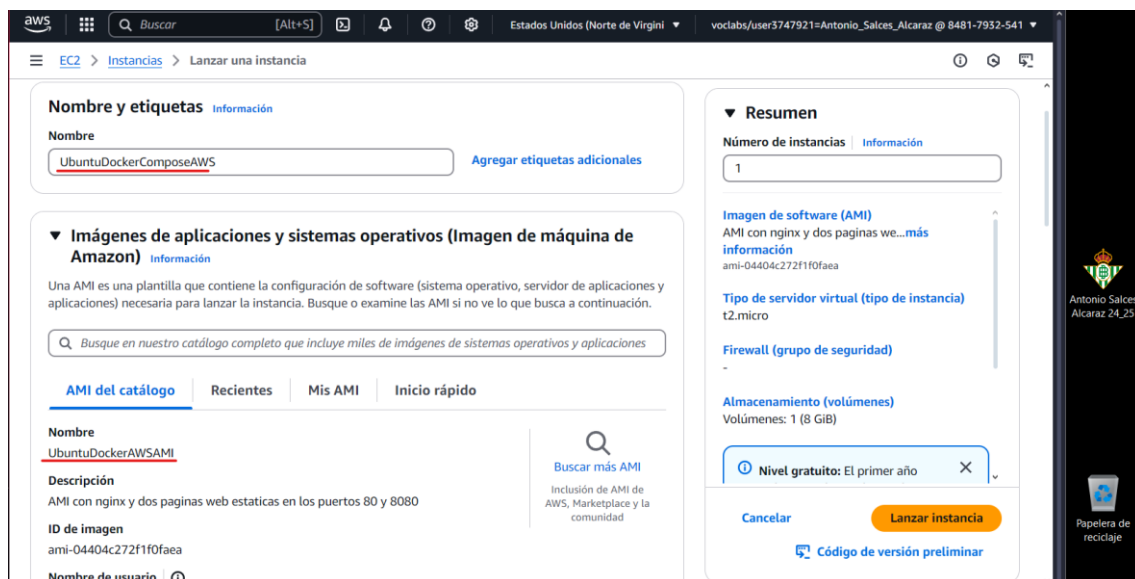
5. Lanzar instancia con la nueva AMI y trabajar con docker-compose y Wordpress.

5.1. Lanzar instancia con la AMI

En primer lugar, nos iremos a “Imágenes > AMI” dentro de la barra lateral izquierda. Buscaremos la AMI que acabamos de crear y clickaremos sobre “Lanzar instancia a partir de una AMI”.



Ahora pondremos el nombre y verificaremos que estamos utilizando la AMI creada anteriormente. Configuraremos la máquina y revisaremos toda la configuración al igual que en la primera actividad. Cuando este todo listo, haremos click en “Lanzar instancia”.



5.2. Instalar “docker-compose” y desplegar wordpress en la nueva instancia

En primer lugar, tenemos que actualizar los paquetes del sistema con “**apt install**”.



Ahora instalamos “docker compose v2” utilizando “**apt install <paquete>**”.

```
ubuntu@ip-172-31-26-244:~$ sudo apt install docker-compose-v2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Para desplegar el archivo .yml crearemos una carpeta con “**mkdir <nombre>**”. También crearemos y editaremos el archivo con “**nano <nombre>**”. En este paso se puede ver como hemos entrado desde “PuTTY”, en las siguientes imágenes se explicará el por qué.

```
ubuntu@ip-172-31-26-244: ~/wordpress
ubuntu@ip-172-31-26-244:~$ mkdir wordpress
ubuntu@ip-172-31-26-244:~$ cd wordpress/
ubuntu@ip-172-31-26-244:~/wordpress$ nano wordpress.yml
```

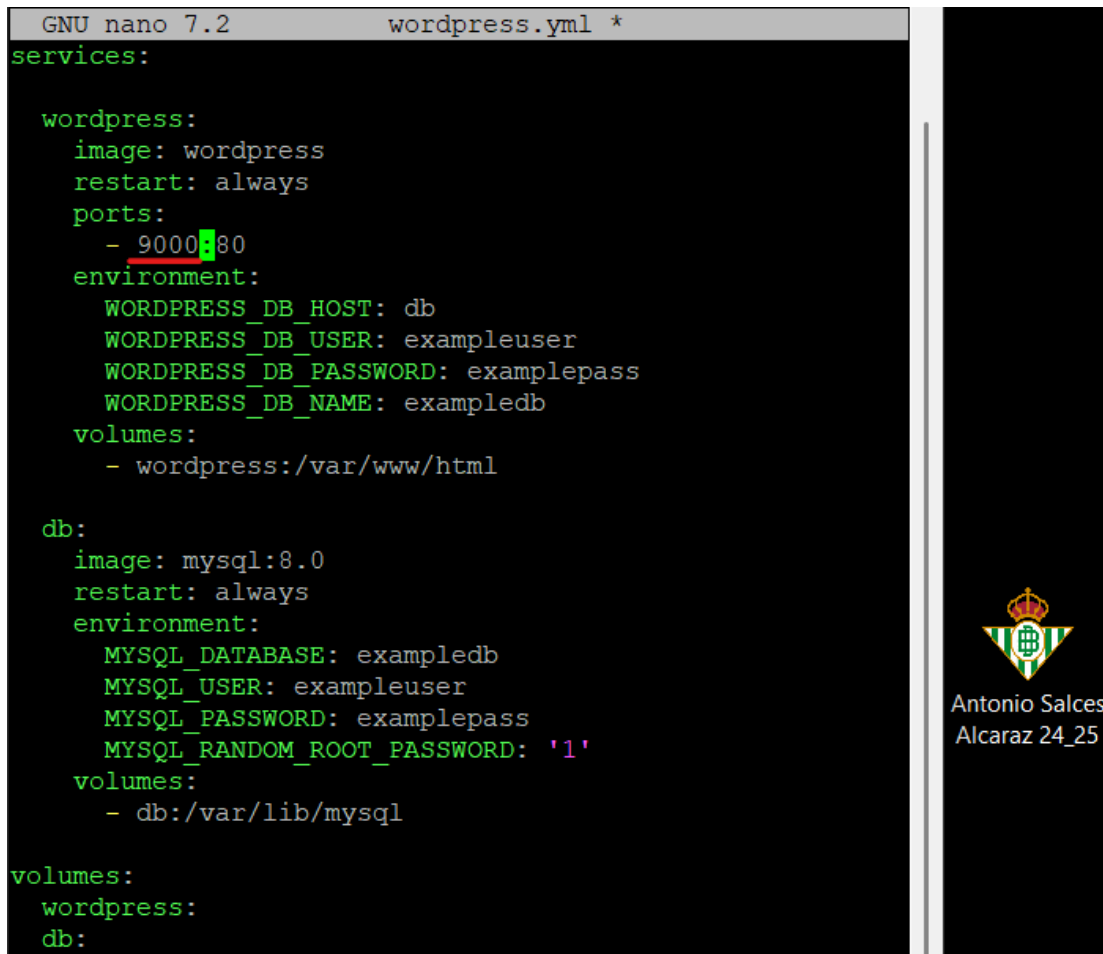
Podemos encontrar en la entrada oficial de “[Wordpress](#)” en la página “[Docker Hub](#)” un ejemplo de archivo .yml. Lo copiaremos en el portapapeles.

```
services:
  wordpress:
    image: wordpress
    restart: always
    ports:
      - 8080:80
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: exampleuser
      WORDPRESS_DB_PASSWORD: examplepass
      WORDPRESS_DB_NAME: exampledb
    volumes:
      - wordpress:/var/www/html

  db:
    image: mysql:8.0
    restart: always
    environment:
      MYSQL_DATABASE: exampledb
      MYSQL_USER: exampleuser
      MYSQL_PASSWORD: examplepass
      MYSQL_RANDOM_ROOT_PASSWORD: '1'
    volumes:
      - db:/var/lib/mysql

volumes:
  wordpress:
  db:
```

Desde “PuTTY”, ahora podremos pegar más cómodamente, por eso hemos accedido al sistema con él. Copiaremos todo el contenido en el archivo .yaml creado, y le cambiaremos el puerto al indicado en el enunciado (9000).



```

GNU nano 7.2      wordpress.yml *
services:

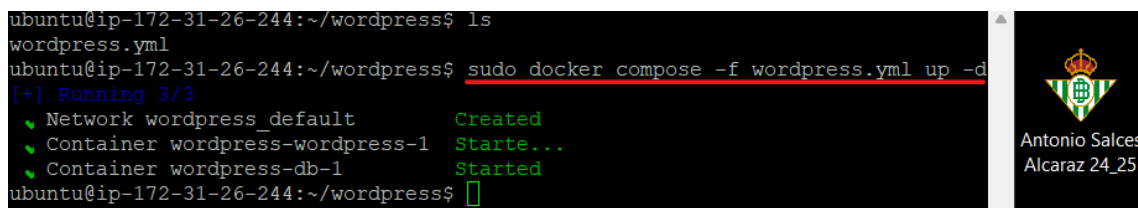
  wordpress:
    image: wordpress
    restart: always
    ports:
      - 9000:80
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: exampleuser
      WORDPRESS_DB_PASSWORD: examplepass
      WORDPRESS_DB_NAME: exampledb
    volumes:
      - wordpress:/var/www/html

  db:
    image: mysql:8.0
    restart: always
    environment:
      MYSQL_DATABASE: exampledb
      MYSQL_USER: exampleuser
      MYSQL_PASSWORD: examplepass
      MYSQL_RANDOM_ROOT_PASSWORD: '1'
    volumes:
      - db:/var/lib/mysql

volumes:
  wordpress:
  db:

```

Ahora solo debemos de poner en marcha el archivo con “***docker compose [-f <archivo>] up [-d]***”.

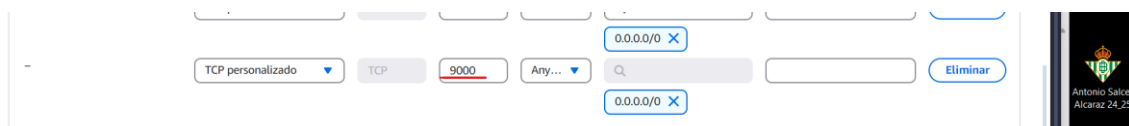


```

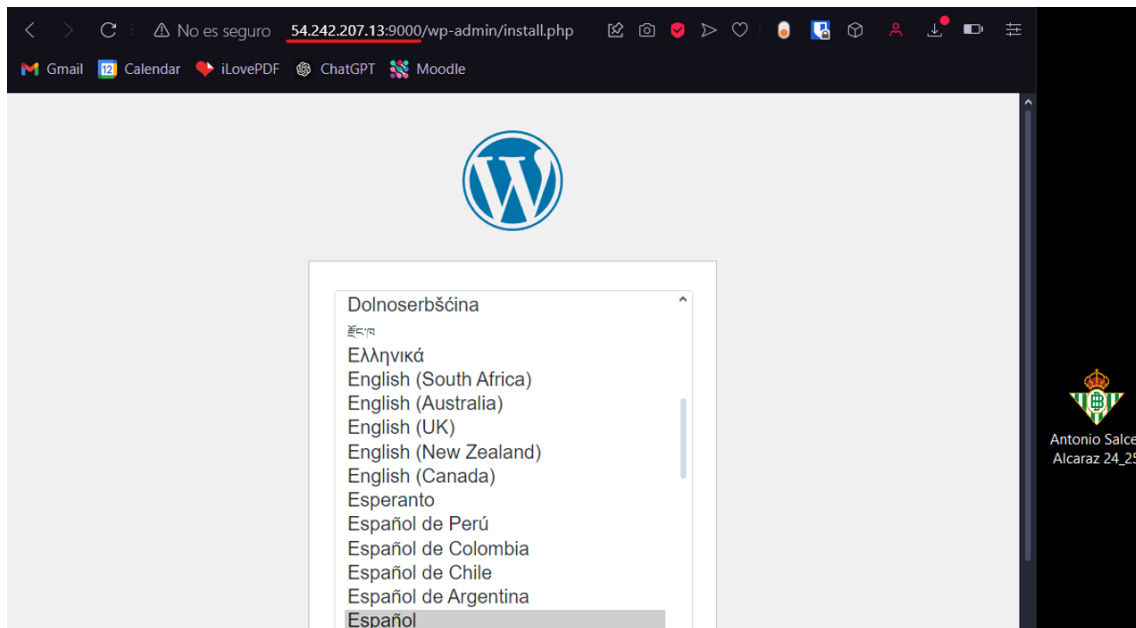
ubuntu@ip-172-31-26-244:~/wordpress$ ls
wordpress.yml
ubuntu@ip-172-31-26-244:~/wordpress$ sudo docker compose -f wordpress.yml up -d
[+] Running 3/3
  ✓ Network wordpress_default      Created
  ✓ Container wordpress-wordpress-1 Starte...
  ✓ Container wordpress-db-1      Started
ubuntu@ip-172-31-26-244:~/wordpress$

```

Para poder conectarnos a la instancia por Internet al puerto de “Wordpress”, tenemos que abrir el puerto asignado como se ha visto en el punto 1.



Como podemos ver en la siguiente imagen, si buscamos en el buscado la IP seguida del puerto (<IP>:<puerto>), podremos ver como entramos en el proceso de instalación de “Wordpress”.



Una vez lo hayamos instalado, podemos empezar a crear páginas que podremos ver en Internet con el enlace proporcionado por “Wordpress”.

