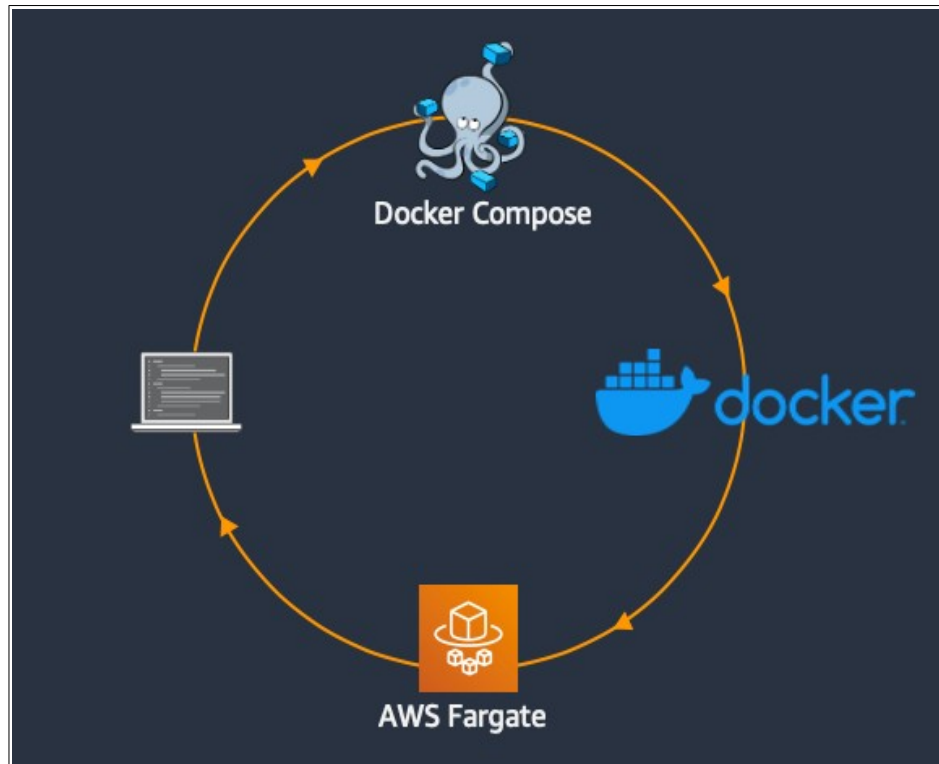


Despliegue de docker en AWS



Index

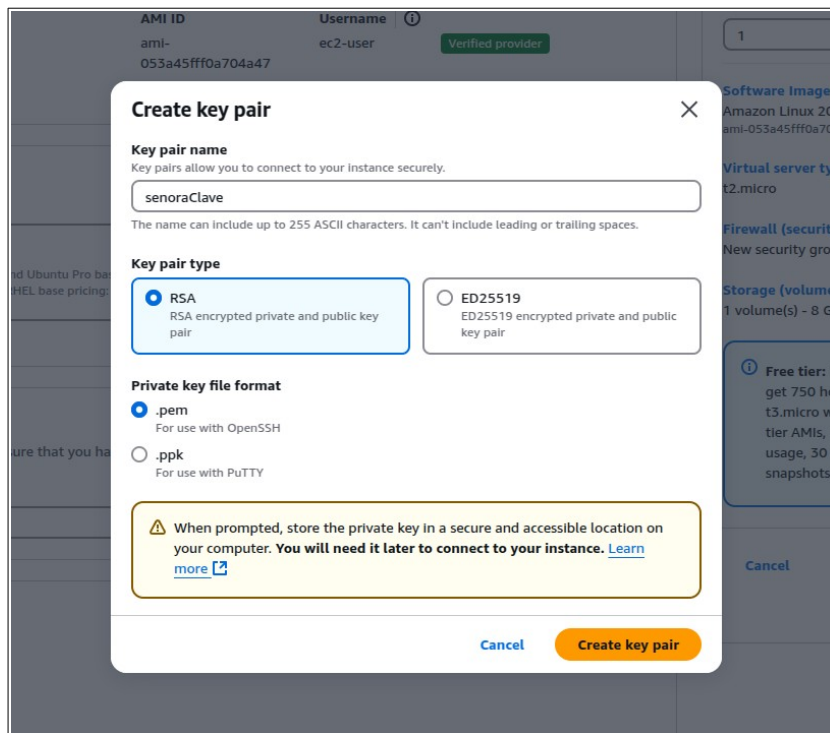
Creación de la instancia.....3

Instalación de docker.....4

Creación del proyecto.....5

Creación de la instancia

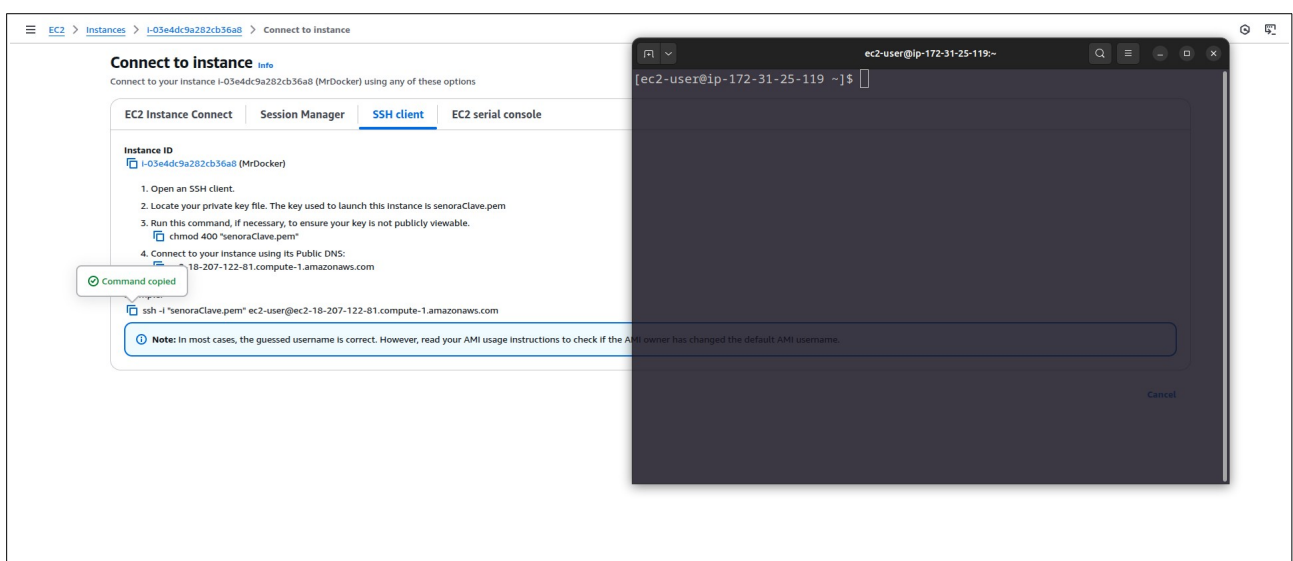
Una vez abierto el laboratorio, vamos a EC2 y crearemos una instancia, y le crearemos una clave para conectarnos a la instancia.



Y también cosica que no puse, habiliten el HTTP y el HTTPS también.

Una vez creada la instancia entramos en esta.

Seguimos los pasos de darle permisos al archivo de las clavecitas que se nos habrá descargado al crearlas y nos conectamos a este:



Instalación de docker

Con el comando:

```
sudo yum install docker
```

Nos pillamos docker para treballar con contenedores de docker.

```
Verifying      : docker-25.0.6-1.amzn2023.0.2.x86_64      2/10
Verifying      : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64 3/10
Verifying      : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64 4/10
Verifying      : libcgrou-3.0-1.amzn2023.0.1.x86_64      5/10
Verifying      : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64 6/10
Verifying      : libnftnl-1.0.1-19.amzn2023.0.2.x86_64   7/10
Verifying      : libnftnl-1.2.2-2.amzn2023.0.2.x86_64    8/10
Verifying      : pigz-2.5-1.amzn2023.0.3.x86_64          9/10
Verifying      : runc-1.2.4-1.amzn2023.0.1.x86_64        10/10

Installed:
 containerd-1.7.25-1.amzn2023.0.1.x86_64
 docker-25.0.6-1.amzn2023.0.2.x86_64
 iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
 iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
 libcgrou-3.0-1.amzn2023.0.1.x86_64
 libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
 libnftnl-1.0.1-19.amzn2023.0.2.x86_64
 libnftnl-1.2.2-2.amzn2023.0.2.x86_64
 pigz-2.5-1.amzn2023.0.3.x86_64
 runc-1.2.4-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-31-25-119 ~]$
```

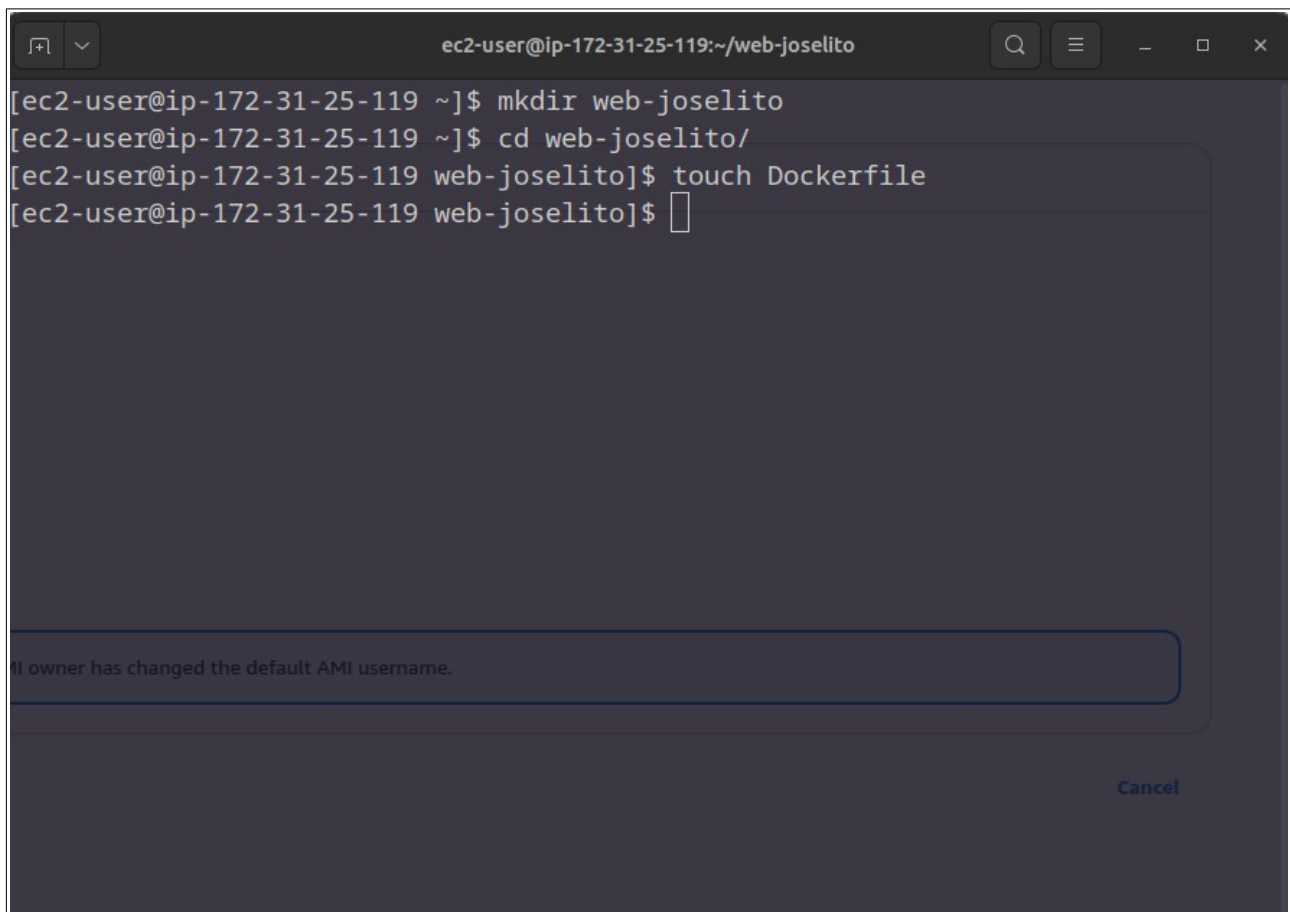
Y ahorita con el comando:

```
sudo usermod -aG docker ec2-user && newgrp docker
```

Le damos al usuario de la instancia permisos de docker para poder usarlo.

Creación del proyecto

Creamos un directorio para ejecutar la imagen de docker que queramos y dentro le preparamos el Dockerfile.

A terminal window with a dark background. The title bar shows 'ec2-user@ip-172-31-25-119:~/web-joselito'. The terminal shows the following commands and output:

```
[ec2-user@ip-172-31-25-119 ~]$ mkdir web-joselito
[ec2-user@ip-172-31-25-119 ~]$ cd web-joselito/
[ec2-user@ip-172-31-25-119 web-joselito]$ touch Dockerfile
[ec2-user@ip-172-31-25-119 web-joselito]$
```

Below the terminal output, there is a system message in a light blue box: 'The owner has changed the default AMI username.' and a 'Cancel' button in the bottom right corner.

Ahora lo editamos y le metemos lo que queramos ejecutar.

En mi caso yo le daré la siguiente configuración.

Será la de una imagen de nginx.

FROM nginx:latest

COPY index.html /usr/share/nginx/html/

EXPOSE 80

El index.html le meteré el siguiente contenido, pero eso ya va a cuenta de cada uno que poner:

```
<!DOCTYPE html>
<html>
<head>
  <title>Bienvenido a la página del gordo de babun</title>
</head>
<body>
  <h1>Hola Maricarmen</h1>
  <p>Ayudame, babu me tiene secuestrado en su sótano TwT</p>
```

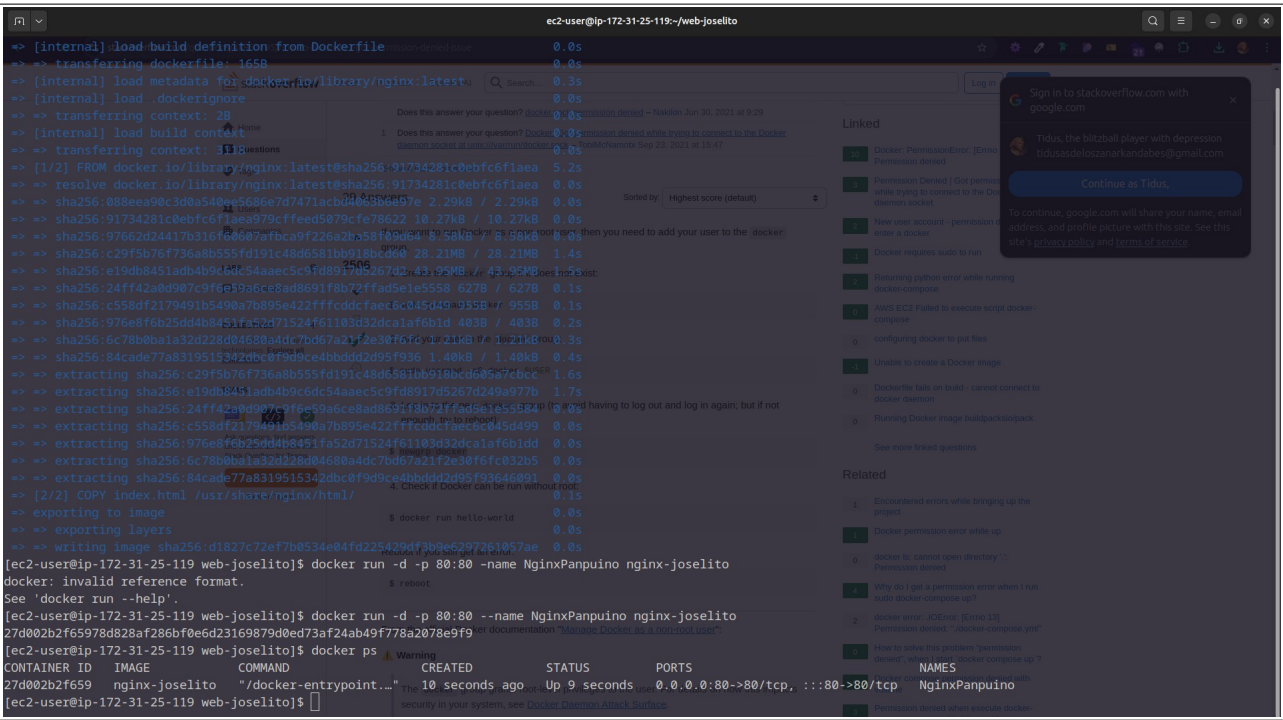
</body>
</html>

Ahorita para acabar hacemos un docker build para montar el contenedor de nginx:

docker build -t nombreDeLaImagen .

Y una vez creada la imagen la ejecutamos a través del puerto 80:

docker run -d -p 80:80 --name nombreDelContenedor nombreDeLaImagen



Y una vez accedamos a la instancia que hemos hecho por el navegador debería salirnos:

