Ejercicio 2. Unidad 6

Creación de un contenedor y registro en ECR

requisitos

- Cuenta de AWS
- Conocimiento de Docker Dokerfile

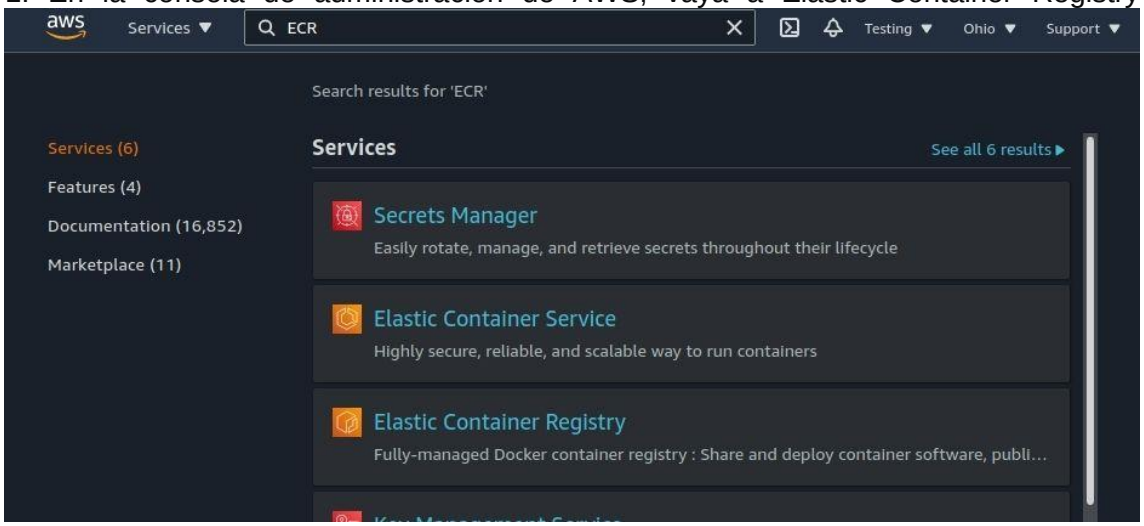
pasos

- Crear un repositorio
- Ejecutar una instancia con el rol de IAM (RoleLab)
- Crear un archivo Docker
- Crear una imagen desde el Dockerfile
- Subir la imagen al Registro

NOTA: Este Lab se supone que es lanzado en la región de N.Virginia. (us-east-1). En caso de que AWS Academy cambie la región hay que cambiar la región en los códigos de los ejemplos.

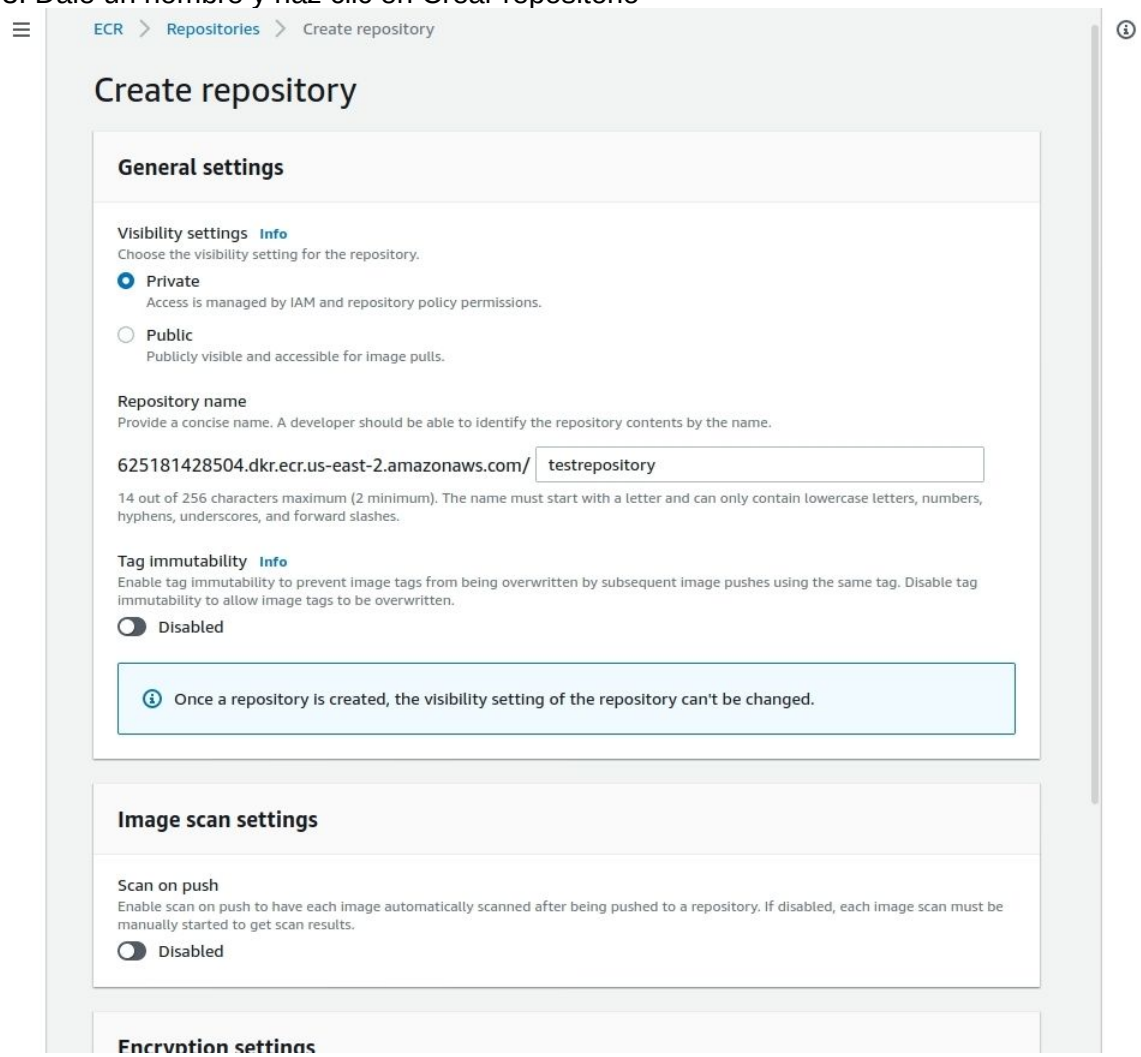
Por otro lado en donde ponga XXXXXXXX corresponde al numero de cuenta de AWS Academy que te lanza en ese momento que se puede consultar en el la consola de AWS en la parte superiro derecha.

1. En la consola de administración de AWS, vaya a Elastic Container Registry

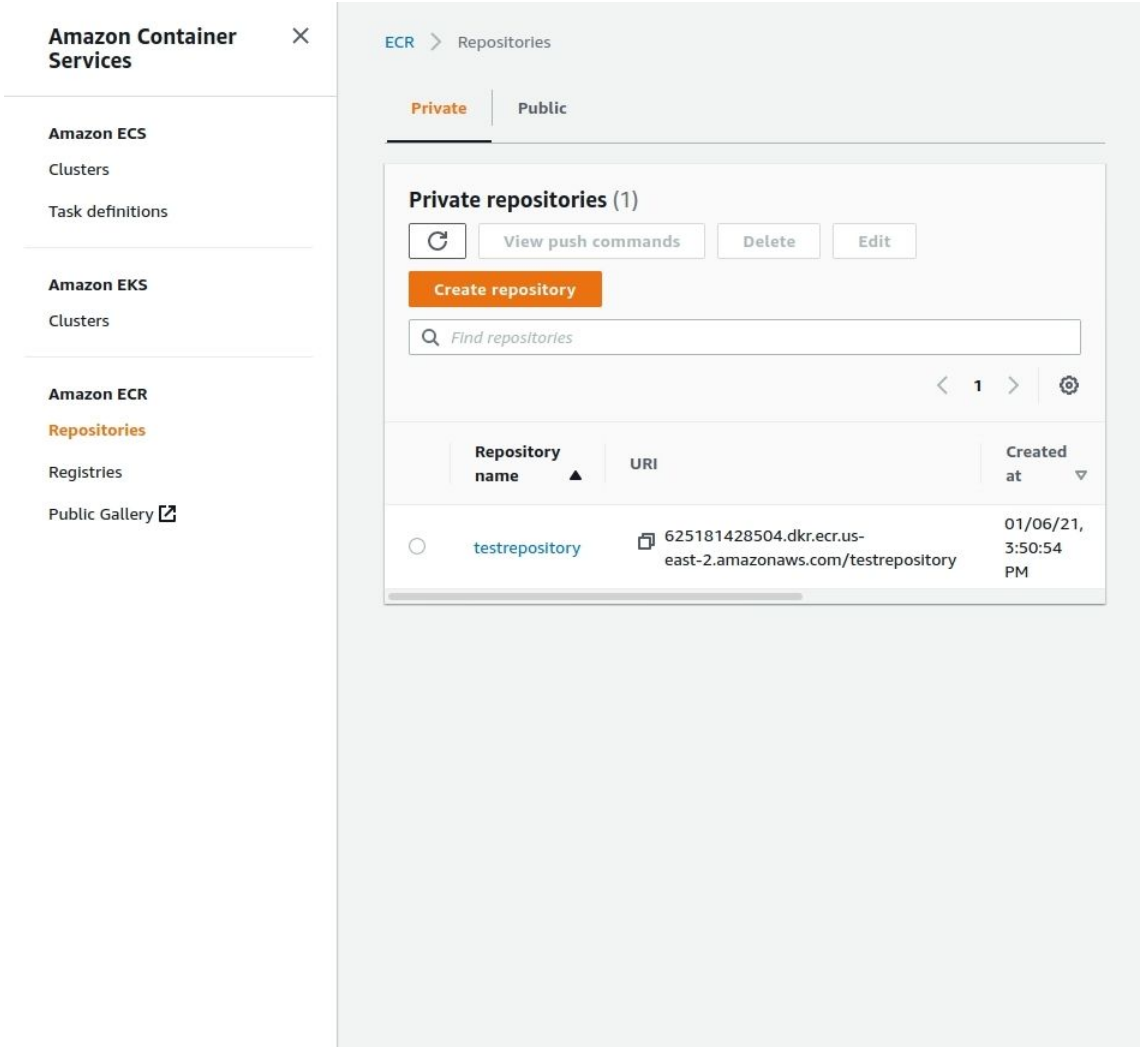


2. Pulsa en Get Started

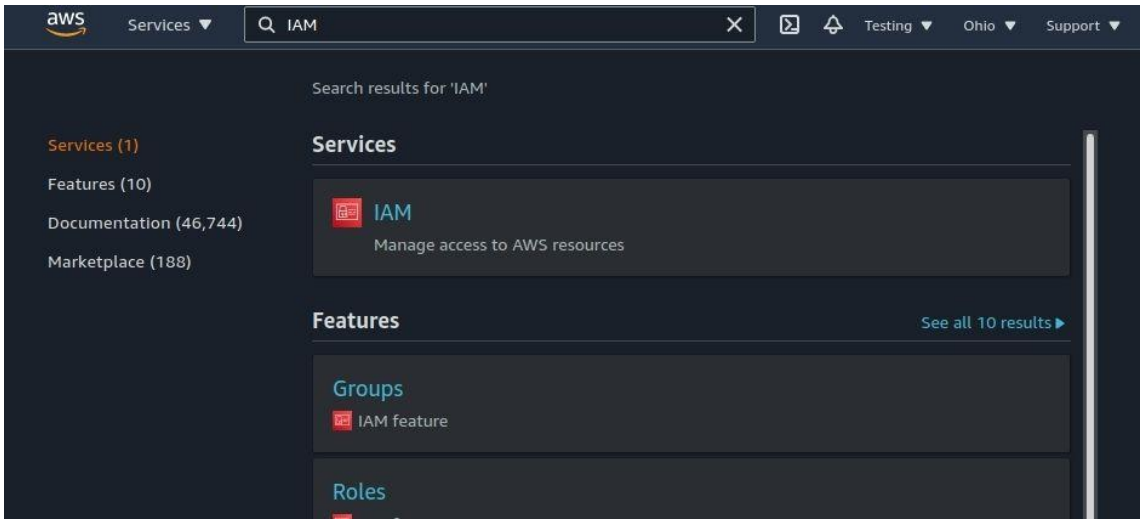
3. Dale un nombre y haz clic en Crear repositorio



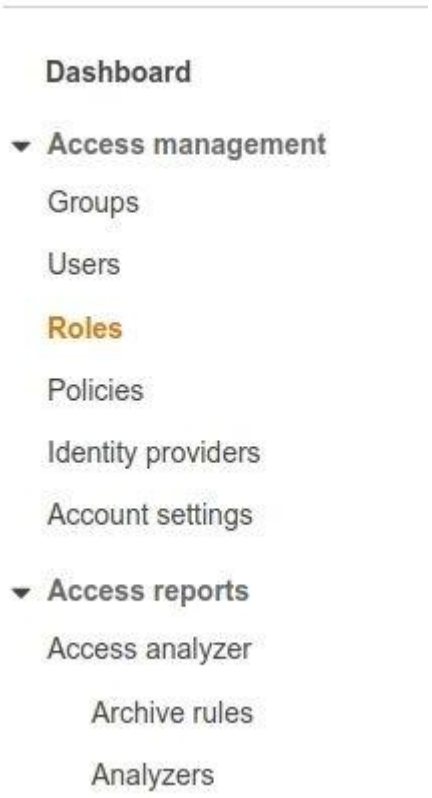
4. Una vez que termine, copie la URI para usarla más tarde.



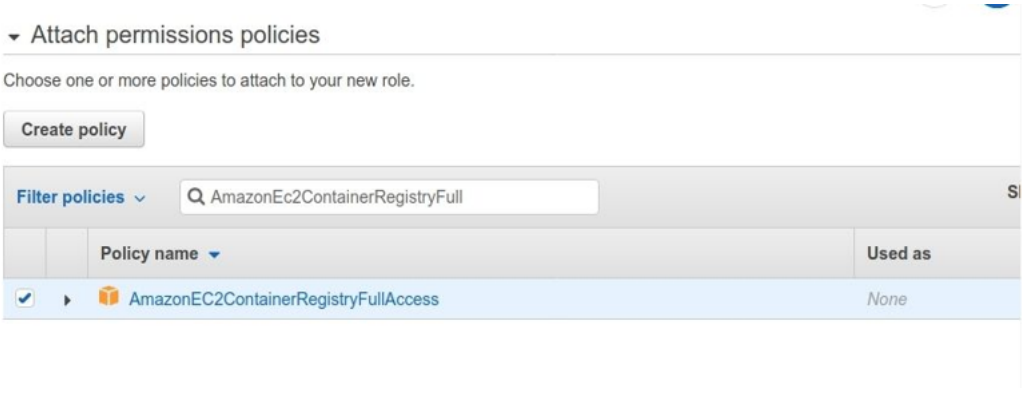
5. Una vez que se crea el repositorio, vaya a IAM

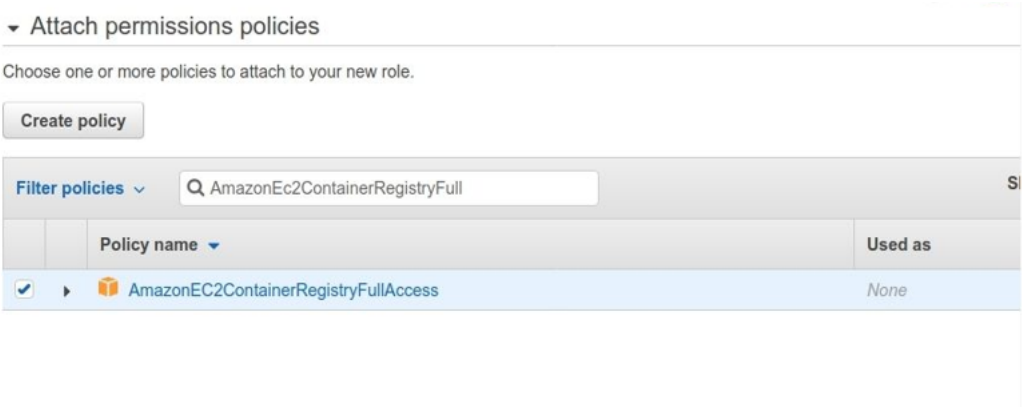


6. Pulsa en Roles

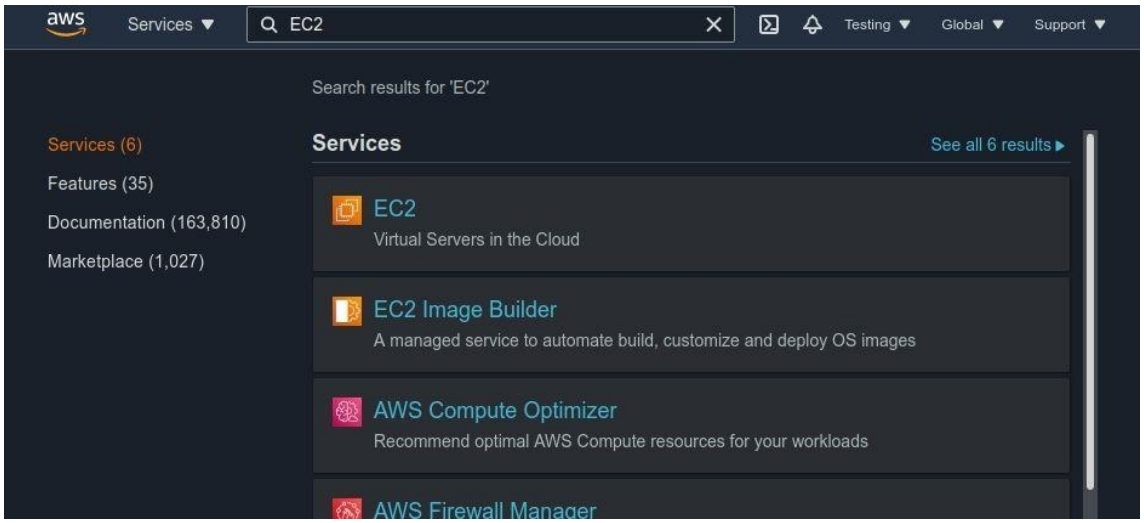


7. Agregue a LabRole estas políticas:

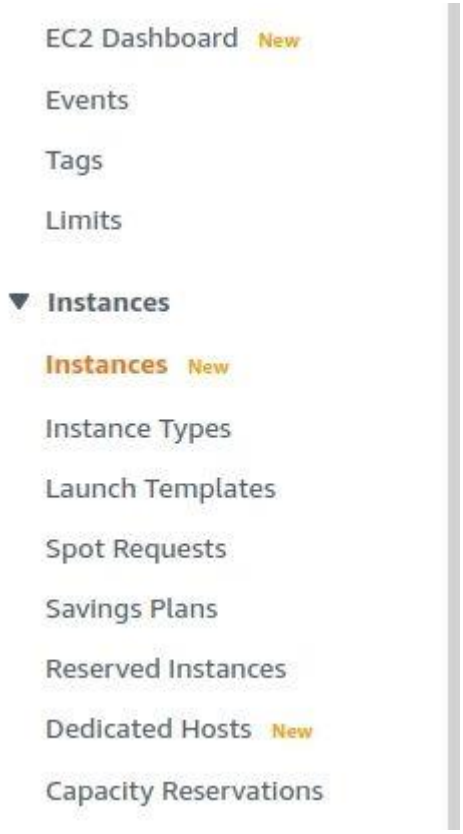




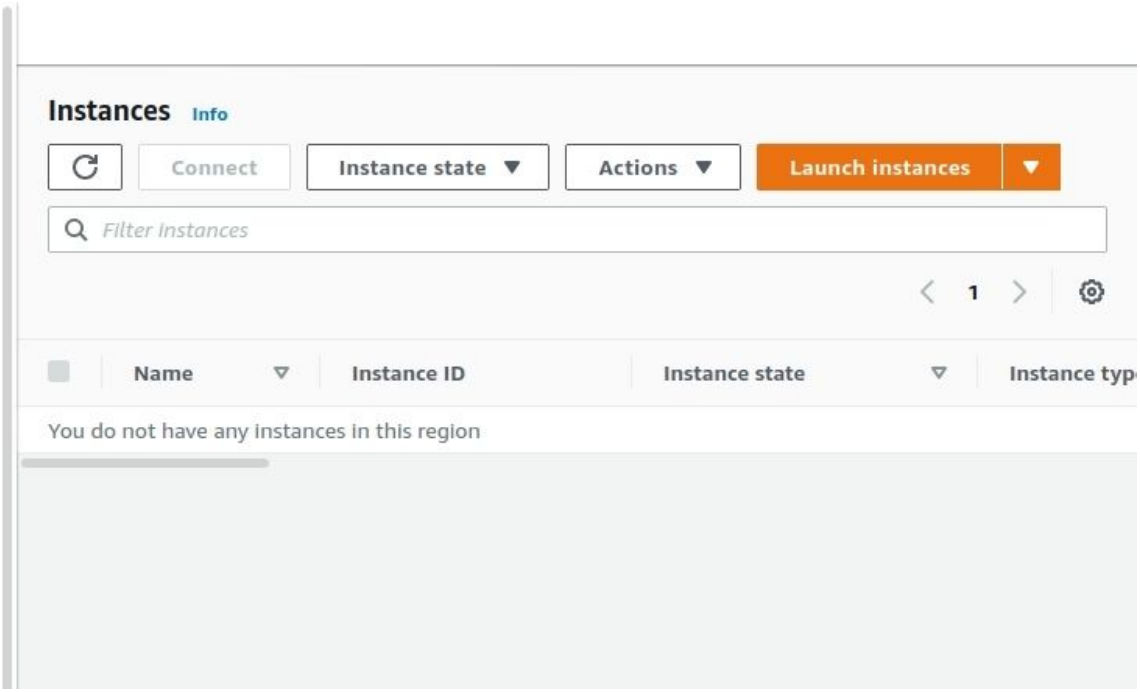
8. Pulsa en EC2



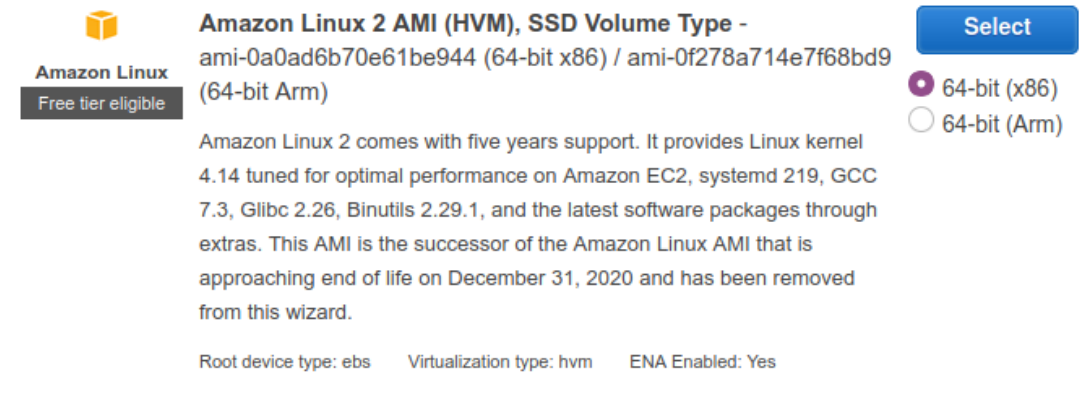
9. Ve a Instances



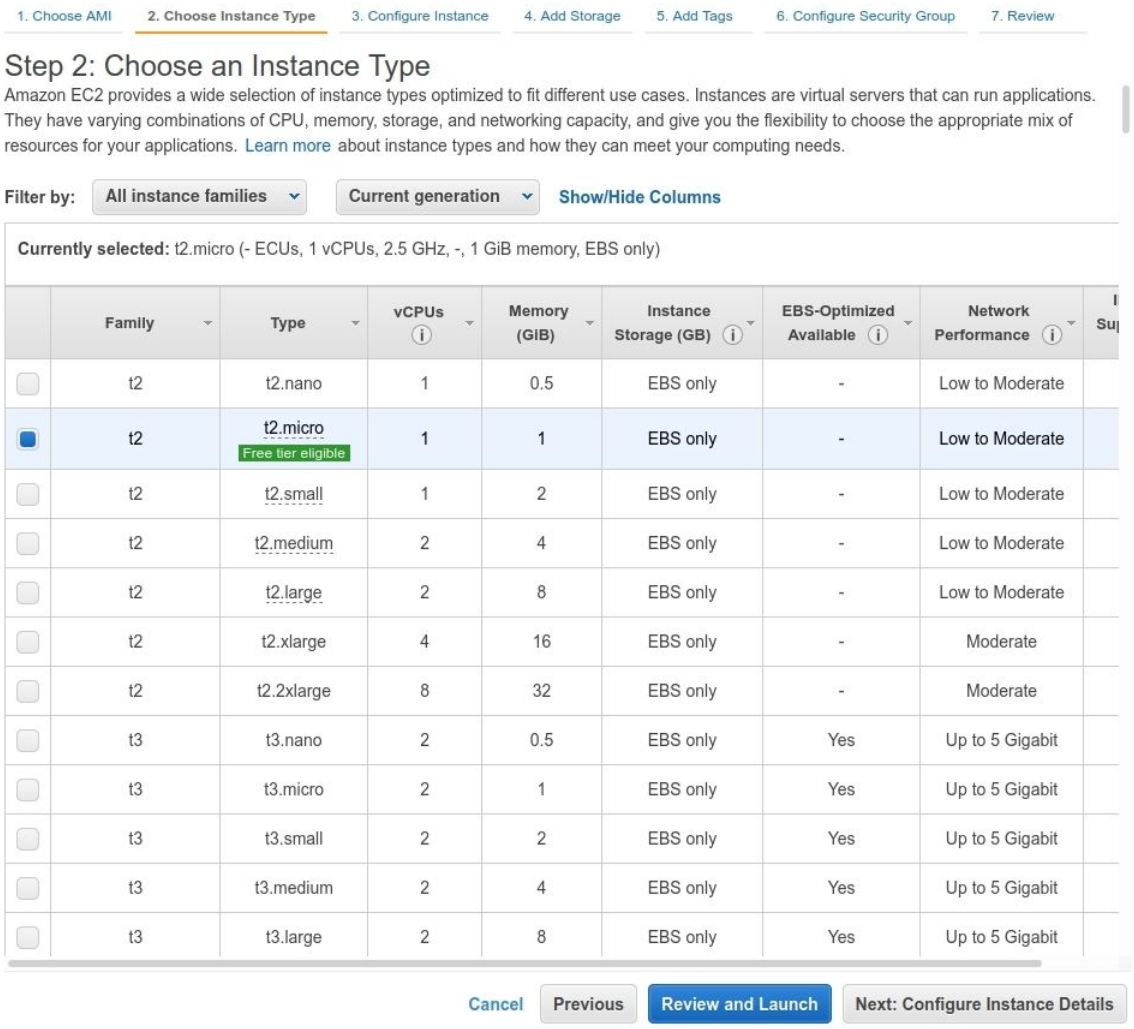
10. Pulsa en Launch instances



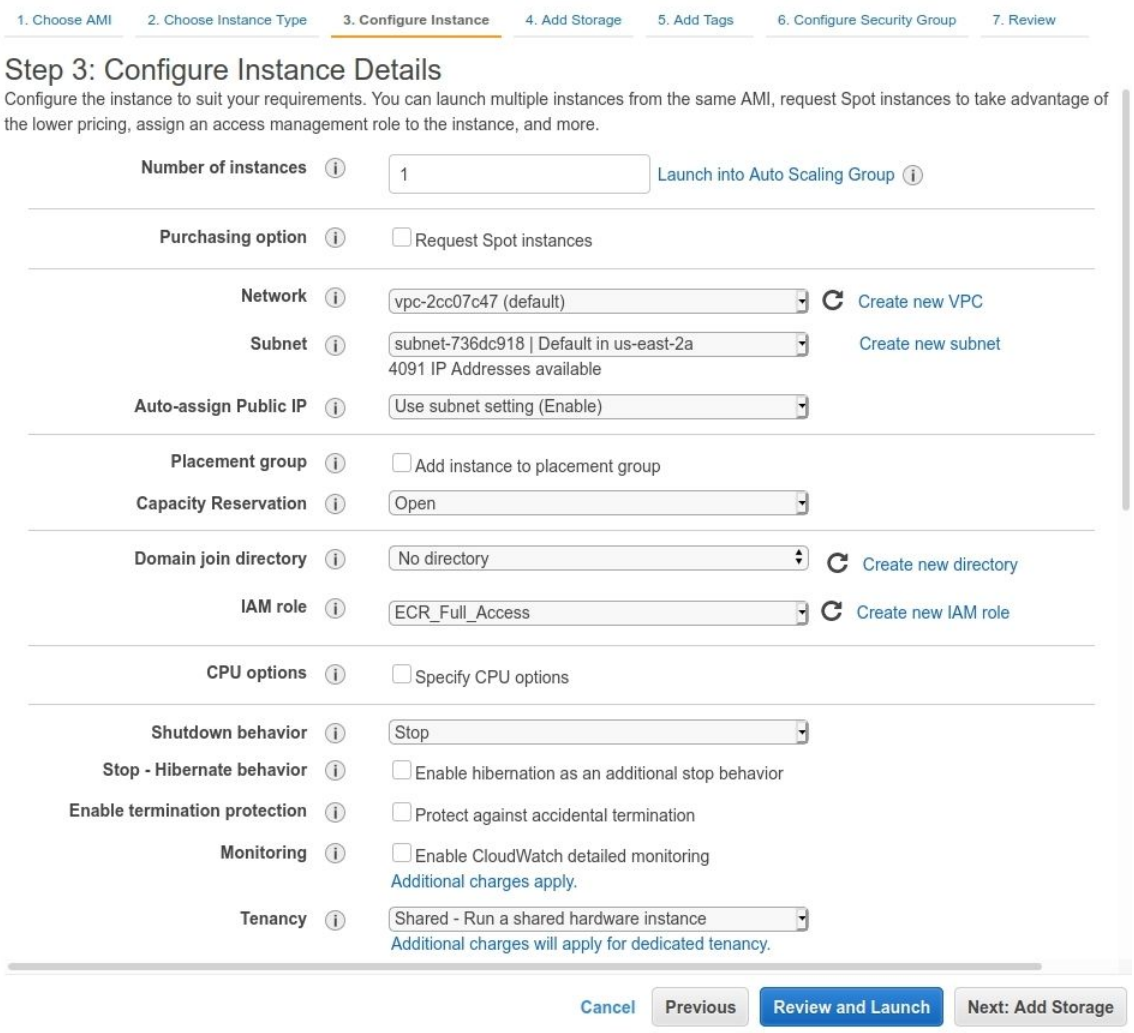
11. Seleccione Amazon Linux 2 AMI de 64 bits



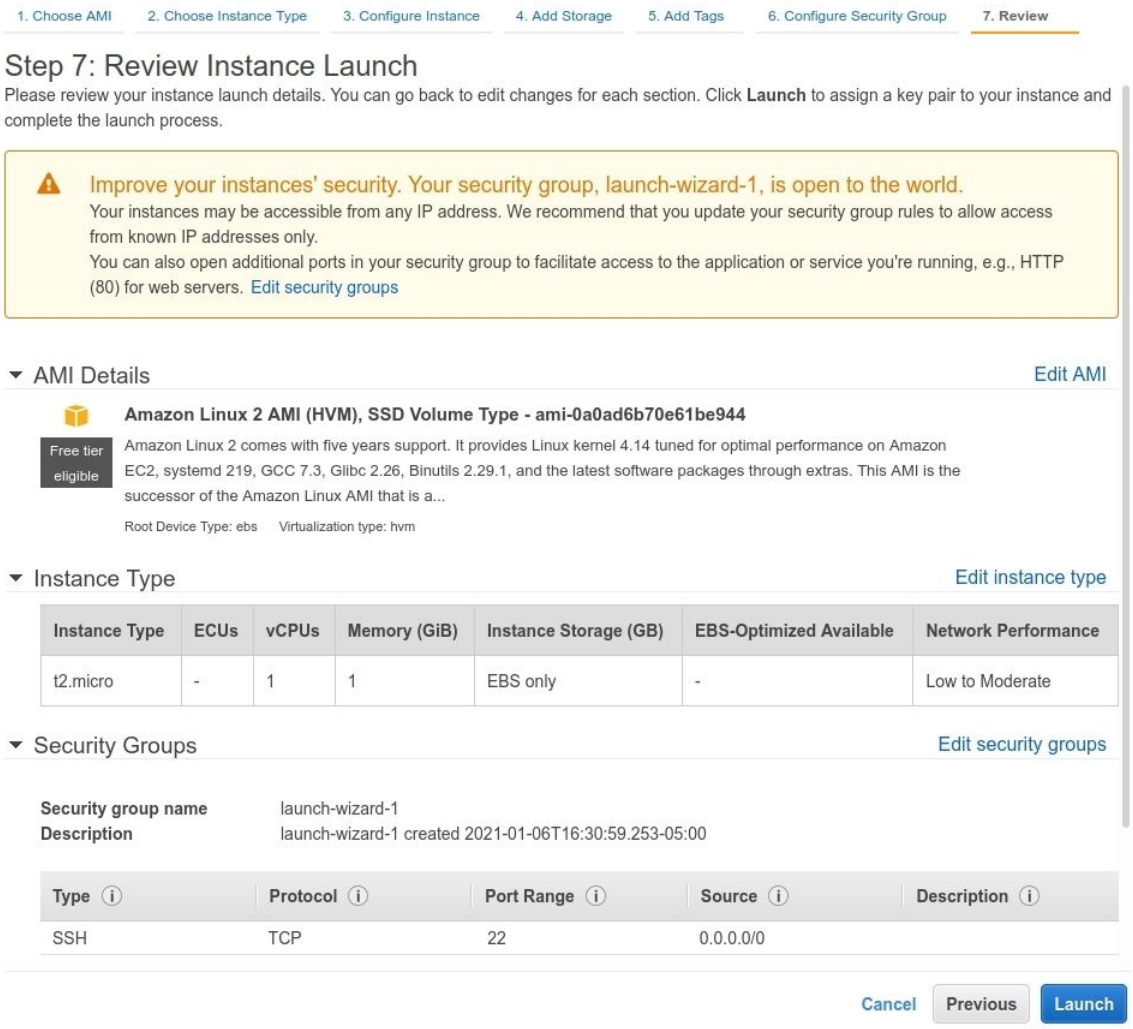
12. Seleccione t2.micro y haga clic en Siguiente



13. En Configurar detalles de la instancia, seleccione 1 instancia, asegúrese de tener habilitada la asignación automática de IP pública y, en el rol de IAM, seleccione el rol que creamos ECR_Full_Access y haga clic en Revisar y lanzar

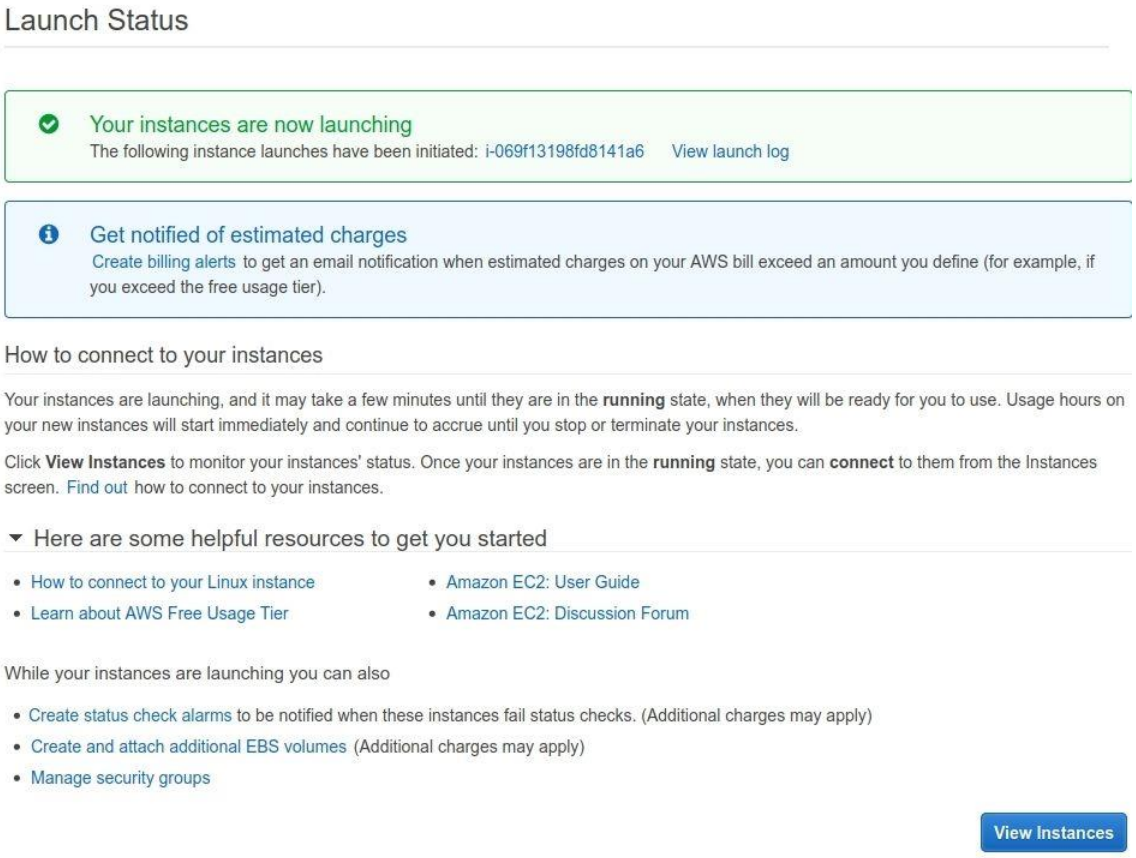


14. Revisa la configuración y haz clic en Iniciar

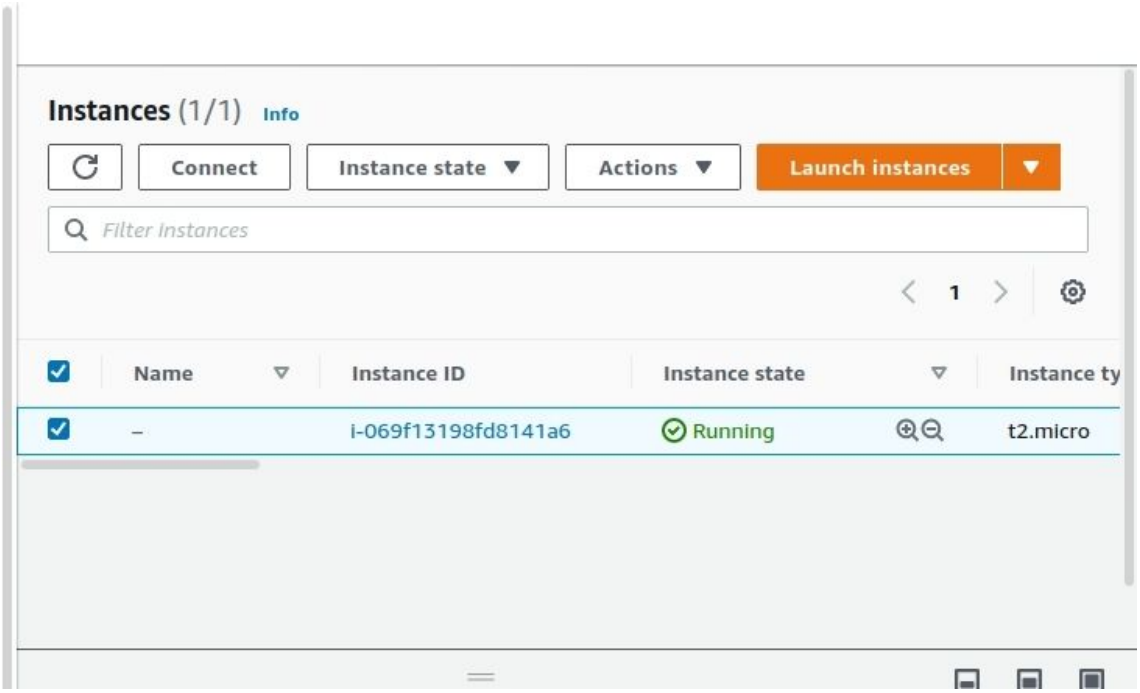


15. Usa el par de claves, Vockey.

16. Si todo está bien, debería ver el mensaje de éxito, haga clic en Ver instancias



17 -18-19. Seleccione la instancia y haga clic en Conectar



20. Conéctese a la instancia

```
ssh -i "ecr-test.pem" ec2-user@ec2-3-12-71-26.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-3-12-71-26.us-east-2.compute.amazonaws.com
(3.12.71.26)' can't be established.
ECDSA key fingerprint is
SHA256:1tnWyvB5JvWGBG/seJYL9bC2I3QDxiTtusalwmyn8js.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-12-71-26.us-east-
2.compute.amazonaws.com,3.12.71.26' (ECDSA) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-14-127 ~]$
```

21. Instala Docker

```
sudo su
yum -y install docker
```

22. Inicia Docker

```
systemctl start docker
```

(Nota: también lo podemos inicar con la instrucción “service docker start”)

22b. Descarga en la carpeta de trabajo una sencillala web, la descomprimimos y borramos el fichero .xip:

```
wget https://sanvalero-static-webs.s3.eu-west-1.amazonaws.com/Drink-Water.zip

unzip Drink-Water.zip

rm Drink-Water.zip
```

23. Cree un archivo llamado Dockerfile para crear una imagen Docker con un apache y la página web.

```
Vim Dockerfile
```

24. Copia los datos:

Utilizamos una imagen pública con PHP y Apache de Docker Enterprise. (https://hub.docker.com/ /php). Copiamos los ficheros de la carpeta de trabajo a la ruta del apache del Docker y lo exponemos por el puerto 80.

```
FROM php:7.0-apache
COPY . /var/www/html/
EXPOSE 80
```

26. Creamos la imagen desde Dockerfile con docker built -t (nombre) y listamos las imágenes.

(NOTA: el espacio y el punto después del nombre es importante)

```
docker build -t myapache .
docker images
```

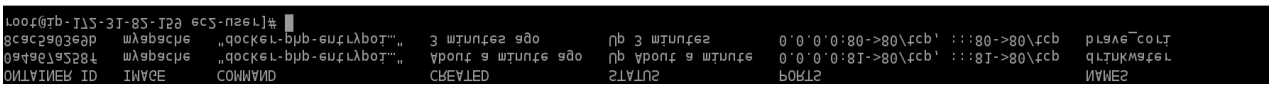
27. Ejecuta la imagen y lista los procesos. Ejecutamos el Docker en el puerto 80.

```
docker run -d -p 80:80 myapache
docker ps
```

(Opcional: también podríamos ejercutar esta otra instrucción:

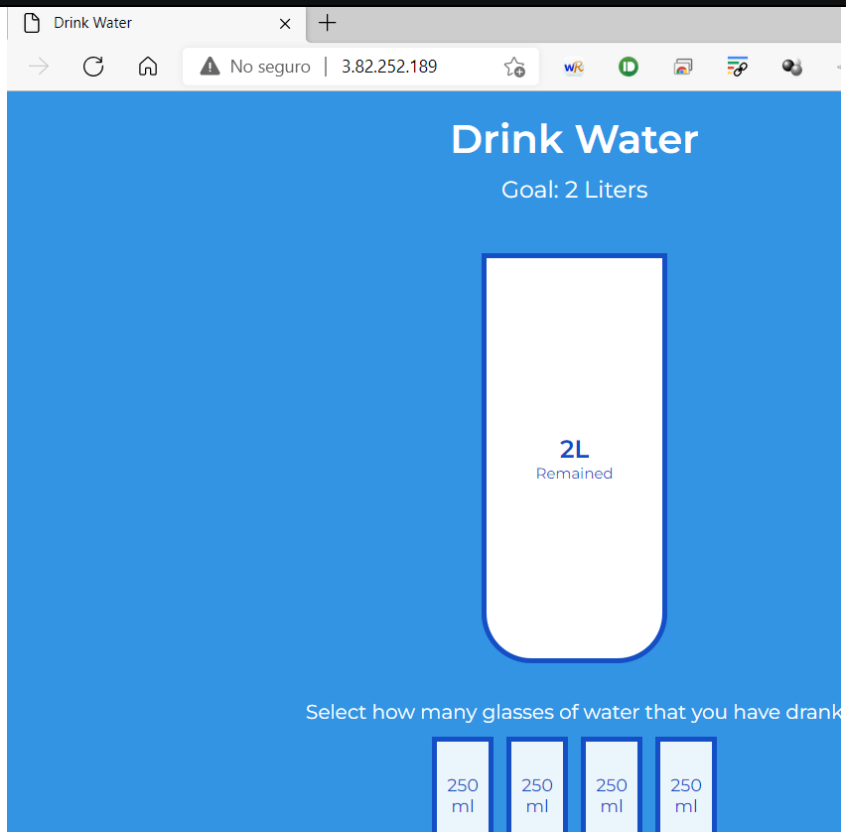
docker run -d -p 81:80 --name drinkwater myapache

y arrancaríamos la misma imagen y lo expondríamos en el puerto 81. Además la imagen myapache se levantaría como un proceso Docker con el nombre drinkwater. Lo podemos comprobar con Docker ps)



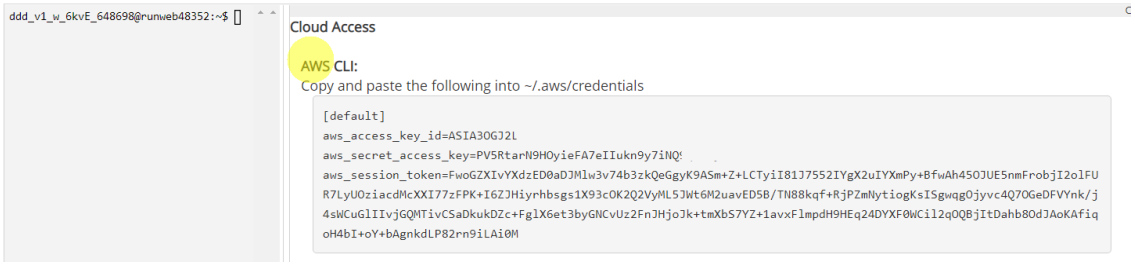
28. Pruebe que el contenedor se está adaptando con curl desde Linux o por navegador por la Ip de la EC2:

```
curl 172.0.0.1
```



29. Ejecutamos Aws configure en el Linux AMI e introducimos el acces key el secret acces key de AWS Academy. En la región, us-east-1 normalmente o la región donde tengamos el laboratorio creado y en tipo de fichero json.

Posteriormente modificamos el fichero ~/.aws/credentials y le agregamos el aws_session_token.



30. Obten el acceso de inicio de sesión al ECR con el comando aws ecr get-login donde xxxxxxx es el numero de cuenta de AWS.

```
aws ecr get-login-password | docker login --username AWS --password-stdin
xxxxxxx.dkr.ecr.us-east-1.amazonaws.com
```

31. Ahora etiquete la imagen con la imagen con el nombre ECR-DNS, donde xxxxx es el número de cuenta de AWS Academy.

```
docker tag myapache xxxxxxxxxxx.dkr.ecr.us-east-
1.amazonaws.com/testrepository:myapache
```

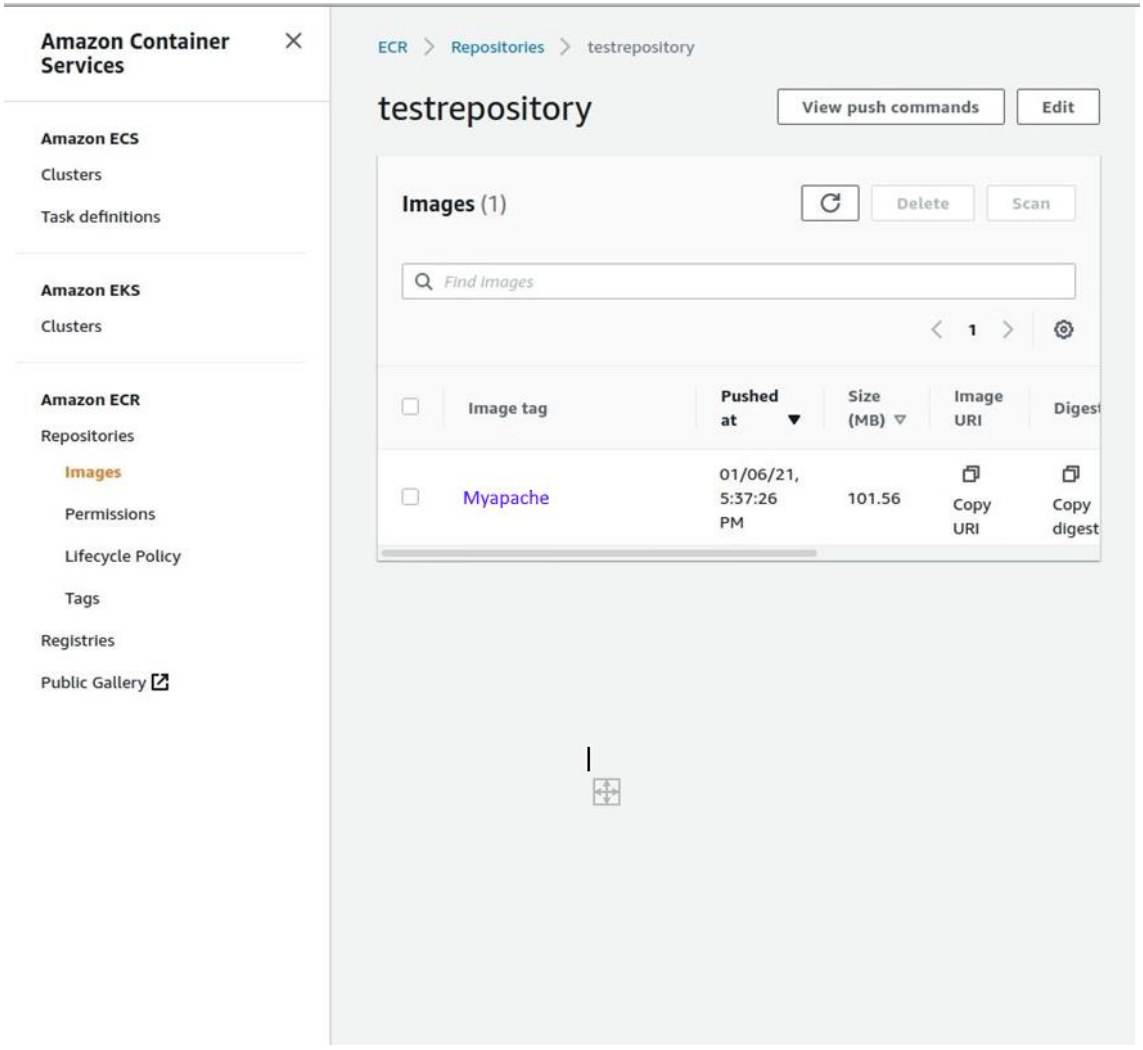
32. Puede verificar las imágenes con docker image ls

```
docker image ls
REPOSITORY                                                 TAG             IMAGE ID
CREATED          SIZE
625181428504.dkr.ecr.us-east-2.amazonaws.com/testrepository   latest
f79bc35853c9       8 minutes ago       260MB
myapache                                                   latest          f79bc35853c9       8
minutes ago       260MB
ubuntu                                                     16.04           9499db781771       6
weeks ago         131MB
```

33. Sube la imagen a la ECR donde xxxxx es el número de cuenta de AWS Academy.

```
docker push xxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com/testrepository:myapache
```

34. En la Administración de la consola de AWS, dentro del repositorio, verá la imagen de la ventana con el nombre de myapache)



Estupendo, aprenderá a crear un repositorio de ECR y a crear y cargar una imagen de Docker.