

APUNTES - NGINX

Creación de una instancia en AWS Academy, lanzando NGINX

1. Creamos nuestra instancia
2. Recordar elegir el archivo vockey.pem y crear grupo de seguridad con el tráfico permitido para SSH y HTTP.

3. Si queremos instalar directamente Nginx en nuestro servidor podemos añadir el siguiente código en Datos del usuario antes de lanzar la instancia.

4. O de esta manera

Vamos a instalar un servidor web en la instancia. Primero debemos conectarnos a la instancia con ssh.

Una vez conectados, si usamos Debian basta instalar `nginx-light` o `nginx` o similar:

```
admin@ip-172-31-50-215:~$ sudo apt update
...
admin@ip-172-31-50-215:~$ sudo apt install nginx
...
admin@ip-172-31-50-215:~$
```

5. Para conectarse mediante ssh

Conectarse a la instancia Información

Conéctese a la instancia i-06de042ba30c0aad8 (testeo-gninx) mediante cualquiera de estas opciones


Conexión de la instancia EC2



Administrador de sesiones

Ciente SSH


Consola de serie de EC2


ID de la instancia

 i-06de042ba30c0aad8 (testeo-gninx)

1. Abra un cliente SSH.
2. Localice el archivo de clave privada. La clave utilizada para lanzar esta instancia es vockey.pem
3. Ejecute este comando, si es necesario, para garantizar que la clave no se pueda ver públicamente.
 `chmod 400 vockey.pem`
4. Conéctese a la instancia mediante su DNS público:
 `ec2-54-90-247-94.compute-1.amazonaws.com`

Ejemplo:

 `ssh -i "vockey.pem" admin@ec2-54-90-247-94.compute-1.amazonaws.com`

 **Nota:** En la mayoría de los casos, el nombre de usuario adivinado es correcto. Sin embargo, lea las instrucciones de uso de la AMI para comprobar si el propietario de la AMI ha cambiado el nombre de usuario predeterminado de la AMI.

6. Para poder cambiar el contenido de esa web.

Admin

Entramos en /var/www/html.

Borramos el contenido de la web.

Local

Luego para copiar el archivo con el comando :

```
scp -i vockey.pem [Archivo] admin@ec2-xx-xx-xxx-xx.... :[Archivo]
```

Admin

Descomprimos el zip con el comando

```
sudo unzip /home/admin/[Archivo]
```

```
admin@ip-172-31-32-241:~$ cd /var/www/html/

admin@ip-172-31-32-241:/var/www/html$ sudo rm index.nginx-debian.html

clases@daw2:~/Descargas$ scp -i ../Escritorio/vockey.pem Netflix.zip admin@ec2-54-90-247-94.compute-1.amazonaws.com:Netflix.zip
Netflix.zip                                100% 1993    7.2KB/s   00:00

admin@ip-172-31-32-241:/var/www/html$ sudo unzip /home/admin/Netflix.zip
Archive:  /home/admin/Netflix.zip
  inflating: index.html
  inflating: script.js
  inflating: style.css
```

APUNTES - DOCKER.NGINX

Creación de una instancia en AWS Academy, usando Docker
Lanzamos la imagen de un servidor nginx redireccionando el puerto 80

Creamos la instancia, ssh y tambien http de permisos.
Entramos en ella con el ssh

```
clases@daw2:~/Escritorio$ ssh -i "vockey.pem" admin@ec2-3-82-241-222.compute-1
.amazonaws.com
Linux ip-172-31-92-37 6.1.0-13-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.5
5-1 (2023-09-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
admin@ip-172-31-92-37:~$
```

Instalamos docker

```
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
```

```
admin@ip-172-31-92-37:~$ curl -fsSL https://get.docker.com -o get-docker.sh
admin@ip-172-31-92-37:~$ sudo sh get-docker.sh
# Executing docker install script, commit: e5543d473431b782227f8908005543bb4389b8de
+ sh -c apt-get update -qq >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq apt-transport-https ca-certificat
es curl gnupg >/dev/null
+ sh -c install -m 0755 -d /etc/apt/keyrings
```

Continuamos la instalación añadiendo el usuario al grupo Docker (es posible que ya esté) y cerrando la conexión para que se reflejen los cambios (o abrimos una segunda conexión o ejecutamos el comando newgrp docker).

```
sudo groupadd docker
sudo usermod -aG docker $USER
exit
```

```
admin@ip-172-31-92-37:~$ sudo groupadd docker
groupadd: group 'docker' already exists
admin@ip-172-31-92-37:~$ sudo usermod -aG docker $USER
admin@ip-172-31-92-37:~$ exit
logout
Connection to ec2-3-82-241-222.compute-1.amazonaws.com closed.
clases@daw2:~/Escritorio$
```

Volvemos a entrar en la instancia

```
clases@daw2:~/Escritorio$ ssh -i "vockey.pem" admin@ec2-3-82-241-222.compute-1.amazonaws.com
Linux ip-172-31-92-37 6.1.0-13-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1 (2023-09-29)
x86_64

The programs included with the Debian GNU/Linux system are free software;
```

Probamos el comando *docker*

```
admin@ip-172-31-92-37:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES

admin@ip-172-31-92-37:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:3155e04f30ad5e4629fac67d6789f8809d74fea22d4e9a82f757d28cee79e0c5
```

Lanzamos la imagen de un servidor nginx redireccionando el puerto 80

docker run -d --name web -p 80:80 nginx

```
admin@ip-172-31-92-37:~$ docker run -d --name web -p 80:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
1f7ce2fa46ab: Pull complete
9b16c94bb686: Pull complete
```

Comprobamos que está activa:

```
admin@ip-172-31-92-37:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
a86cd16fde44   nginx     "/docker-entrypoint..." 34 seconds ago    Up 33 seconds    0.0.0.0:80->80/tcp, :::80->80/tcp    web
```

Probamos acceder al contenedor en local:

```
admin@ip-172-31-92-37:~$ curl localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
```

Y probamos el acceso desde el exterior usando un navegador y los datos públicos de la instancia:



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

No olvide detener el contenedor (docker stop), detener y terminar la instancia, cerrar la consola y salir del laboratorio.

```
admin@ip-172-31-92-37:~$ docker stop web
web
admin@ip-172-31-92-37:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
admin@ip-172-31-92-37:~$
```

En este caso por ejemplo podría actualizar los repositorios e instalar docker. El script sería similar a el archivo que esta en 2-DAW/DAW/INSTALACIONES

En la pestaña "Inbound rules" o "Reglas de entrada", agrega reglas que permitan el tráfico en los puertos que necesitas abrir. Por ejemplo, si Traefik utiliza los puertos 80 y 8080, deberías agregar las siguientes reglas:

Tipo: HTTP, Puerto: 80, Origen: 0.0.0.0/0 (para permitir acceso desde cualquier lugar)

Tipo: Personalizado TCP, Puerto: 8080, Origen: 0.0.0.0/0 (para permitir acceso a la interfaz web de Traefik desde cualquier lugar)
para poder acceder a esos puertos