

Práctica 1.4

Nube Amazon (AWS)

Utilizando el entorno cloud de AWS, albergaré el servidor web realizado en la práctica 1.3.

En primer lugar, los recursos estarán asociados a la región predeterminada (US East - N. Virginia), dado que utilizando una europea he tenido problemas a la hora de lanzar la instancia. Por tanto, entiendo que los niveles de latencia no son los más bajos posibles.

El sistema operativo y VM escogidos han sido los indicados en la práctica (Ubuntu Server 16.04 y t2.micro), y el nombre del fichero *.pem* es *“servidor_web.pem”*.

En cuanto a los grupos de seguridad, habilitamos el acceso por SSH a la máquina que viene predeterminado. He cambiado el destino a *Anywhere* (0.0.0.0/0), de esta forma podremos realizar la conexión SSH desde cualquier dirección IP, pero sólo podremos hacerlo nosotros con el archivo de incryptación. También añadimos el acceso HTTP con destino *Anywhere* también, ya que queremos acceder a la página web de nuestro servidor desde cualquier sitio:

▼ Inbound rules			
<input type="text" value="Filter rules"/>			
Port range	Protocol	Source	Security groups
80	TCP	0.0.0.0/0	launch-wizard-4
80	TCP	::/0	launch-wizard-4
22	TCP	0.0.0.0/0	launch-wizard-4
22	TCP	::/0	launch-wizard-4

▼ Outbound rules			
<input type="text" value="Filter rules"/>			
Port range	Protocol	Destination	Security groups
All	All	0.0.0.0/0	launch-wizard-4

Para poder acceder a la máquina virtual mediante SSH, tenemos que tener en cuenta que los archivos *.pem* solo pueden ser legibles por el administrador. Una vez nos aseguramos de que solo el administrador tiene permiso de lectura sobre el fichero correspondiente, podemos acceder a la máquina virtual a través de SSH mediante el fichero *.pem*, el nombre del usuario administrador y la URL de la instancia. En este caso: ***ssh -i "servidor_web.pem" ubuntu@ec2-Dir_IP_pública.compute-1.amazonaws.com:***

```
root@silviu-VirtualBox:/home/silviu/Descargas# chmod 400 servidor_web.pem
root@silviu-VirtualBox:/home/silviu/Descargas# ls -l
total 3340
-rw-rw-r-- 1 silviu silviu 548816 abr 24 16:57 a.jpg
-r----- 1 silviu silviu 1704 may 13 19:17 CER-EC2-VM.pem
-rw-rw-r-- 1 silviu silviu 21089 abr 24 16:45 Have_a_Great_Day.jpg
-r----- 1 silviu silviu 1700 may 13 19:56 servidor_web.pem
drwxrwxr-x 7 silviu silviu 4096 abr 6 20:31 ServTelematicos-main
-rw-rw-r-- 1 silviu silviu 2829893 may 6 11:07 ServTelematicos-main.zip
-rw-rw-r-- 1 silviu silviu 129 abr 29 20:00 welcome.php
root@silviu-VirtualBox:/home/silviu/Descargas# ssh -i "servidor_web.pem" ubuntu@ec2-3-80-35-89.compute-1.amazonaws.com
The authenticity of host 'ec2-3-80-35-89.compute-1.amazonaws.com (3.80.35.89)' can't be established.
ECDSA key fingerprint is SHA256:Mq5GTLwLN8Y89Lo4GSiWw7wFo6m0LbYQnfl1DvL00LQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? SHA256:Mq5GTLwLN8Y89Lo4GSiWw7wFo6m0LbYQnfl1DvL00LQ
Warning: Permanently added 'ec2-3-80-35-89.compute-1.amazonaws.com,3.80.35.89' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-1128-aws x86_64)

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

0 packages can be updated.
0 of these updates are security updates.

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>".
```

Para copiar los ficheros que necesito para el servidor a la VM he utilizado SCP:

(Las direcciones IPs de la máquina virtual no coinciden ya que la captura de la copia de ficheros mediante SCP y la segunda captura, se hicieron en momentos diferentes)

```
root@silviu-VirtualBox:/home/silviu/Descargas# scp -i servidor_web.pem /home/silviu/Documentos/redes/Lab-de-redes-sistemas-y-servicios/p1.3/servidor/4servidorWeb.py ubuntu@ec2-3-80-35-89.compute-1.amazonaws.com:~/p1.4/
4servidorWeb.py 100% 9077 43.3KB/s 00:00
```

```
root@silviu-VirtualBox:/home/silviu/Descargas# ssh -i "servidor_web.pem" ubuntu@ec2-34-207-82-197.compute-1.amazonaws.com
The authenticity of host 'ec2-34-207-82-197.compute-1.amazonaws.com (34.207.82.197)' can't be established.
ECDSA key fingerprint is SHA256:Mq5GTLwLN8Y89Lo4GSiWw7wFo6m0LbYQnfl1DvL00LQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-207-82-197.compute-1.amazonaws.com,34.207.82.197' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-1128-aws x86_64)

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

14 packages can be updated.
2 of these updates are security updates.
To see these additional updates run: apt list --upgradable

*** System restart required ***
Last login: Thu May 13 18:23:50 2021 from 80.38.199.118
ubuntu@ip-172-31-24-169:~$ ls
p1.4
ubuntu@ip-172-31-24-169:~$ cd p1.4
ubuntu@ip-172-31-24-169:~/p1.4$ ls
4servidorWeb.py error_404.html Have_a_Great_Day.jpg index.html
ubuntu@ip-172-31-24-169:~/p1.4$
```

Una vez tenemos todos los ficheros necesarios, ejecutamos el script del servidor de la práctica 1.3. Le pasamos como parámetros el modo en el que va a trabajar el servidor (persistente o no persistente) y el puerto donde va a escuchar dicho servidor.

```
ubuntu@ip-172-31-24-169:~/p1.4$ sudo python3 4servidorWeb.py 1 80
```

```
if len(sys.argv) != 3:
    print('Usage:', sys.argv[0], '<Modo> <Server Port>\n')
    exit(1)

mode = int(sys.argv[1]) # El segundo parámetro establece el modo

if mode == 1:
    print('Modo persistente\n')
elif mode == 0:
    print('Modo no persistente\n')
else:
    print('Valor introducido inválido:\r')
    print('Modo persistente = 1\r')
    print('Modo no persistente = 0\n')
    exit(1)

ServPort = sys.argv[2]

# Socket TCP del servidor
ServSock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# Configurarlos para que no se bloquee
ServSock.setblocking(0)

# Unimos (bind) socket al puerto
ServSock.bind(('', int(ServPort)))
```

Asociamos el socket (función bind()) con la dirección del servidor. El puerto es el 80 y la dirección IP es la de la máquina virtual, pero no va a ser la misma dirección IP a la que se acceda desde el navegador. Para acceder desde el navegador se debe utilizar la dirección IP pública obtenida en la consola AWS (<http://Dir IP pública>).

Vídeo correspondiente al resultado de esta parte:

https://universidaddealcala-my.sharepoint.com/:v/g/personal/silviu_sofrone_edu_uah_es/ESvel06OqaNBs9XTpZcqE9EB4mDgG4QkckJvYSTQsmU61A?e=wPOEaj