Administración Remota de Servidores Web en AWS a Través de SSH

Estudiante: David Arbelaez
Instancia EC2: Practica4DavidArbelaez

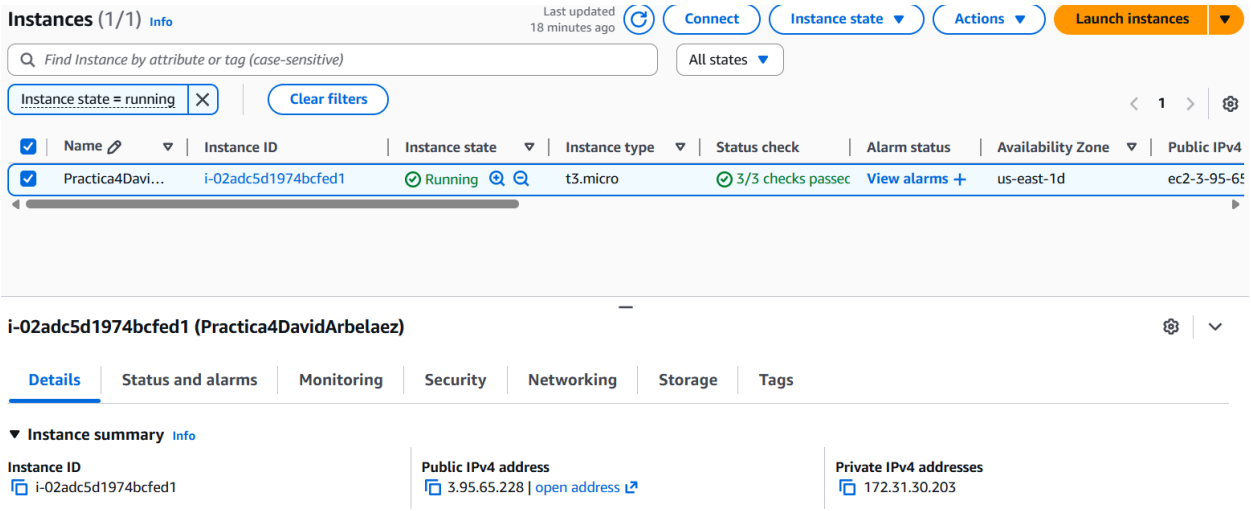
1. Configuración de la Instancia EC2 y Red en AWS

Esta sección detalla la creación de la infraestructura en la nube, asegurando que se cumplen las especificaciones de la práctica.

1.1. Creación de la Instancia EC2

Se creó una instancia de Amazon EC2, el servidor virtual donde se alojarán los servicios web, con las siguientes características:

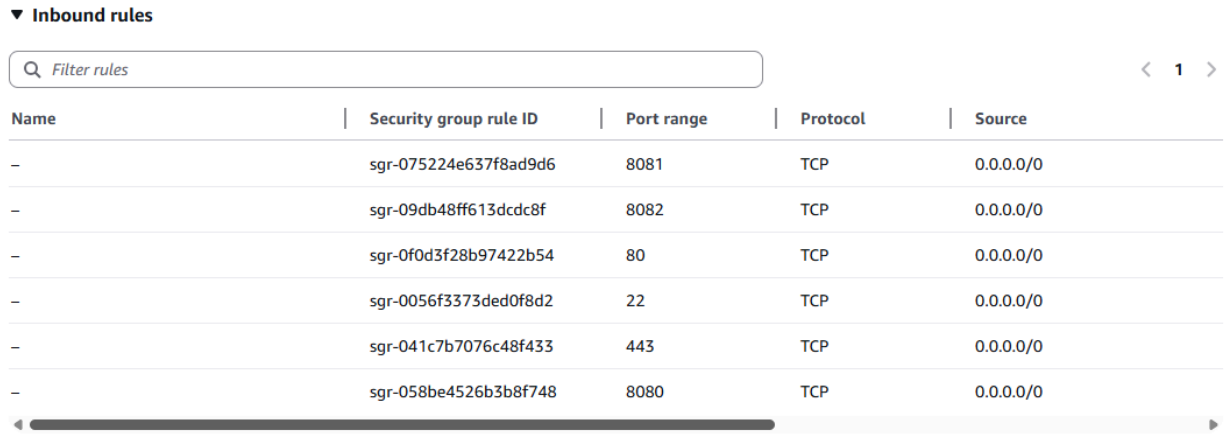
- **Nombre:** Practica4DavidArbelaez
- **AMI (Imagen):** Ubuntu 22.04 LTS (o 24.04 LTS), conforme a las directrices.
- **Tipo de Instancia:** t2.micro (dentro de la capa gratuita/Free Tier).
- **IP Pública:** Habilitada (necesaria para la conexión remota SSH y el acceso web).
- **Par de Claves:** labsuser.pem (clave de acceso seleccionada).



<Imagen 1. Creación de la máquina virtual, especificando AMI Ubuntu y la clave 'labsuser.pem'>

1.2. Configuración del Security Group (Reglas de Entrada)

Se configuró un Grupo de Seguridad (el "guardaespaldas" de la instancia) para definir qué tipo de tráfico se permite acceder al servidor. Se incluyeron un total de 5 reglas TCP de entrada, tal como lo requiere la práctica:



<Imagen 2. Configuración de las 5 reglas de entrada del Security Group para acceso web y SSH>

2. Gestión de la Clave de Acceso PEM en WSL

La gestión de la clave privada (.pem) es crucial para garantizar una conexión segura, ya que AWS no almacena esta clave.

2.1. Descarga y Posición de la Clave

La clave privada (labsuser.pem) fue descargada durante el proceso de creación del par de claves y movida a la ubicación estándar de claves SSH en el entorno WSL, asegurando que el cliente OpenSSH pueda encontrarla.

Comandos ejecutados en WSL:

```
# Creación del directorio y asignación de permisos solo para el propietario
mkdir -p ~/.ssh
chmod 700 ~/.ssh

# Movimiento del archivo .pem
cp /mnt/c/labsuser.pem ~/.ssh/
```

2.2. Aplicación de Permisos Restrictivos

Se aplicaron los permisos estrictos (solo lectura para el propietario) a la clave privada. **Este paso fue vital para solucionar el error de autenticación inicial** (Permission denied (publickey)), ya que el cliente OpenSSH rechaza claves con permisos abiertos.

Comando de permisos:chmod 400 ~/.ssh/labsuser.pem

```
root@DESKTOP-IUQ70D4:~/.ssh# cp /mnt/c/labsuser.pem ~/.ssh/
root@DESKTOP-IUQ70D4:~/.ssh# ls
DavidKeypar.ppk  known_hosts  labsuser.pem
root@DESKTOP-IUQ70D4:~/.ssh# chmod 400 ~/.ssh/labsuser.pem
root@DESKTOP-IUQ70D4:~/.ssh# ls -la ~/.ssh/labsuser.pem
-r-------- 1 root root 1678 Nov  6 15:54 /root/.ssh/labsuser.pem
```

<Imagen 3. Verificación de permisos de la clave PEM (400) en el directorio ~/.ssh/ en WSL>

3. Conexión SSH Exitosa y Acceso Remoto

Una vez configurada la instancia y protegida la clave, se estableció la conexión remota desde WSL.

3.1. Establecimiento de la Conexión SSH

Se utilizó el comando ssh especificando la clave privada (-i), el usuario remoto (ubuntu para la AMI de Ubuntu), y la dirección IP pública de la instancia.

Comando de conexión: ssh -i ~/.ssh/labsuser.pem ubuntu@3.95.65.228

```
root@DESKTOP-IUQ70D4:~/.ssh# ssh -i ~/.ssh/labsuser.pem ubuntu@3.95.65.228
The authenticity of host '3.95.65.228 (3.95.65.228)' can't be established.
ED25519 key fingerprint is SHA256:QiPzRck+2RErT0oFKvqp5T9jAo+ZZvPP/3nUGWre9Fk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.95.65.228' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

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

 System information as of Thu Nov  6 14:54:58 UTC 2025

  System load:  0.09              Temperature:           -273.1 C
  Usage of /:   25.8% of 6.71GB   Processes:             120
  Memory usage: 22%               Users logged in:       0
  Swap usage:   0%                IPv4 address for ens5: 172.31.30.203

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 list of available updates is more than a week old.
To check for new updates run: sudo apt update


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.
```

<Imagen 4. Conexión SSH exitosa a la instancia 'Practica4DavidArbelaez' con la IP 3.95.65.228>

3.2. Ejecución de la Segunda Práctica (Administración Remota)

Una vez conectado, se procedió a ejecutar la instalación y configuración del servidore web (Apache.) directamente en la instancia EC2, simulando la administración remota de un servidor en producción.

Verificación remota:

Verificación de que Apache se está ejecutando en el servidor
systemctl status apache2

```
root@ip-172-31-30-203:/home/ubuntu# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Thu 2025-11-06 15:04:27 UTC; 20s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2276 (apache2)
      Tasks: 55 (limit: 1008)
     Memory: 5.3M (peak: 5.6M)
        CPU: 42ms
     CGroup: /system.slice/apache2.service
             ├─2276 /usr/sbin/apache2 -k start
             ├─2279 /usr/sbin/apache2 -k start
             └─2280 /usr/sbin/apache2 -k start

Nov 06 15:04:27 ip-172-31-30-203 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Nov 06 15:04:27 ip-172-31-30-203 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@ip-172-31-30-203:/home/ubuntu#
```

<Imagen 5. Verificación de que Apache se está ejecutando en el servidor>

Verificación de que el servicio escucha en los puertos configurados (80) con el comando sudo netstat -tuln | grep 80

```
root@ip-172-31-30-203:/home/ubuntu# sudo netstat -tuln | grep 80
tcp6       0      0 :::80                   :::*                    LISTEN
```

<Imagen 6. Verificación de que el servicio escucha>

Se editan los archivos de configuración de apache para que responda por el puerto 8080 inicialmente el 000-default.conf

```
  GNU nano 7.2                                                  000-default.conf
<VirtualHost *:8080>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html
```

<Imagen 7. Edición del puerto de escucha de apache archivo 000-default.conf>

Y posteriormente el ports.conf ubicado en /etc/apache2/ports.conf

```
  GNU nano 7.2                                                  /etc/apache2/ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 8080

<IfModule ssl_module>
        Listen 443
</IfModule>

<IfModule mod_gnutls.c>
        Listen 443
</IfModule>
```

<Imagen 8. Edición del puerto de escucha de apache archivo 000-default.conf>

Se valida que el servicio que responde por el puerto 8080 sea el correspondiente a Apache

```
root@ip-172-31-30-203:/etc/apache2/sites-available# sudo netstat -tuln | grep 8080
tcp6       0      0 :::8080                 :::*                    LISTEN
```

<Imagen 9. Validación del puerto 8080>

Prueba de Acceso Web al servidor de apache

<Imagen 10. Prueba de acceso Web a apache>

Se instala NGNIX, y se edita el archivo de configuración /etc/nginx/sites-enabled y se cambia el puerto de 8080 a 8081

```
# Default server configuration
#
server {
        listen 8081 default_server;
        listen [::]:8081 default_server;
```

<Imagen 11. Edición de archivo de NGNIX cambio de puerto 8080 al 8081>

Posteriormente se prueba el uso del puerto 8081

```
root@ip-172-31-30-203:/etc/apache2/sites-available# sudo netstat -tuln | grep 8081
tcp        0      0 0.0.0.0:8081            0.0.0.0:*               LISTEN
tcp6       0      0 :::8081                 :::*                    LISTEN
```

<Imagen 12. Prueba de uso del puerto 8081>

Por último se realiza la prueba de que el servicio de NGINX se esté ejecutando correctamente

```
root@ip-172-31-30-203:/etc/nginx/sites-enabled# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Tue 2025-11-11 13:26:16 UTC; 6s ago
       Docs: man:nginx(8)
    Process: 2036 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 2038 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 2039 (nginx)
      Tasks: 3 (limit: 1017)
     Memory: 2.4M (peak: 2.8M)
        CPU: 13ms
     CGroup: /system.slice/nginx.service
             ├─2039 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─2040 "nginx: worker process"
             └─2041 "nginx: worker process"
```

<Imagen 13. Prueba de el servicio de Nginx>

Se realiza la instalación de Caddy, se edita el archivo **/etc/caddy/Caddyfile** para que la escucha sea por el puerto 8082. Adicionalmente se prueba el estado del servicio con el systemctl status caddy.

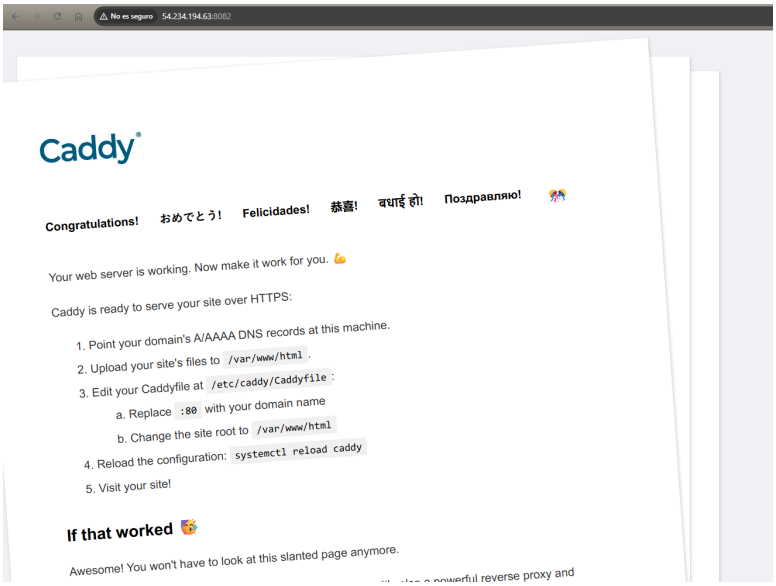
```
root@ip-172-31-30-203:/etc/apache2/sites-available# sudo netstat -tuln | grep 8082
tcp6       0      0 :::8082                 :::*                    LISTEN
```

<Imagen 13. Prueba de uso del puerto 8082>

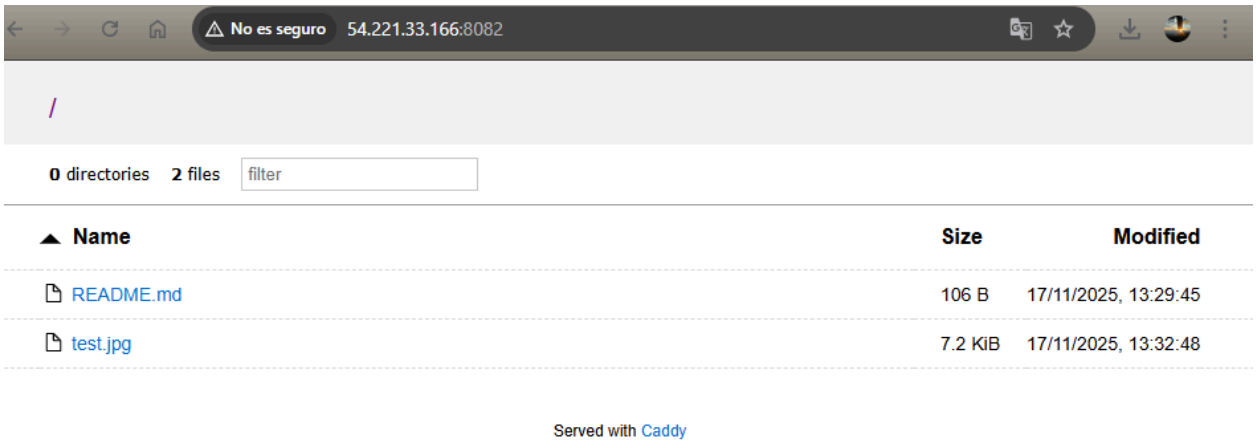
```
root@ip-172-31-30-203:/etc/nginx/sites-enabled# systemctl status caddy
● caddy.service - Caddy
     Loaded: loaded (/usr/lib/systemd/system/caddy.service; enabled; preset: enabled)
     Active: active (running) since Tue 2025-11-11 13:31:54 UTC; 7s ago
       Docs: https://caddyserver.com/docs/
   Main PID: 2864 (caddy)
      Tasks: 7 (limit: 1017)
     Memory: 6.7M (peak: 7.2M)
        CPU: 30ms
     CGroup: /system.slice/caddy.service
             └─2864 /usr/bin/caddy run --environ --config /etc/caddy/Caddyfile
```

<Imagen 14. Prueba de el servicio de Caddy>

Finalmente se realiza la prueba de acceso web y se crea la imagen test.jpg.



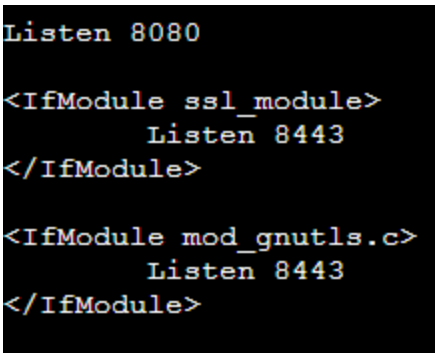
<Imagen 15. Prueba de acceso web a Caddy>



<Imagen 16. Prueba de acceso web a Caddy>

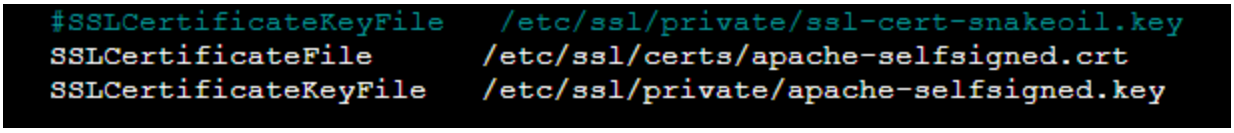
3.3 CONFIGURACIÓN DE HTTPS CON CERTBOT EN APACHE

Con el CERBOT vamos a generar un certificado SSL autofirmado para asociar a las solicitudes por https de apache. se dejan los valores del certificado con el CN de localhost, pais España y ciudad Madrid con el comando **sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt**



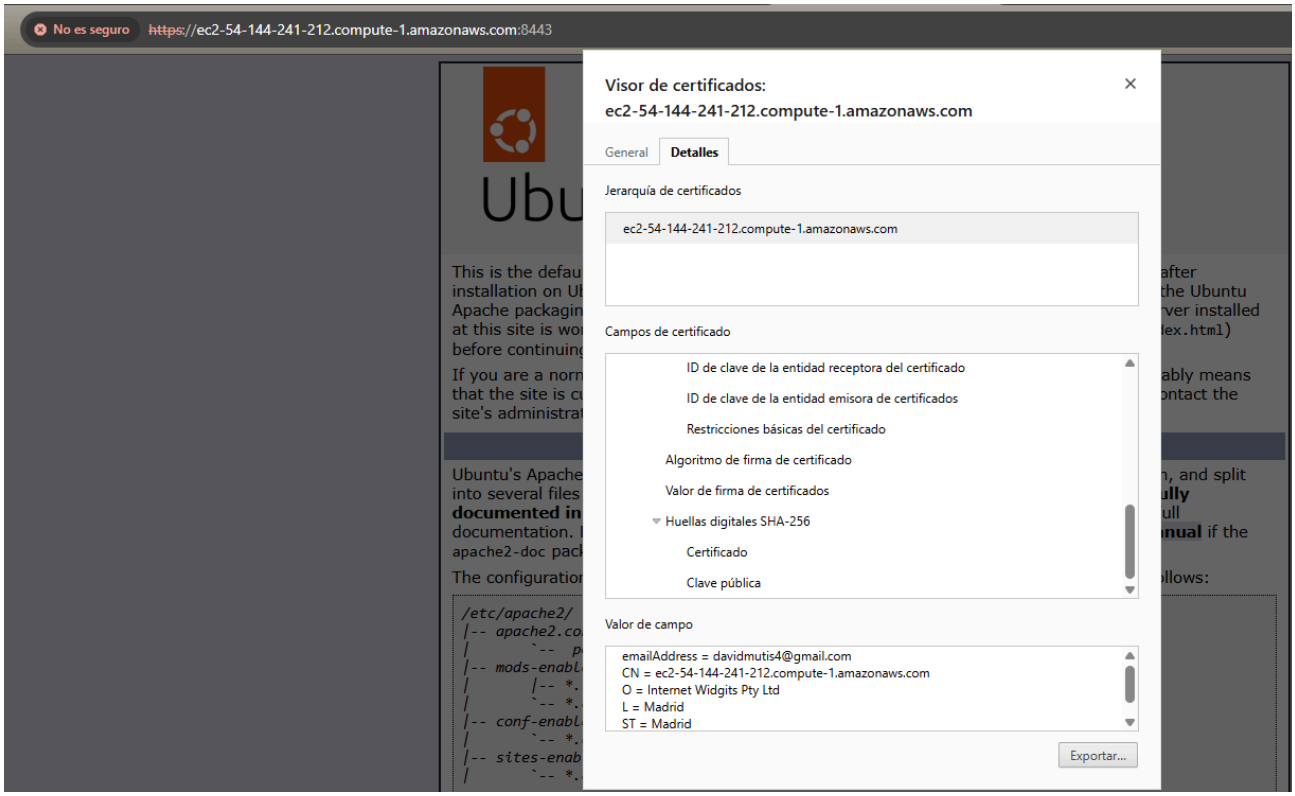
<Imagen 16. Configuración del puerto de escucha de apache>

Posteriormente se configura el certificado de apache en la ruta **/etc/apache2/sites-available/default-ssl.conf** y se cambia el puerto certificado que hay por defecto, por el que hemos generado en el paso anterior.



<Imagen 17. Cambio del certificado default por el apache-selfsigned.crt>

Se realiza la prueba de acceso por web a través del puerto 443 y se accede exitosamente.



<Imagen 18. Prueba de acceso por https a apache y visualización del certificado>

También se realizan las pruebas correspondientes al **systemctl status apache2** para verificar que el servicio esté en ejecución. Por otra parte se validan que puertos esta usando el apache con el comando **netstat -tulpn | grep apache2**

```
root@ip-172-31-30-203:/home/ubuntu# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Tue 2025-11-18 09:19:48 UTC; 7min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 3620 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 3623 (apache2)
      Tasks: 55 (limit: 1017)
     Memory: 9.0M (peak: 9.4M)
        CPU: 99ms
     CGroup: /system.slice/apache2.service
             ├─3623 /usr/sbin/apache2 -k start
             ├─3625 /usr/sbin/apache2 -k start
             └─3626 /usr/sbin/apache2 -k start

Nov 18 09:19:48 ip-172-31-30-203 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Nov 18 09:19:48 ip-172-31-30-203 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@ip-172-31-30-203:/home/ubuntu# sudo netstat -tulpn | grep apache2
tcp6       0      0 :::8080                 :::*                    LISTEN      3623/apache2
tcp6       0      0 :::8443                 :::*                    LISTEN      3623/apache2
```

<Imagen 19. Estado del servicio de apache y puertos de escucha>

4. Troubleshooting:

4.1 El error inicial fue diagnosticado y resuelto antes de proceder con el resto de la práctica.

Error Reportado (antes de la solución):

Load key "/root/.ssh/Davidkeypar.ppk": error in libcrypto
ubuntu@13.218.141.184: Permission denied (publickey).

Análisis y Solución:

1. **Error en libcrypto:** Se identificó el uso del formato de clave .ppk (PuTTY Private Key) que es incompatible con el cliente OpenSSH de WSL/Linux.
2. **Permission denied:** Se confirmó la necesidad de utilizar el archivo **.pem original** (formato OpenSSH) y aplicar estrictamente los permisos **chmod 400**.

La solución fue eliminar la instancia con la clave mal gestionada y crear una nueva (Practica4DavidArbelaez) asegurando el uso del archivo .pem original con los permisos correctos, lo que permitió la conexión SSH.

4.2 El segundo error fue al habilitar el certificado de apache y configurarlo para escuchar por el puerto 8443, el servicio no respondía por el acceso web.

Análisis y Solución: Se realiza el comando wget con la URL de apache y responde correctamente pero al momento de acceder por web no funciona correctamente

1. **Error en la configuración de los puertos:** Se identifica después de muchas pruebas que el problema se daba debido a que no se creó la regla en el firewall correspondiente al puerto 8443 y por esta razón no se conectaba por el navegador web. Posteriormente se crea la regla y funciona correctamente