

Ivan Castillo

S5: Laboratorio: Comandos de solución de problemas del protocolo de Internet

Objetivos

En este laboratorio usted:

- Practicar los comandos de solución de problemas
- Identificar cómo puede usar estos comandos en las situaciones del cliente

Los siguientes componentes se crean para usted como parte del entorno de laboratorio :

Amazon EC2: host de comandos (en la subred pública): inicie sesión en esta instancia para utilizar los comandos enumerados en esta práctica de laboratorio.

Tarea 1: utilizar SSH para conectarse a una instancia EC2 de Amazon Linux

En esta tarea, se conectará a una instancia EC2 de Amazon Linux. Utilizará una utilidad SSH para realizar todas estas operaciones. Las siguientes instrucciones varían ligeramente dependiendo de si está utilizando Windows o Mac/Linux.

USUARIOS DE WINDOWS: USO DE SSH PARA CONECTARSE

Estas instrucciones son específicamente para usuarios de Windows. Si está utilizando macOS o Linux, pase a la siguiente sección.

3. En el panel **Información del laboratorio** , seleccione el enlace **PPK** y guarde el archivo. El nombre del archivo será similar a *Ec2KeyPair-PPK.ppk* . Normalmente su navegador lo guardará en el directorio de Descargas.
4. Tome nota de la dirección **PublicIP** .
5. Descargue **PuTTY** a SSH en la instancia de Amazon EC2.
6. Abrir **PuTTY .exe**

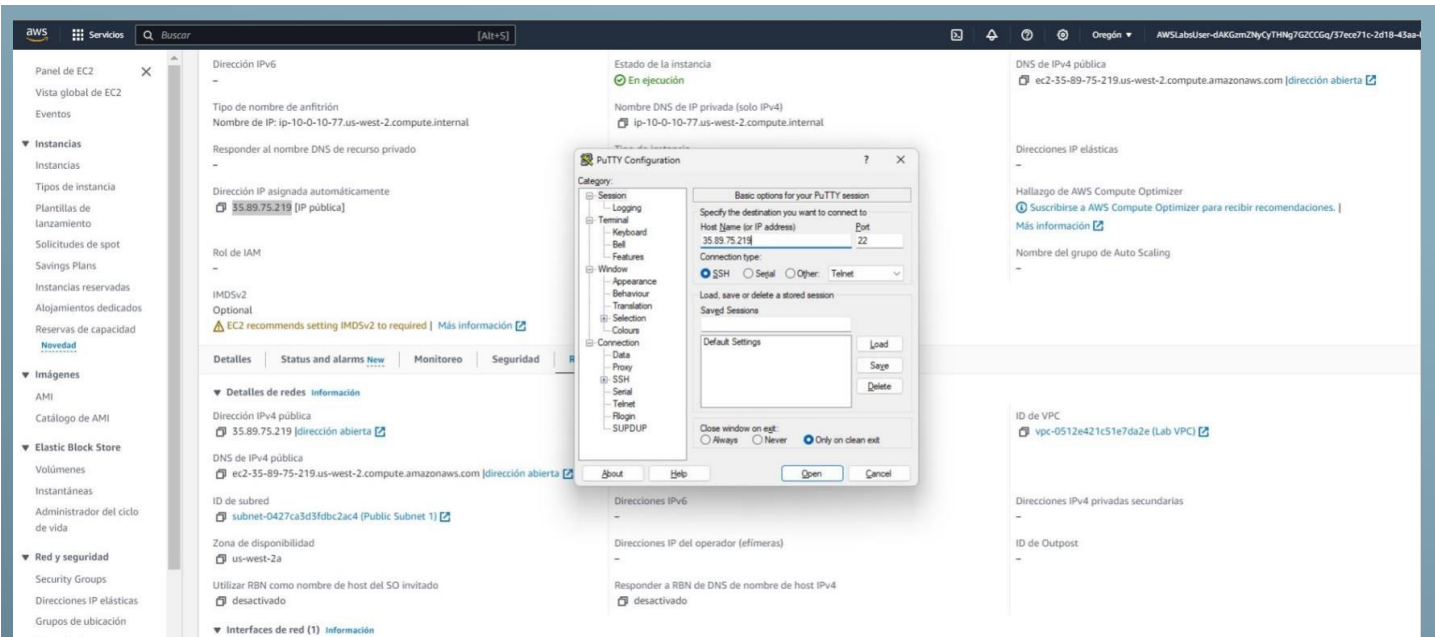
7. Configure el tiempo de espera de PuTTY para mantener abierta la sesión de PuTTY durante un período de tiempo más largo:

- Seleccionar **conexión**
- Establecer **segundos entre keepalives** en 30

The screenshot shows the AWS training and certification interface. On the left, there is a sidebar with a tree view containing categories like Session, Logging, Terminal, Keyboard, Bell, Features, Window, Appearance, Behaviour, Translation, Selection, Colours, Connection, Data, Proxy, SSH, Serial, Telnet, Rlogin, and SUPDUP. The main content area displays the 'Machine Image (AMI) (EN)' page. Overlaid on this is the 'PuTTY Configuration' dialog box. The 'Options controlling the connection' tab is active, showing 'Seconds between keepalives (0 to turn off)' set to 30. The 'Low-level TCP connection options' section has 'Disable Nagle's algorithm (TCP_NODELAY option)' checked and 'Enable TCP keepalives (SO_KEEPALIVE option)' unchecked. The 'Internet protocol version' is set to 'Auto'. The 'Logical name of remote host' and 'Logical name of remote host (e.g. for SSH key lookup)' fields are empty. A 'Copied' tooltip is visible over the '30' value in the keepalives field. The background page includes instructions for connecting to an Amazon Linux EC2 instance via SSH, mentioning the 'Public IP' address and providing a link to download PuTTY.

8. Configure su sesión PuTTY:

- Seleccionar **sesión**
- **Nombre de host (o dirección IP):** pegue el **DNS público o la dirección IPv4** de la instancia que anotó anteriormente. Alternativamente, regrese a la Consola EC2 y seleccione **Instancias** . Marque la casilla junto a la instancia a la que desea conectarse y en la pestaña *Descripción* copie el valor de **IP pública IPv4** .



10. Cuando se le solicite **iniciar sesión como** , ingrese:

ec2-user

Esto lo conectará a la instancia EC2.

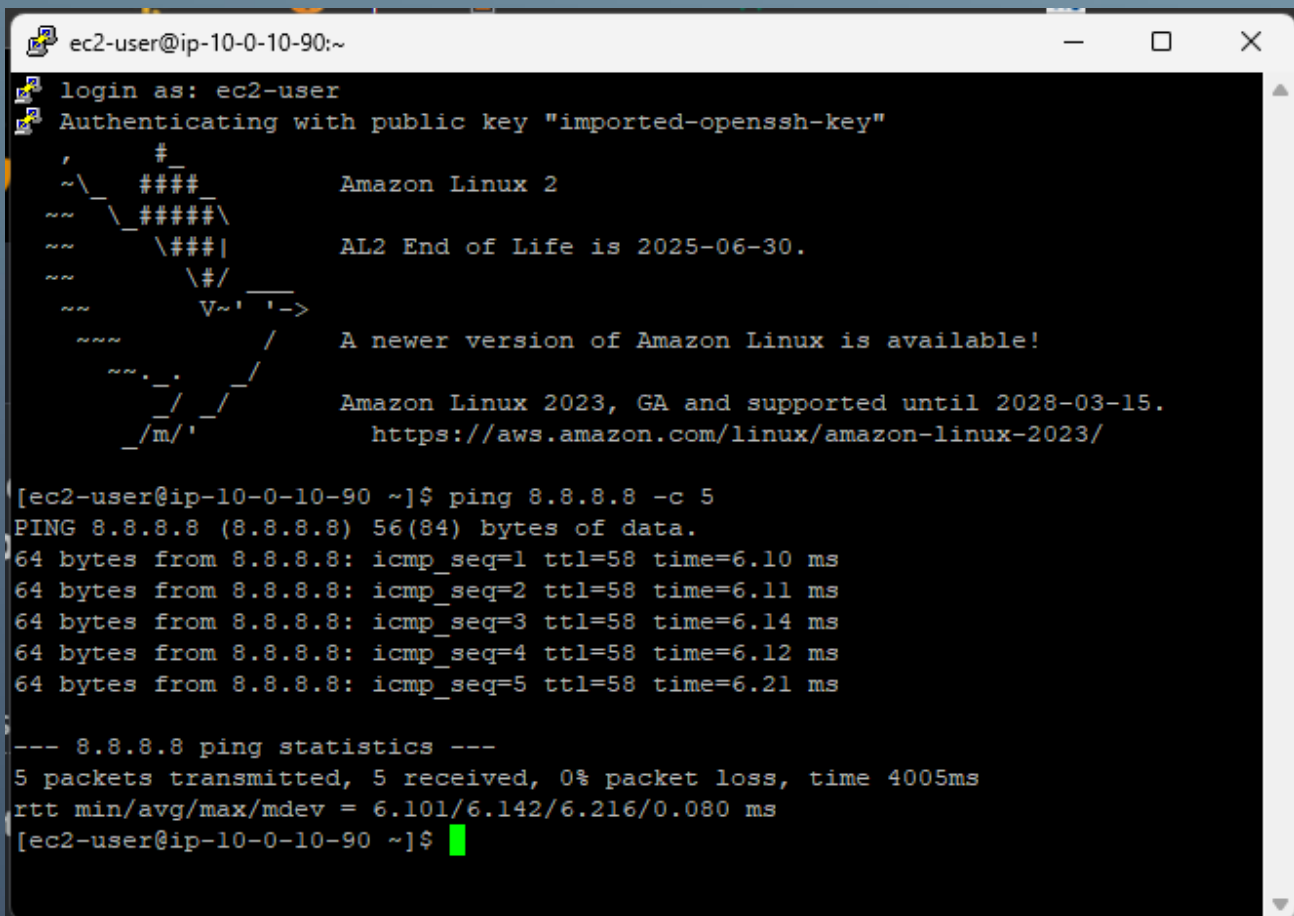
Tarea 2: Practicar los comandos de solución de problemas

1. El siguiente es un ejemplo de una situación de cliente en el que puede usar el comando ping:

El cliente ha lanzado una instancia de EC2. Para probar la conectividad hacia y desde la instancia, ejecute el comando ping. Puede usar este comando para probar la conectividad y asegurarse de que permite las solicitudes del Protocolo de mensajes de control de Internet (ICMP) en el nivel de seguridad, como grupos de seguridad y ACL de red.

En la terminal de Linux, ejecute el siguiente comando y presione Enter:

```
ping 8.8.8.8 -c 5
```



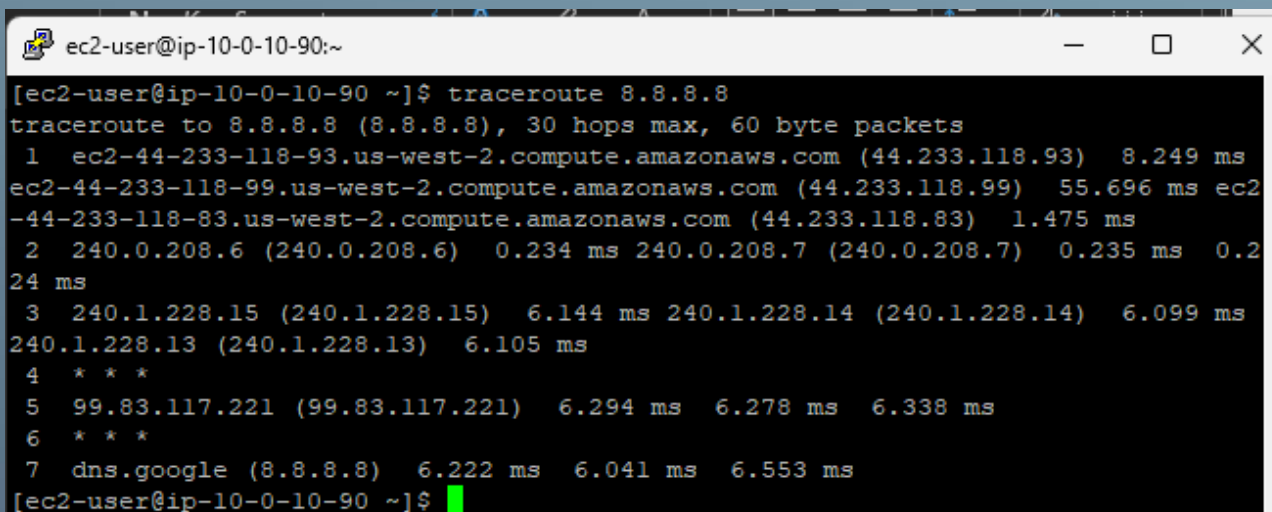
```
ec2-user@ip-10-0-10-90:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
#_      Amazon Linux 2  
~\_####  
~~\_#####  
~~\_###|    AL2 End of Life is 2025-06-30.  
~~\_#/        
~~_V~'-'>  
~~~      A newer version of Amazon Linux is available!  
~~.-.-/      Amazon Linux 2023, GA and supported until 2028-03-15.  
_/_/_/_/      https://aws.amazon.com/linux/amazon-linux-2023/  
_/_/_/_/_/        
[ec2-user@ip-10-0-10-90 ~]$ ping 8.8.8.8 -c 5  
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data:  
64 bytes from 8.8.8.8: icmp_seq=1 ttl=58 time=6.10 ms  
64 bytes from 8.8.8.8: icmp_seq=2 ttl=58 time=6.11 ms  
64 bytes from 8.8.8.8: icmp_seq=3 ttl=58 time=6.14 ms  
64 bytes from 8.8.8.8: icmp_seq=4 ttl=58 time=6.12 ms  
64 bytes from 8.8.8.8: icmp_seq=5 ttl=58 time=6.21 ms  
  
--- 8.8.8.8 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 4005ms  
rtt min/avg/max/mdev = 6.101/6.142/6.216/0.080 ms  
[ec2-user@ip-10-0-10-90 ~]$
```

2. El siguiente es un ejemplo de una situación de cliente en el que puede usar el comando traceroute:

El cliente tiene problemas de latencia. Dice que su conexión tarda mucho y que pierde paquetes. No está seguro de si está relacionado con AWS o con su proveedor de servicios de Internet (ISP). Para investigar, puede ejecutar el comando traceroute desde su recurso de AWS al servidor al que intentan acceder. Si la pérdida ocurre hacia el servidor, lo más probable es que el problema sea el ISP. Si la pérdida es para AWS, es posible que deba investigar otros factores que pudieran limitar la conectividad de red.

En la terminal de Linux, ejecute el siguiente comando y presione Enter:

```
traceroute 8.8.8.8
```



The screenshot shows a terminal window titled "ec2-user@ip-10-0-10-90:~". The user has entered the command `traceroute 8.8.8.8`. The output shows the path from the EC2 instance to the destination IP 8.8.8.8. The first hop is an AWS endpoint with a latency of 8.249 ms. The second hop is another AWS endpoint with a latency of 55.696 ms. The third hop is a third AWS endpoint with a latency of 1.475 ms. The fourth hop is a public IP 240.0.208.6 with a latency of 0.234 ms. The fifth hop is another public IP 240.0.208.7 with a latency of 0.235 ms. The sixth hop is a public IP 240.1.228.15 with a latency of 6.144 ms. The seventh hop is another public IP 240.1.228.14 with a latency of 6.099 ms. The eighth hop is a public IP 240.1.228.13 with a latency of 6.105 ms. The ninth hop is a public IP 99.83.117.221 with a latency of 6.294 ms. The tenth hop is a public IP 99.83.117.221 with a latency of 6.278 ms. The eleventh hop is a public IP 99.83.117.221 with a latency of 6.338 ms. The twelfth hop is a public IP 99.83.117.221 with a latency of 6.222 ms. The thirteenth hop is a public IP 99.83.117.221 with a latency of 6.041 ms. The fourteenth hop is a public IP 99.83.117.221 with a latency of 6.553 ms. The final hop is the destination IP 8.8.8.8 with a latency of 6.222 ms. The terminal shows a green cursor at the end of the command line.

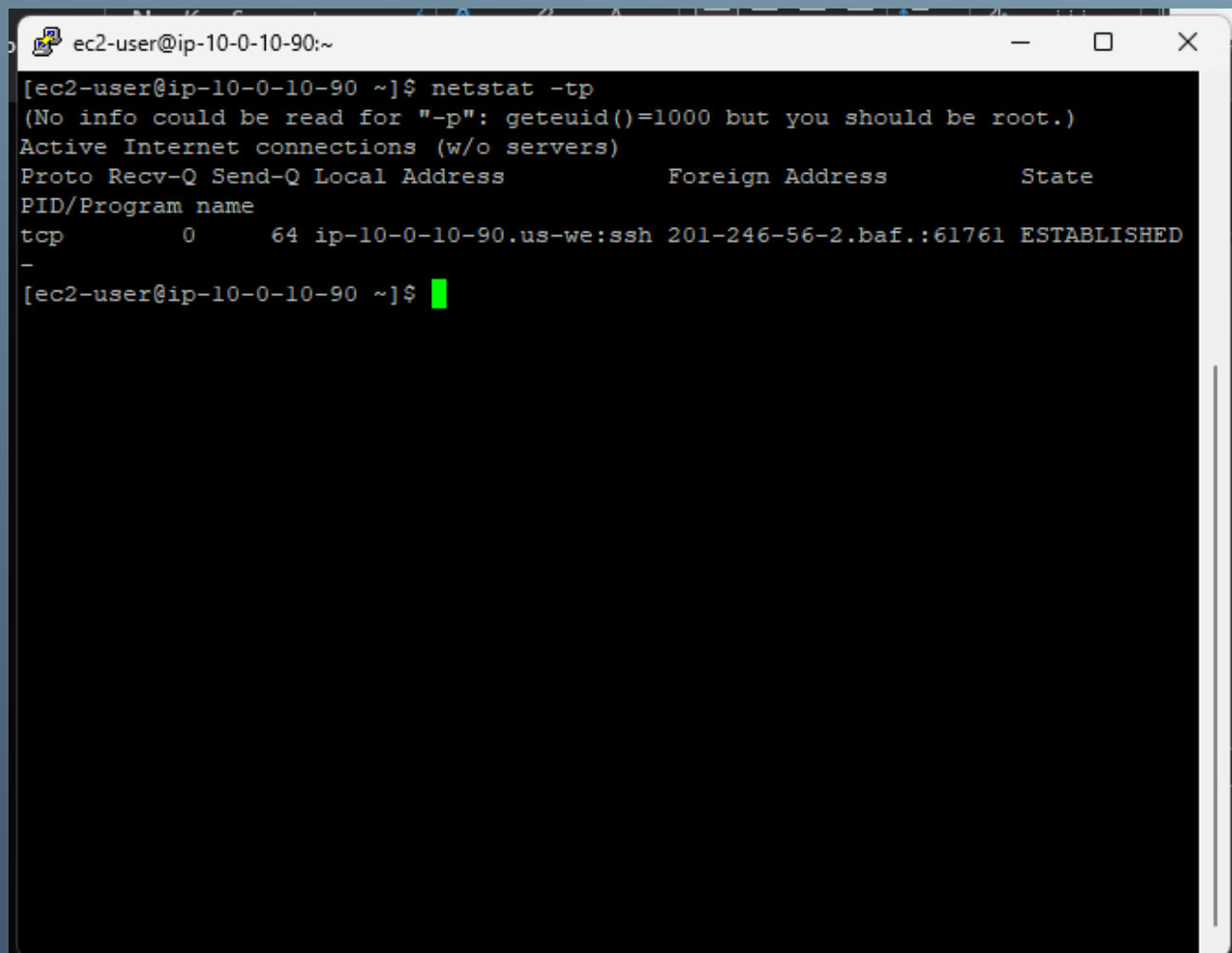
```
ec2-user@ip-10-0-10-90:~  
[ec2-user@ip-10-0-10-90 ~]$ traceroute 8.8.8.8  
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets  
 1  ec2-44-233-118-93.us-west-2.compute.amazonaws.com (44.233.118.93)  8.249 ms  
ec2-44-233-118-99.us-west-2.compute.amazonaws.com (44.233.118.99)  55.696 ms ec2-  
-44-233-118-83.us-west-2.compute.amazonaws.com (44.233.118.83)  1.475 ms  
 2  240.0.208.6 (240.0.208.6)  0.234 ms 240.0.208.7 (240.0.208.7)  0.235 ms 0.2  
24 ms  
 3  240.1.228.15 (240.1.228.15)  6.144 ms 240.1.228.14 (240.1.228.14)  6.099 ms  
240.1.228.13 (240.1.228.13)  6.105 ms  
 4  * * *  
 5  99.83.117.221 (99.83.117.221)  6.294 ms  6.278 ms  6.338 ms  
 6  * * *  
 7  dns.google (8.8.8.8)  6.222 ms  6.041 ms  6.553 ms  
[ec2-user@ip-10-0-10-90 ~]$
```

3. El siguiente es un ejemplo de una situación de cliente donde puede usar el comando netstat:

La empresa ejecuta un análisis de seguridad de rutina y descubrió que se ha puesto en riesgo uno de los puertos en una determinada subred. Para confirmar, ejecute el comando netstat en un host local en esa subred para confirmar si el puerto escucha cuando no debería hacerlo.

En la terminal de Linux, ejecute el siguiente comando y presione Enter:

netstat -tp



```
ec2-user@ip-10-0-10-90:~  
[ec2-user@ip-10-0-10-90 ~]$ netstat -tp  
(No info could be read for "-p": geteuid()=1000 but you should be root.)  
Active Internet connections (w/o servers)  
Proto Recv-Q Send-Q Local Address           Foreign Address         State  
PID/Program name  
tcp        0      64 ip-10-0-10-90.us-we:ssh 201-246-56-2.baf.:61761 ESTABLISHED  
-  
[ec2-user@ip-10-0-10-90 ~]$
```

4. El siguiente es un ejemplo de una situación de cliente donde puede usar el comando telnet:

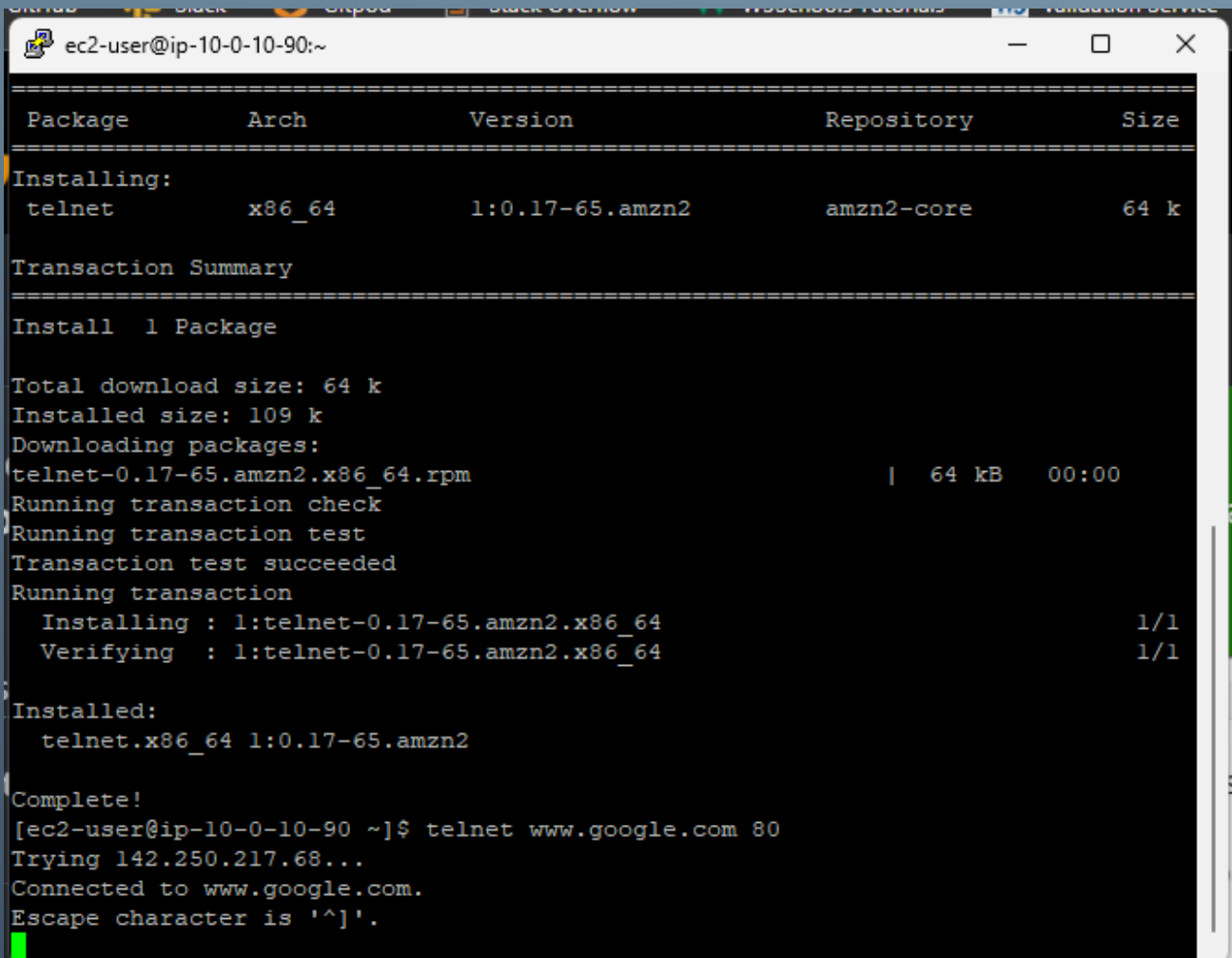
El cliente tiene un servidor web seguro y tiene configuradas reglas de grupo de seguridad personalizadas y reglas de ACL de red. Sin embargo, les preocupa que el puerto 80 esté abierto a pesar de que muestra que su configuración de seguridad indica que su grupo de seguridad bloquee este puerto, puede ejecutar el comando telnet 192.168.10.5 80 para asegurarse de que se rechace la conexión.

En la terminal de Linux, ejecute el siguiente comando y presione Enter para instalar telnet:

```
sudo yum install telnet -y
```

En la terminal de Linux, ejecute el siguiente comando y presione Enter:

```
telnet www.google.com 80
```



```
ec2-user@ip-10-0-10-90:~  
=====
```

Package	Arch	Version	Repository	Size
Installing:				
telnet	x86_64	1:0.17-65.amzn2	amzn2-core	64 k

```
=====
```

Transaction Summary

```
=====
```

Install 1 Package

Total download size: 64 k
Installed size: 109 k
Downloading packages:

Package	Size	Time
telnet-0.17-65.amzn2.x86_64.rpm	64 kB	00:00

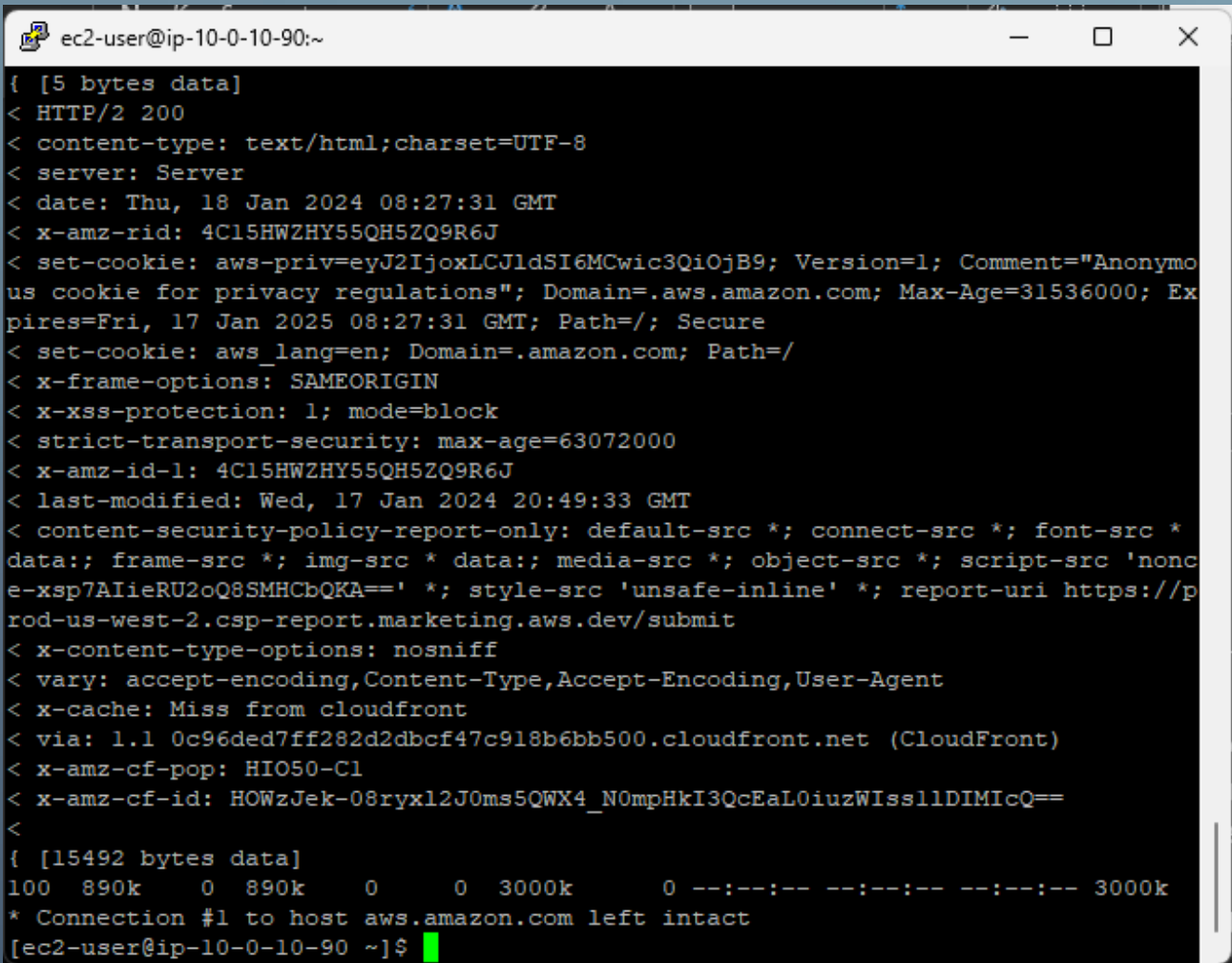
```
Running transaction check  
Running transaction test  
Transaction test succeeded  
Running transaction  
  Installing : 1:telnet-0.17-65.amzn2.x86_64 1/1  
  Verifying  : 1:telnet-0.17-65.amzn2.x86_64 1/1  
Installed:  
telnet.x86_64 1:0.17-65.amzn2  
Complete!  
[ec2-user@ip-10-0-10-90 ~]$ telnet www.google.com 80  
Trying 142.250.217.68...  
Connected to www.google.com.  
Escape character is '^]'.  
[redacted]
```

5. El siguiente es un ejemplo de una situación del cliente en el que puede usar el comando curl:

El cliente tiene un servidor Apache ejecutándose y quiere probar si está recibiendo una solicitud exitosa (200 OK), lo que indica que su sitio web se está ejecutando de manera correcta. Puede ejecutar una solicitud del comando curl para ver si el servidor Apache del cliente devuelve una respuesta 200 OK.

En la terminal de Linux, ejecute el siguiente comando y presione Enter:

```
curl -vLo /dev/null https://aws.com
```



```
ec2-user@ip-10-0-10-90:~  
{ [5 bytes data]  
< HTTP/2 200  
< content-type: text/html; charset=UTF-8  
< server: Server  
< date: Thu, 18 Jan 2024 08:27:31 GMT  
< x-amz-rid: 4C15HWZHY55QH5ZQ9R6J  
< set-cookie: aws-priv=eyJ2IjoxLCJldSI6MCwic3QiOjB9; Version=1; Comment="Anonymous cookie for privacy regulations"; Domain=.aws.amazon.com; Max-Age=31536000; Expires=Fri, 17 Jan 2025 08:27:31 GMT; Path=/; Secure  
< set-cookie: aws_lang=en; Domain=.amazon.com; Path=/  
< x-frame-options: SAMEORIGIN  
< x-xss-protection: 1; mode=block  
< strict-transport-security: max-age=63072000  
< x-amz-id-1: 4C15HWZHY55QH5ZQ9R6J  
< last-modified: Wed, 17 Jan 2024 20:49:33 GMT  
< content-security-policy-report-only: default-src *; connect-src *; font-src * data:; frame-src *; img-src * data:; media-src *; object-src *; script-src 'nonce-xsp7AIieRU2oQ8SMHCbQKA==' *; style-src 'unsafe-inline' *; report-uri https://prod-us-west-2.csp-report.marketing.aws.dev/submit  
< x-content-type-options: nosniff  
< vary: accept-encoding, Content-Type, Accept-Encoding, User-Agent  
< x-cache: Miss from cloudfront  
< via: 1.1 0c96ded7ff282d2dbcf47c918b6bb500.cloudfront.net (CloudFront)  
< x-amz-cf-pop: HIO50-C1  
< x-amz-cf-id: HOWzJek-08ryxl2J0ms5QWX4_N0mpHkI3QcEaL0iuzWIssl1DIMicQ==  
<  
{ [15492 bytes data]  
100 890k 0 890k 0 0 3000k 0 --:--:-- --:--:-- --:--:-- 3000k  
* Connection #1 to host aws.amazon.com left intact  
[ec2-user@ip-10-0-10-90 ~]$
```