

LAB 265

Comandos de solución de problemas del protocolo de Internet

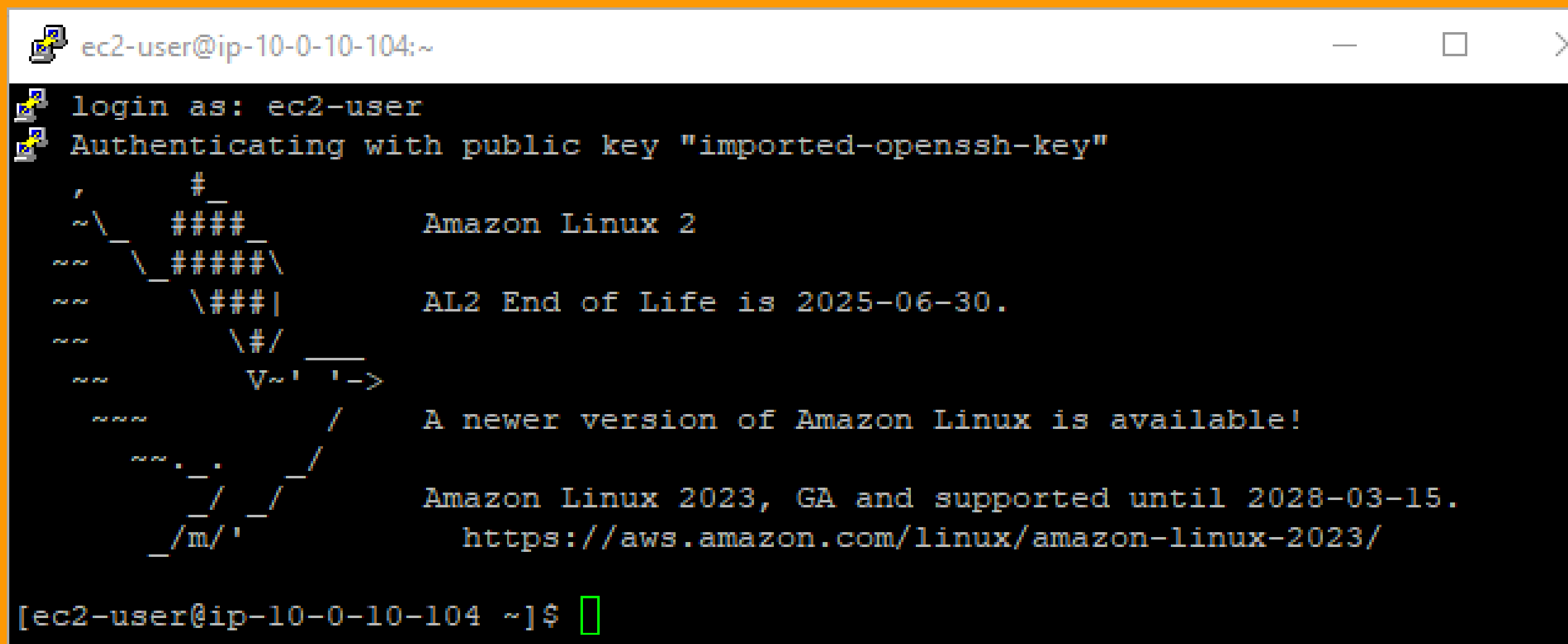
Ramiro Ferreira Da Silva

Objetivos:

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

Situación: "Es un administrador de red nuevo que está solucionando problemas de clientes."

Tarea 1: Conectarse a una instancia EC2 de Amazon Linux mediante SSH.



```
ec2-user@ip-10-0-10-104:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
#  
~\####_ Amazon Linux 2  
~~\#####\  
~~~\####| AL2 End of Life is 2025-06-30.  
~~~~\#/   
~~~~V~' '->  
~~~~/  
~~~~././././ Amazon Linux 2023, GA and supported until 2028-03-15.  
~~~~/m/' https://aws.amazon.com/linux/amazon-linux-2023/  
[ec2-user@ip-10-0-10-104 ~]$
```

Tarea 2: Practicar los comandos de solución de problemas.

Algunas capas tienen comandos relacionados con ellas para ayudar con la resolución de problemas. El siguiente es un ejemplo de cómo fluyen los comandos de solución de problemas con el modelo de interconexión de sistemas abiertos (OSI):

The OSI model and its relation to troubleshooting

	OSI model	Command	Protocol
Layer 7	Application	curl	HTTP/S, SFTP, SSH
Layer 6	Presentation		
Layer 5	Session		
Layer 4	Transport	netstat, ss, telnet	TCP, UDP
Layer 3	Network	ping, traceroute	IP
Layer 2	Data Link		
Layer 1	Physical		

Capa 3 (red):

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

"El cliente ha lanzado una instancia 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."

```
[ec2-user@ip-10-0-10-104 ~]$ 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=40 time=8.75 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=40 time=7.91 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=40 time=7.88 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=40 time=7.88 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=40 time=7.88 ms

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 7.880/8.063/8.750/0.343 ms
```


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 está tardando mucho y que están perdiendo 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.”

```
[ec2-user@ip-10-0-10-104 ~]$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  ec2-34-221-151-97.us-west-2.compute.amazonaws.com (34.221.151.97)  70.116 ms
   ec2-44-233-118-147.us-west-2.compute.amazonaws.com (44.233.118.147)  3.256 ms
   ec2-44-233-117-55.us-west-2.compute.amazonaws.com (44.233.117.55)  1.823 ms
 2  240.3.212.7 (240.3.212.7)  0.332 ms 240.0.120.7 (240.0.120.7)  0.309 ms 100.
   66.8.84 (100.66.8.84)  16.818 ms
 3  242.13.182.69 (242.13.182.69)  0.629 ms 242.2.95.193 (242.2.95.193)  0.482 ms
   242.13.183.197 (242.13.183.197)  0.568 ms
 4  240.1.228.12 (240.1.228.12)  7.901 ms 7.759 ms 240.1.228.13 (240.1.228.13)
   7.079 ms
 5  242.4.194.65 (242.4.194.65)  6.717 ms 242.4.195.65 (242.4.195.65)  9.199 ms
   242.0.55.65 (242.0.55.65)  0.424 ms
 6  108.166.236.21 (108.166.236.21)  0.260 ms * 108.166.236.31 (108.166.236.31)
   0.225 ms
 7  99.83.117.221 (99.83.117.221)  6.462 ms 99.83.116.77 (99.83.116.77)  9.369 ms
   99.83.117.219 (99.83.117.219)  9.548 ms
 8  242.0.55.65 (242.0.55.65)  1.016 ms * 100.95.17.21 (100.95.17.21)  1.673 ms
 9  dns.google (8.8.8.8)  5.771 ms 5.936 ms 5.791 ms
```

Capa 4 (transporte):

El siguiente es un ejemplo de una situación de cliente donde puede usar el comando **netstat**: "Su empresa está ejecutando 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 está listening cuando no debería hacerlo."

```
[ec2-user@ip-10-0-10-104 ~]$ 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      0 ip-10-0-10-104.us-w:ssh 112.173.196.53:62854    TIME_WAIT
-
tcp        0  336 ip-10-0-10-104.us-w:ssh r179-29-48-187.di:13412 ESTABLISHED
-
```

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 está bloqueando este puerto, puede ejecutar el comando `telnet 192.168.10.5 80` para asegurarse de que se rechace la conexión.”

```
ec2-user@ip-10-0-10-104:~  
=====
```

Installing:	Package	Architecture	Version	Source	Size
telnet	x86_64	1:0.17-65.amzn2	amzn2-core	64	

```
=====
```

Transaction Summary	
Install	1 Package

```
=====
```

Total download size: 64 k
Installed size: 109 k
Downloading packages:
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
Verifying : 1:telnet-0.17-65.amzn2.x86_64
Installed:
telnet.x86_64 1:0.17-65.amzn2
Complete!

Primero se instala telnet para luego ejecutar el comando.

```
[ec2-user@ip-10-0-10-104 ~]$ telnet www.google.com 80  
Trying 142.250.217.100...  
Connected to www.google.com.  
Escape character is '^]'.  
[
```


Capa 7 (aplicación):

8

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 correctamente. Puede ejecutar una solicitud del comando curl para ver si el servidor Apache del cliente devuelve una respuesta 200 OK.

```
[ec2-user@ip-10-0-10-104 ~]$ curl -vLO /dev/null https://aws.
% Total    % Received % Xferd  Average Speed   Time    Time
             Dload  Upload  Total   Spent    0
0      0     0     0     0     0     0     0  --:--:--  --:--:
Trying 3.163.24.115:443...
* Connected to aws.com (3.163.24.115) port 443
* ALPN: curl offers h2,http/1.1
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!I
} [5 bytes data]
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
} [512 bytes data]
* CAfile: /etc/pki/tls/certs/ca-bundle.crt
* CAPath: none
{ [5 bytes data]
* TLSv1.2 (IN), TLS handshake, Server hello (2):
{ [100 bytes data]
* TLSv1.2 (IN), TLS handshake, Certificate (11):
{ [4944 bytes data]
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
{ [333 bytes data]
* TLSv1.2 (IN), TLS handshake, Server finished (14):
```

```
pires=Sun, 10-Nov-2024 15:47:28 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: RTP7EFC5K5WDDS6Z37RR
< last-modified: Thu, 09 Nov 2023 12:57:06 GMT
< content-security-policy-report-only: default-src *; connect
data:; frame-src *; img-src * data:; media-src *; object-src
e-xsp7AIieRU2oQ8SMHCbQKA==' *; style-src 'unsafe-inline' *; r
rod-us-west-2.csp-report.marketing.aws.dev/submit
< x-content-type-options: nosniff
< vary: accept-encoding,Content-Type,Accept-Encoding,User-Age
< x-cache: Miss from cloudfront
< via: 1.1 8502ceae0080b3523f89d1a518a99726.cloudfront.net (C
< x-amz-cf-pop: HIO52-P2
< x-amz-cf-id: nKb72_yZOcWLH9jgTTythXWfnLHGbdtpuh-gyLJ41Lmyx-
<
{ [15491 bytes data]
100 285k    0 285k    0    0 1853k    0 --:--:--  --:--:
* Connection #1 to host aws.amazon.com left intact
```


Fin del laboratorio