

PRÁCTICAS DE LABORATORIO

Guillermina Antonaccio

Vigésimo primer laboratorio (265):

Comandos de solución de problemas de protocolo de internet



Instructor:
Roberto Landa

TAREA 1:



En esta tarea, debíamos utilizar SSH para conectarnos a una instancia EC2 de Amazon Linux. Dentro de la misma, debíamos escribir comandos que se utilizan para solucionar problemas.

Los primeros que usamos fueron los comandos ping y tracerout que pertenece a la capa 3 de OSI.

Ping: Su función principal es verificar la conectividad entre un dispositivo (generalmente tu computadora) y otro dispositivo o servidor en una red, ya sea en tu red local o a través de Internet.

Traceroute: sirve para investigar problemas de latencia e identificar si el problema lo tenemos nosotros (en el servidor por nuestro ISP: el proveedor de servicios de internet que utilizamos). Si la pérdida de paquetes se produce hacia el final de la ruta, lo más probable es que el problema esté en la conexión del servidor. Puede identificar un problema o error cuando los nombres de host y las direcciones IP están a ambos lados de un salto fallido, que parece tres asteriscos (***)

Aquí podemos ver el comando ping y el comando traceroute:

```
[ec2-user@ip-10-0-10-139 ~]$ 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=97 time=6.45 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=97 time=6.55 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=97 time=6.41 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=97 time=6.45 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=97 time=6.48 ms

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 6.418/6.472/6.555/0.099 ms
```

```
[ec2-user@ip-10-0-10-139 ~]$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  ec2-50-112-0-70.us-west-2.compute.amazonaws.com (50.112.0.70)  21.799 ms ec2-44-233-117-81.us-west-2.compute.amazonaws.com (44.233.117.81)  7.666 ms ec2-44-233-118-117.us-west-2.compute.amazonaws.com (44.233.118.117)  8.122 ms
 2  100.66.21.218 (100.66.21.218)  13.295 ms 240.3.216.4 (240.3.216.4)  0.246 ms 108.166.240.37 (108.166.240.37)  0.316 ms
 3  240.0.68.16 (240.0.68.16)  0.201 ms 240.0.208.26 (240.0.208.26)  0.215 ms 240.0.208.29 (240.0.208.29)  0.192 ms
 4  108.166.232.52 (108.166.232.52)  0.273 ms 108.166.232.44 (108.166.232.44)  0.339 ms 240.3.216.22 (240.3.216.22)  0.331 ms
 5  108.166.232.60 (108.166.232.60)  0.320 ms 108.166.232.61 (108.166.232.61)  0.309 ms 108.166.232.66 (108.166.232.66)  0.303 ms
 6  108.166.232.94 (108.166.232.94)  0.292 ms 108.166.232.83 (108.166.232.83)  0.350 ms 108.166.240.27 (108.166.240.27)  0.330 ms
 7  242.0.60.1 (242.0.60.1)  1.193 ms 242.0.60.17 (242.0.60.17)  0.441 ms 100.100.20.94 (100.100.20.94)  0.529 ms
 8  100.95.1.129 (100.95.1.129)  1.424 ms 100.92.85.212 (100.92.85.212)  8.386 ms 242.0.31.1 (242.0.31.1)  21.645 ms
 9  100.100.20.80 (100.100.20.80)  0.414 ms 100.95.1.149 (100.95.1.149)  1.330 ms 100.95.17.145 (100.95.17.145)  0.800 ms
10  100.92.37.142 (100.92.37.142)  8.499 ms 100.92.85.206 (100.92.85.206)  8.858 ms 100.92.31.34 (100.92.31.34)  11.111 ms
11  100.92.29.116 (100.92.29.116)  9.832 ms 100.92.35.94 (100.92.35.94)  8.104 ms 100.92.84.126 (100.92.84.126)  8.232 ms
12  100.92.32.19 (100.92.32.19)  8.395 ms 100.92.28.9 (100.92.28.9)  7.030 ms 100.92.84.43 (100.92.84.43)  8.083 ms
```

TAREA 1:



Para la capa 4 (transporte) de OSI utilizamos los comandos netstat y telnet.

Netstat: muestra las conexiones TCP establecidas actualmente desde las cuales el host está escuchando. Al solucionar problemas de red comenzando con la máquina host y avanzando hacia afuera, puede ejecutar este comando para comprender qué puertos están escuchando y cuáles no.

Telnet: se utiliza para establecer una conexión de red a un servidor remoto a través del protocolo Telnet. Por ejemplo podemos confirmar la conexión TCP a un servidor web realizando una solicitud HTTP si se utiliza el puerto 80 para telnet. Si puede conectarse exitosamente al servidor web, entonces no hay nada que impida que usted o el servidor se conecten. Si la conexión falla y aparece un mensaje como "conexión rechazada", es probable que algo esté bloqueando la conexión, como un firewall o un grupo de seguridad

Aquí podemos ver el comando netstat y el comando telnet:

```
[ec2-user@ip-10-0-10-139 ~]$ 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      128 ip-10-0-10-139.us-w:ssh r186-49-55-242.di:62472 ESTABLISHED
-
```

```
[ec2-user@ip-10-0-10-139 ~]$ telnet www.google.com 80
Trying 142.251.211.228...
Connected to www.google.com.
Escape character is '^]'.

```

TAREA 1:




Por ultimo para la capa 7 (aplicación) utilizamos el comando curl.

Este comando nos indica si nuestro servidor está ejecutándose exitosamente. Si nos devuelve un 200 OK entonces el servidor esta funcionando.

Puede utilizar el comando curl para transferir datos entre usted y el servidor. El comando curl puede utilizar muchos protocolos diferentes, pero los más comunes son HTTP y HTTPS. Puede utilizar el comando curl para solucionar problemas de comunicación desde su dispositivo local a un servidor.

Aquí podemos ver el comando curl:

```
{ [5 bytes data]
< HTTP/2 200
< content-type: text/html; charset=UTF-8
< server: Server
< date: Mon, 06 Nov 2023 16:31:31 GMT
< x-amz-rid: F7S2PMTBR2NP9DKJDPTG
< set-cookie: aws-priv=eyJ2IjoxLCJldSI6MCwic3QiOjB9; Version=1; Comment="Anonymous cookie for privacy regulations"; Domain=.aws.amazon.com; Max-Age=31536000; Expires=Tue, 05-Nov-2024 16:31:30 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: F7S2PMTBR2NP9DKJDPTG
< last-modified: Wed, 01 Nov 2023 14:31:54 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 049ca50de603d43d8c9d0f7716efb414.cloudfront.net (CloudFront)
< x-amz-cf-pop: HIO52-P2
< x-amz-cf-id: hQGd2hEC4IV5NRbKaHns2wHuzBnGnvQamDvizAsprNlV3NkCLZerw==
<
{ [15489 bytes data]
100 282k 0 282k 0 0 948k 0 --:--:-- --:--:-- --:--:-- 948k
* Connection #1 to host aws.amazon.com left intact
```



Aquí termina el
laboratorio, muchas
gracias