

Solución a problemas de una red

Lab 266

Participantes:

Luz basso, Florencia Cáceres, Facundo Morales,
Tatiana Rosa, Nacho Suarez y Fernanda Urman.

OBJETIVOS

- Practicar comandos de solución de problemas.
- Identificar como usar esos comandos en el entorno del cliente.

Practicar los comandos de solución de problemas

Comando ping: muestra la conectividad IP a un servidor web.

```
ec2-user@ip-10-0-10-243 ~]$ ping 8.8.8.8 -c 5
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
 4 bytes from 8.8.8.8: icmp_seq=1 ttl=58 time=8.08 ms
 4 bytes from 8.8.8.8: icmp_seq=2 ttl=58 time=8.17 ms
 4 bytes from 8.8.8.8: icmp_seq=3 ttl=58 time=8.11 ms
 4 bytes from 8.8.8.8: icmp_seq=4 ttl=58 time=8.19 ms
 4 bytes from 8.8.8.8: icmp_seq=5 ttl=58 time=8.14 ms

--- 8.8.8.8 ping statistics ---
 5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 8.087/8.142/8.190/0.068 ms
ec2-user@ip-10-0-10-243 ~]$
```

Comando Traceroute: muestra la ruta tomada al servidor web y la latencia que se le aplica.

```
[ec2-user@ip-10-0-10-243 ~]$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  244.5.2.201 (244.5.2.201)  4.329 ms 244.5.2.179 (244.5.2.179)  4.743 ms 244.5.2.211 (244.5.2.211)  4.312 ms
 2  240.3.216.37 (240.3.216.37)  0.343 ms 240.3.216.36 (240.3.216.36)  0.320 ms 240.3.216.4 (240.3.216.4)  0.321 ms
 3  240.1.228.12 (240.1.228.12)  6.870 ms 240.1.228.15 (240.1.228.15)  8.239 ms 10.069 ms
 4  99.83.116.84 (99.83.116.84)  8.550 ms 99.83.117.222 (99.83.117.222)  8.536 ms 99.83.116.84 (99.83.116.84)  8.666 ms
 5  99.83.116.81 (99.83.116.81)  6.683 ms 99.83.116.77 (99.83.116.77)  7.454 ms 99.83.117.219 (99.83.117.219)  7.749 ms
 6  * * *
 7  dns.google (8.8.8.8)  9.707 ms  9.077 ms  8.390 ms
[ec2-user@ip-10-0-10-243 ~]$
```

Comando *netstat*

Muestra las conexiones TCP establecidas actualmente desde las cuales el host está escuchando

netstat -tp: Confirma las conexiones establecidas.

netstat -tlp: Produce los servicios de escucha.

netstat -ntlp: Produce los servicios de escucha pero no resuelve números de puertos.

Brinda conectividad instantánea en capa 4, ayuda a ahorrar tiempo al intentar resolver un problema de red importante.

```
ec2-user@ip-10-0-10-243:~$ netstat -tp
[ec2-user@ip-10-0-10-243 ~]$ 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
tcp        0      0 ip-10-0-10-243.us-w:ssh r190-134-91-48.di:51063 ESTABLISHED -

[ec2-user@ip-10-0-10-243 ~]$ netstat -tlp
(No info could be read for "-p": geteuid()=1000 but you should be root.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN      -
tcp        0      0 localhost:smtp          0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:sunrpc          0.0.0.0:*               LISTEN      -
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN      -
tcp6       0      0 [::]:sunrpc             [::]:*                  LISTEN      -

[ec2-user@ip-10-0-10-243 ~]$ netstat -ntlp
(No info could be read for "-p": geteuid()=1000 but you should be root.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      -
tcp6       0      0 :::22                   :::*                     LISTEN      -
tcp6       0      0 :::111                   :::*                     LISTEN      -

[ec2-user@ip-10-0-10-243 ~]$
```

Comando telnet

telnet: Puede ingresar una IP o URL seguida del número de puerto para conectarse a ese puerto.

Confirma 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.

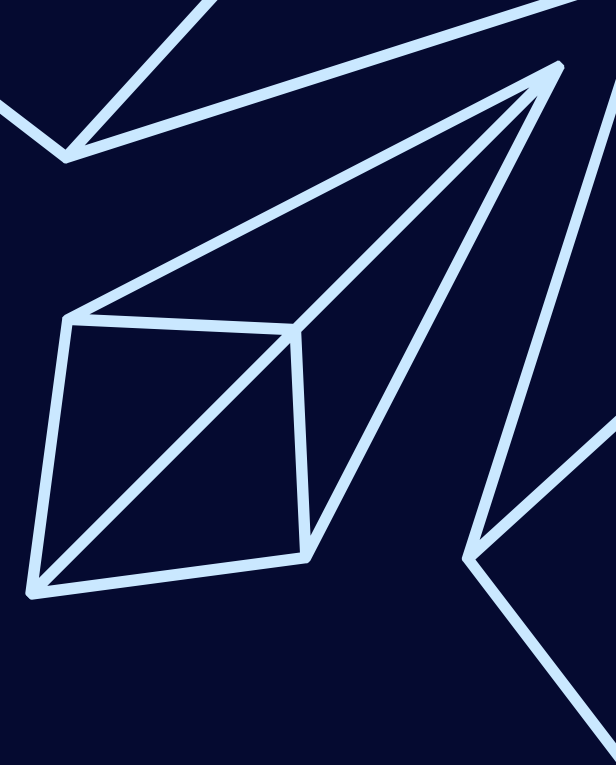
```
telneting to www
[ec2-user@ip-10-0-10-243 ~]$ telnet www.google.com 80
Trying 142.251.33.68...
Connected to www.google.com.
Escape character is '^]'.
█
```

PROBLEMAS

Si la conexión falla...

mensaje "conexión rechazada", es probable que algo esté bloqueando la conexión, como un firewall o un grupo de seguridad.

mensaje "se agotó el tiempo de espera de la conexión", entonces el problema puede ser que no haya ruta de red o conectividad.



COMANDO

Comando curl

Para transferir datos entre usted y el servidor.
Puede utilizar el comando curl para solucionar problemas de comunicación desde su dispositivo local a un servidor.

- I: Esta opción proporciona información del encabezado y especifica que el método de solicitud es Head.
- i: Esta opción especifica que el método de solicitud es GET.
- k: Esta opción le indica al comando que ignore los errores de SSL.
- v: Esta opción es detallada. Muestra qué está haciendo la computadora o qué está cargando el software durante el inicio.
- o: /dev/null: Esta opción enviará HTML y CSS en respuesta a nulo.

```
[ec2-user@ip-10-0-10-243 ~]$ curl -vLo /dev/null https://aws.com
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left     Speed
  0     0    0     0    0     0      0      0  --:--:-- --:--:-- --:--:--     0*   Trying 9
9.84.66.52:443...
* Connected to aws.com (99.84.66.52) port 443
* ALPN: curl offers h2,http/1.1
* Cipher selection: ALL:!EXPORT:!EXPORT40:!EXPORT56:!aNULL:!LOW:!RC4:@STRENGTH
} [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):
{ [4941 bytes data]
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
{ [333 bytes data]
* TLSv1.2 (IN), TLS handshake, Server finished (14):
{ [4 bytes data]
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
} [70 bytes data]
* TLSv1.2 (OUT), TLS change cipher, Change cipher spec (1):
} [1 bytes data]
* TLSv1.2 (OUT), TLS handshake, Finished (20):
} [16 bytes data]
* TLSv1.2 (IN), TLS change cipher, Change cipher spec (1):
{ [1 bytes data]
* TLSv1.2 (IN), TLS handshake, Finished (20):
{ [16 bytes data]
* SSL connection using TLSv1.2 / ECDHE-RSA-AES128-GCM-SHA256
* ALPN: server accepted h2
* Server certificate:
*   subject: CN=aws.com
*   start date: Apr  1 00:00:00 2024 GMT
*   expire date: Apr 29 23:59:59 2025 GMT
*   subjectAltName: host "aws.com" matched cert's "aws.com"
*   issuer: C=US; O=Amazon; CN=Amazon RSA 2048 M02
*   SSL certificate verify ok.
{ [5 bytes data]
* using HTTP/2
* [HTTP/2] [1] OPENED stream for https://aws.com/
```

The background features a dark blue field with a pattern of light blue hexagons. Overlaid on this are several rows of white chevrons pointing downwards and upwards, creating a geometric, crystalline aesthetic.

GRACIAS