



AWS  
re:Start  
LAB

# Troubleshooting Commands



**WEEK 3**





# Overview

---

When troubleshooting network issues, several command-line tools can provide valuable insights into connectivity and network performance. These tools help diagnose problems, identify network delays, and check the status of network connections.

For instance, some tools can simulate communication with remote hosts, revealing response times and potential bottlenecks along the route. Others focus on examining active connections and ports, providing detailed information about network usage and potential issues. These troubleshooting commands are indispensable for network administrators and IT professionals, enabling them to maintain robust and reliable network infrastructures by quickly identifying and resolving connectivity problems.

**Note:** This lab was made using Windows Subsystem for Linux.

## Topics covered

- Practice troubleshooting commands
- Identify how you can use these commands in customer scenarios



# Task 1

## Use SSH to connect to an Amazon Linux EC2 instance

### Initial Preparations

In the AWS Management Console, select the EC2 instance and make note of the **Public IPv4 address**.

Download the private key file **labsuser.pem**. Change to the Downloads directory and modify the permissions on the key to be read-only (r-----).

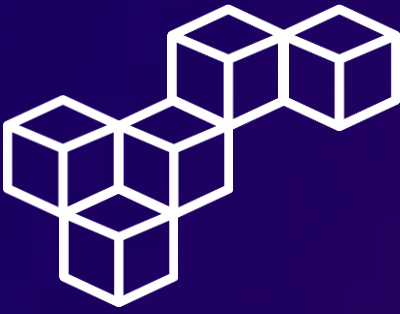
### Connect to the instance using SSH

Establish a connection to the EC2 instance using the `ssh` command, the key and the instance's public IPv4 address.

```
support@HP-Pavilion-Laptop:~/Downloads$ ssh -i labsuser.pem ec2-user@52.32.212.83
The authenticity of host '52.32.212.83 (52.32.212.83)' can't be established.
ED25519 key fingerprint is SHA256:4oMIT9TdtJ2M0qVdRKDMODGqjnf7JId6v4Dd1K7h9aA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '52.32.212.83' (ED25519) to the list of known hosts.

_#_
~\##### 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/
_/_/m/'_/_/_

[ec2-user@ip-10-0-10-205 ~]$
```



## Task 2

---

# Practice troubleshooting commands

### The ping command

In the Linux terminal, run the command `ping 8.8.8.8 -c 5`.

When you run the ping command, you can input an IP or URL followed by options.

You can use the ping command for a few reasons, but the most common reason is to test connectivity to something such as a server.

In this example, the `-c` stands for count, and 5 stands for how many requests you are requesting. The ping command shows IP connectivity to the web server.

```
[ec2-user@ip-10-0-10-205 ~]$ 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=117 time=7.70 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=7.74 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=117 time=7.76 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=117 time=7.72 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=117 time=7.72 ms

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 7.700/7.731/7.765/0.021 ms
[ec2-user@ip-10-0-10-205 ~]$
```



## Task 2

# Practice troubleshooting commands

### The traceroute command

In the Linux terminal, run the command `traceroute 8.8.8.8`.

When you run the traceroute command, you can input an IP or URL followed by options.

The traceroute command reports on the path and latency that a packet takes to get from your machine to the destination.

In this example, the traceroute command shows the path taken to the web server and the latency taken to it.

```
[ec2-user@ip-10-0-10-205 ~]$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  244.5.0.173 (244.5.0.173)  2.815 ms  ec2-50-112-0-4.us-west-2.compute.amazonaws.com (50.112.0.4)  12.189 ms  ec2-44-233-118-175.us-west-2.compute.amazonaws.com (44.233.118.175)  7.902 ms
 2  240.4.228.2 (240.4.228.2)  0.209 ms  240.0.64.4 (240.0.64.4)  0.312 ms  240.4.228.3 (240.4.228.3)  0.212 ms
 3  242.1.38.237 (242.1.38.237)  1.215 ms  242.1.38.109 (242.1.38.109)  1.209 ms  240.1.228.13 (240.1.228.13)  6.312 ms
 4  240.1.228.14 (240.1.228.14)  9.002 ms  241.0.2.69 (241.0.2.69)  0.244 ms  *
 5  99.83.116.77 (99.83.116.77)  8.556 ms  99.83.117.221 (99.83.117.221)  9.153 ms  99.83.116.77 (99.83.116.77)  10.040 ms
 6  99.83.117.219 (99.83.117.219)  9.448 ms  * *
 7  * dns.google (8.8.8.8)  7.722 ms  7.712 ms
[ec2-user@ip-10-0-10-205 ~]$
```



# Task 2

## Practice troubleshooting commands

### The netstat command

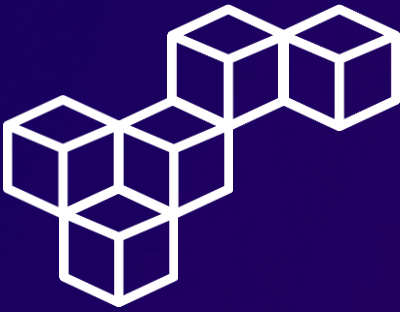
In the Linux terminal, run the command `netstat -tp`.

When you run the netstat command, you can use the following options:

- The command `netstat -tp` confirms established connections.
- The command `netstat -tlp` outputs listening services.
- The command `netstat -ntlp` outputs listening services but does not resolve port numbers.

In this example, the netstat command shows the current established TCP connections from which the host is listening.

```
[ec2-user@ip-10-0-10-205 ~]$ 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-10-0-10-205.us-w:ssh 190.117.58.32:59298     ESTABLISHED -
[ec2-user@ip-10-0-10-205 ~]$
```



## Task 2

---

# Practice troubleshooting commands

### The telnet command

Run the command `sudo yum install telnet -y` to install telnet. Then, run the command `telnet www.google.com 80`.

When you run the telnet command, you can input an IP or URL followed by the port number to connect to that port.

In this example, the telnet command confirms the TCP connection to a web server making an HTTP request using port 80 to telnet.

```
[ec2-user@ip-10-0-10-205 ~]$ telnet www.google.com 80
Trying 142.251.211.228...
Connected to www.google.com.
Escape character is '^['.
```





## Task 2

# Practice troubleshooting commands

## The curl command

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

When you run the curl command, you can use the following options:

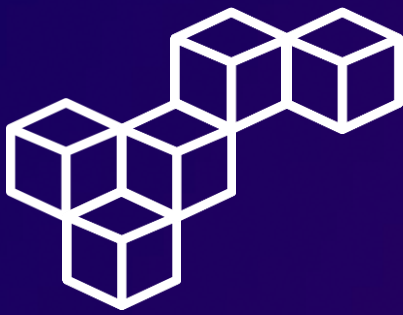
- `-v`: This option is verbose. It shows what the computer is doing or what the software is loading during startup.
- `-I`: This option provides header information and specifies that the request method is Head.
- `-o /dev/null`: This option will send HTML and CSS in response to null.

The curl command can be used to troubleshoot communication and transfer data between your local device and a server.

In the example, the curl command tests the connection to a web service, such as AWS, and submits the HTTP request.

```
[ec2-user@ip-10-0-10-205 ~]$ 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     0  --:--:-- --:--:-- --:--:--     0*   Trying 3.163.24.19:443...
* Connected to aws.com (3.163.24.19) port 443
* ALPN: curl offers h2,http/1.1
```





# Conclusions

---

## **The ping command**

The ping command tests network connectivity by sending ICMP echo requests and receiving ICMP echo replies.

## **The traceroute command**

The traceroute command traces the path that packets take through the network to a destination, showing each hop and its response time.

## **The netstat command**

The netstat command displays network statistics and connections, including listening ports and active connections.

## **The telnet command**

The telnet command establishes a connection to a remote host over a network, often used for testing port availability and connectivity.

## **The curl command**

The curl command retrieves and transfers data to or from a server using various protocols, commonly used for web requests and file transfers.



**Cristhian Becerra**



[cristhian-becerra-espinoza](#)



+51 951 634 354



cristhianbecerra99@gmail.com



Lima, Peru

