



# Testing VPC Connectivity

MA

mallangiharinathreddy727@gmail.com

# Introducing Today's Project!

## What is Amazon VPC?

Amazon VPC is AWS's foundational networking service. Having VPC's is the reason why our resources can made in private or public to the internet! we also setup our own security rules/traffic flow using a VPC.

## How I used Amazon VPC in this project

In today's project we use Amazon VPC to set up a VPC and its components using VPC wizard, and then launched EC2 instances AND tested the connectivity between my network resources.

## One thing I didn't expect in this project was...

we didn't expect for network ACLs Source to point incorrect subnet CIDR block. This was a great reminder of the importance of slowing down and double check configuration settings when using the VPC wizard to automate creating the VPC.

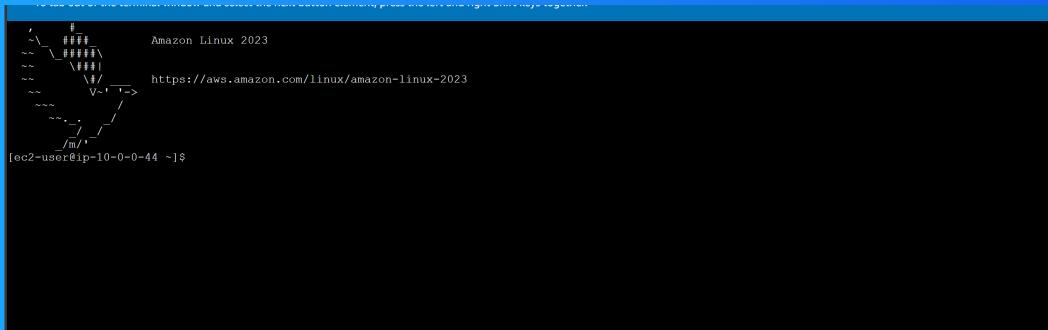
## This project took me...

This project took me 2 Hours and 40 minutes to complete including troubleshooting an unexpected error.

# Connecting to an EC2 Instance

Connectivity means getting resources in our network to communicate each other, and how well they can communicate/deliver data to each other. without connectivity, resources in our network cannot communicate e.g. users cannot access our applications.

My first connectivity test was whether I could connect to my network's public server (an Ec2 instance).



A screenshot of a terminal window on an Amazon Linux 2023 system. The window shows the progress of a file download from the AWS website. The URL is <https://aws.amazon.com/linux/amazon-linux-2023>. The terminal output includes a progress bar consisting of vertical bars and the text "Amazon Linux 2023". The command entered was "curl -O https://aws.amazon.com/linux/amazon-linux-2023". The user is identified as "ec2-user" and the IP address is "10-0-0-44".

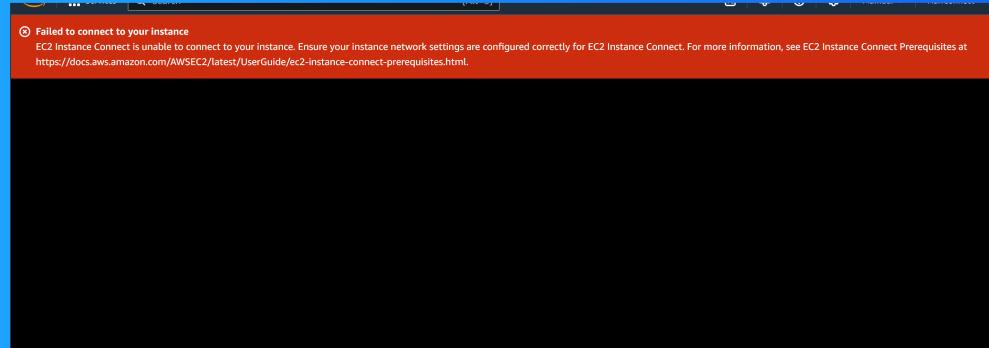
```
curl -O https://aws.amazon.com/linux/amazon-linux-2023
Amazon Linux 2023
[ec2-user@ip-10-0-0-44 ~]$
```

# EC2 Instance Connect

I connected to my EC2 instance using EC2 Instance Connect, which is a tool that allows us to directly access an Ec2 instance using AWS Management console! we no longer need to manage key pairs or use SSH client to connect to our EC2 instances.

My first attempt at getting direct access to my public server resulted in an error, because my private server had a security group that didn't allow SSH traffic- it only allowed HTTP traffic i.e. a different protocol.

I fixed this error by adding a new inbound rule in my private server's security group that allows SSH traffic from anywhere!

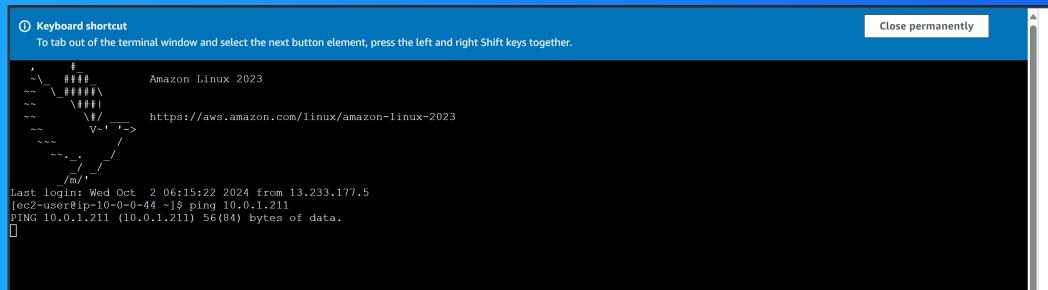


# Connectivity Between Servers

Ping is a tool to test the connectivity between two servers and also the response times (i.e. the performance of the connection). I used ping to test the connectivity between my public and private servers.

The ping command I ran was ' ping 10.0.1.211 ', where 10.0.1.211 is the private IPV4 address of private server.

The first ping returned NO replies from the private server. This meant security settings with my private server was blocking the inbound (and or outbound) ICMP traffic, which is the traffic type of ping messages.



A screenshot of a terminal window titled "Amazon Linux 2023". The window shows a command-line interface with the following text:

```
Keyboard shortcut
To tab out of the terminal window and select the next button element, press the left and right Shift keys together.

Close permanently

Last login: Wed Oct  2 06:15:22 2024 from 13.233.177.5
[ec2-user@ip-10-0-0-44 ~]$ ping 10.0.1.211
PING 10.0.1.211 (10.0.1.211) 56(84) bytes of data.
```



# Troubleshooting Connectivity

I troubleshooted this by enabling ICMP traffic in my private server's network ACLs and security groups! As a bonus, I also made sure the Source I defined correctly pointed to my public subnet.

```
-- \\\|/  
-- \|/  
-- V- ->  
-- /  
-- /  
-- /m/  
Last login: Wed Oct  2 07:16:28 2024 from 13.233.177.4  
[ec2-user@ip-10-0-0-44 ~]$ ping 10.0.1.211  
PING 10.0.1.211(10.0.1.211) 56(84) bytes of data.  
64 bytes from 10.0.1.211: icmp_seq=1 ttl=127 time=0.472 ms  
64 bytes from 10.0.1.211: icmp_seq=2 ttl=127 time=0.457 ms  
64 bytes from 10.0.1.211: icmp_seq=3 ttl=127 time=0.498 ms  
64 bytes from 10.0.1.211: icmp_seq=4 ttl=127 time=0.450 ms  
64 bytes from 10.0.1.211: icmp_seq=5 ttl=127 time=0.532 ms  
64 bytes from 10.0.1.211: icmp_seq=6 ttl=127 time=0.529 ms  
64 bytes from 10.0.1.211: icmp_seq=7 ttl=127 time=0.474 ms  
64 bytes from 10.0.1.211: icmp_seq=8 ttl=127 time=0.468 ms  
64 bytes from 10.0.1.211: icmp_seq=9 ttl=127 time=0.479 ms  
64 bytes from 10.0.1.211: icmp_seq=10 ttl=127 time=0.439 ms  
64 bytes from 10.0.1.211: icmp_seq=11 ttl=127 time=0.414 ms  
64 bytes from 10.0.1.211: icmp_seq=12 ttl=127 time=0.459 ms  
64 bytes from 10.0.1.211: icmp_seq=13 ttl=127 time=0.507 ms  
64 bytes from 10.0.1.211: icmp_seq=14 ttl=127 time=0.519 ms  
64 bytes from 10.0.1.211: icmp_seq=15 ttl=127 time=0.486 ms  
64 bytes from 10.0.1.211: icmp_seq=16 ttl=127 time=0.440 ms  
64 bytes from 10.0.1.211: icmp_seq=17 ttl=127 time=0.495 ms  
64 bytes from 10.0.1.211: icmp_seq=18 ttl=127 time=0.525 ms  
64 bytes from 10.0.1.211: icmp_seq=19 ttl=127 time=0.457 ms  
64 bytes from 10.0.1.211: icmp_seq=20 ttl=127 time=0.424 ms  
64 bytes from 10.0.1.211: icmp_seq=21 ttl=127 time=0.452 ms  
64 bytes from 10.0.1.211: icmp_seq=22 ttl=127 time=0.421 ms  
64 bytes from 10.0.1.211: icmp_seq=23 ttl=127 time=0.397 ms  
64 bytes from 10.0.1.211: icmp_seq=24 ttl=127 time=0.482 ms  
64 bytes from 10.0.1.211: icmp_seq=25 ttl=127 time=0.418 ms  
64 bytes from 10.0.1.211: icmp_seq=26 ttl=127 time=0.480 ms  
64 bytes from 10.0.1.211: icmp_seq=27 ttl=127 time=0.449 ms
```

# Connectivity to the Internet

Curl is a connectivity tool that tests connectivity from a server to another server and retrieves data from the target server too.

I used curl to test the connectivity between my network's public server with the public internet! This test would only be successful If my Internet gateway, network ACLs, security groups and route tables were set up correctly.

## Ping vs Curl

Ping and curl are different because they return different responses to my public server's terminal - ping responds with a report on the performance of connectivity with my private server, Curl responded with HTML data from another public sever.



# Connectivity to the Internet

I ran the curl command `curl 'https://learn.nextwork.org/projects/aws-host-a-website-on-s3'` which returned the HTML content of Nextwork's first project guide.



NextWork.org

# Everyone should be in a job they love.

Check out nextwork.org for  
more projects

