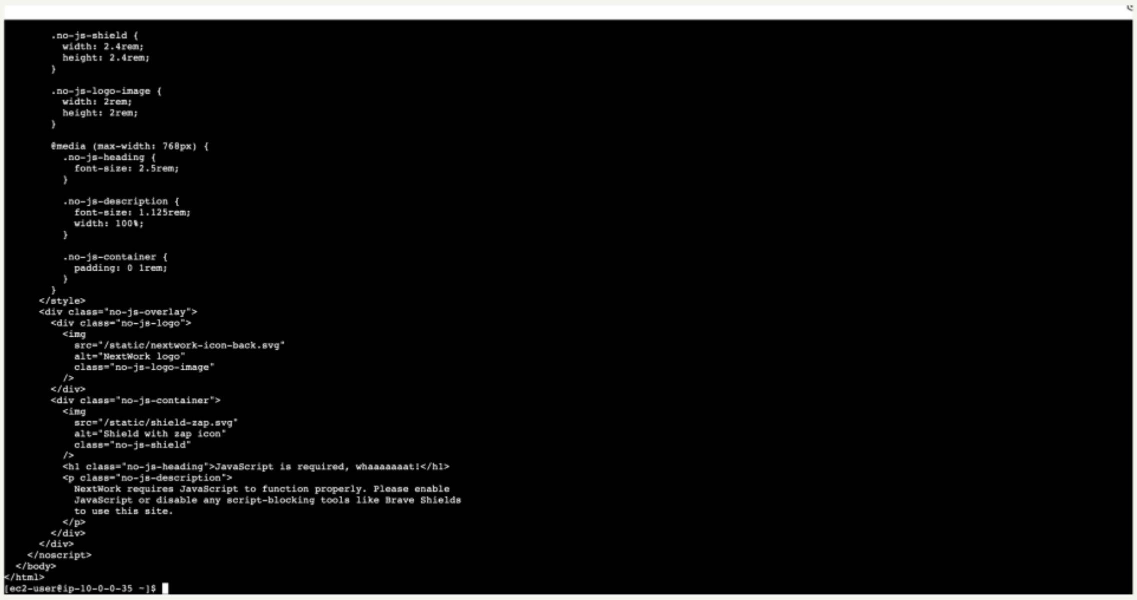
Testing VPC Connectivity



```
        .no-js-shield {
          width: 2.4rem;
          height: 2.4rem;
        }

        .no-js-logo-image {
          width: 2rem;
          height: 2rem;
        }

        @media (max-width: 768px) {
          .no-js-heading {
            font-size: 2.5rem;
          }

          .no-js-description {
            font-size: 1.125rem;
            width: 100%;
          }

          .no-js-container {
            padding: 0 1rem;
          }
        }
      </style>
      <div class="no-js-overlay">
        <div class="no-js-logo">
          <img
            src="/static/nextwork-icon-back.svg"
            alt="NextWork logo"
            class="no-js-logo-image"
          />
        </div>
        <div class="no-js-container">
          <img
            src="/static/shield-zap.svg"
            alt="Shield with zap icon"
            class="no-js-shield"
          />
          <h1 class="no-js-heading">JavaScript is required, whaaaaaaat!</h1>
          <p class="no-js-description">
            NextWork requires JavaScript to function properly. Please enable
            JavaScript or disable any script-blocking tools like Brave Shields
            to use this site.
          </p>
        </div>
      </div>
    </noscript>
  </body>
</html>
[ec2-user@ip-10-0-0-35 ~]$ 
```

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is a service that lets you launch Amazon Web Services (AWS) resources in a logically isolated virtual network that you define. This is useful because it allows users to monitor or control their networks.

How I used Amazon VPC in this project

In today's project, I used Amazon VPC to connect and communicate between two servers.

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

One thing I didn't expect in this project was the curl command spawning HTML data from another public server.

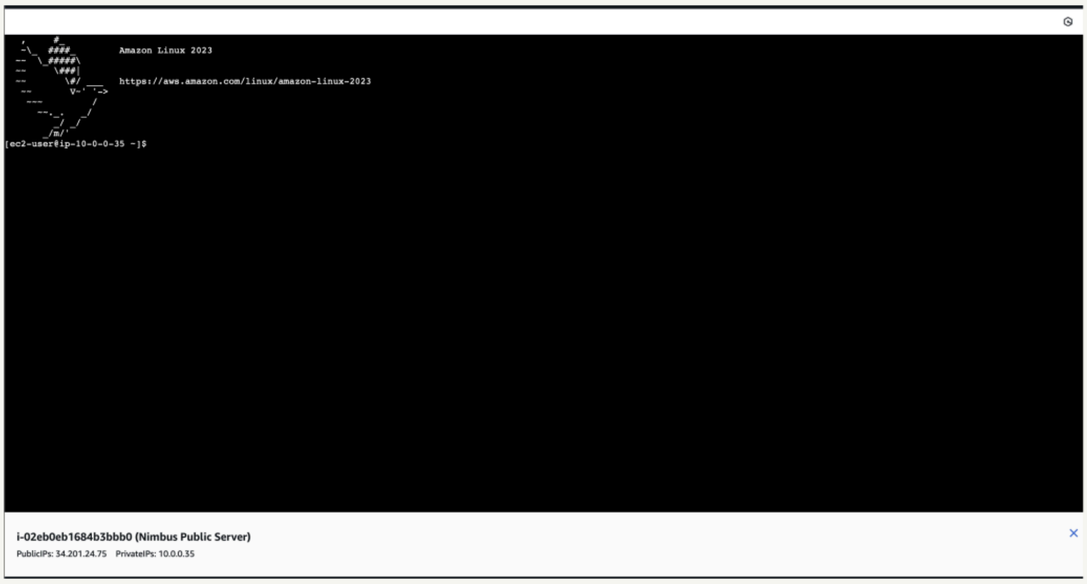
This project took me...

This project took me roughly about an hour.

Connecting to an EC2 Instance

Connectivity means different parts of a network talking to each other and with external networks. It's essential because connectivity is how data flows smoothly across your network, powering everything from simple web hosting on the Internet to complex operations e.g. Netflix using over 100,000 EC2 instances to power its streaming platform.

My first connectivity test was whether I could connect to my public instance. I could not, instead, i got an error



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-0-35 ~]$
```

i-02eb0eb1684b3bbb0 (Nimbus Public Server)

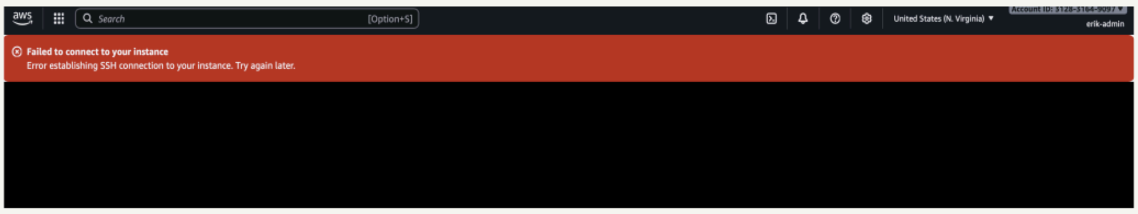
PublicIPs: 34.201.24.75   PrivateIPs: 10.0.0.35

EC2 Instance Connect

I connected to my EC2 instance using EC2 Instance Connect, which is a tool provided by Amazon EC2 which allows us to directly access an EC2 instance using the AWS management console. We no longer need to manage key pairs or use a SSH client to connect.

My first attempt at getting direct access to my public server resulted in an error, because my private server had a security group that did not allow SSH traffic. It only allowed HTTP traffic - 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!



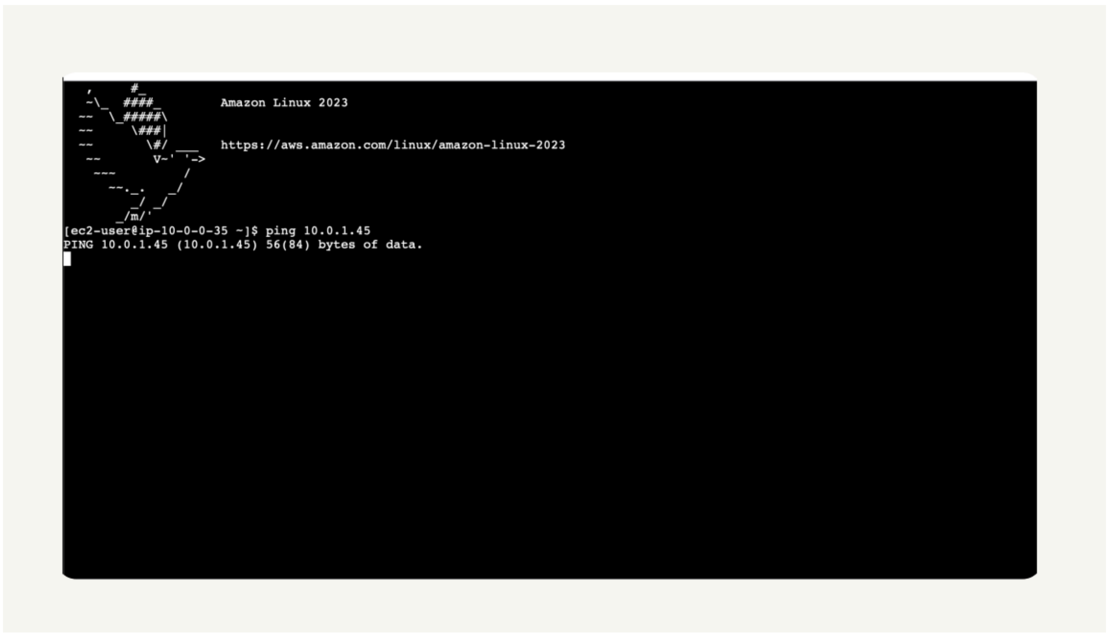
Failed to connect to your instance
Error establishing SSH connection to your instance. Try again later.

Connectivity Between Servers

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

The ping command I ran was 'ping <Private IPv4 Address>'.

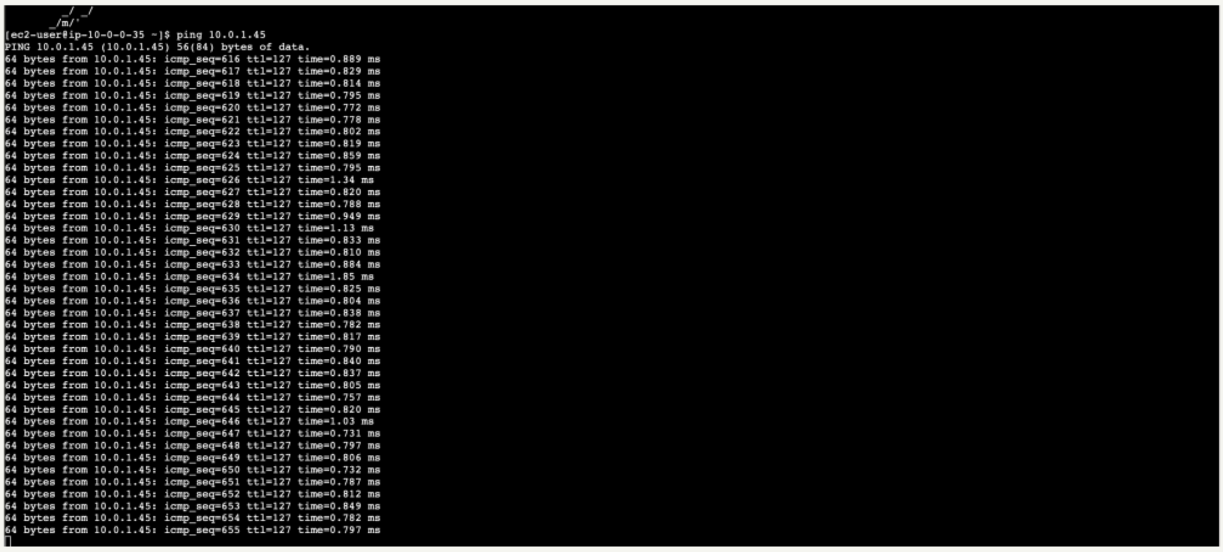
The first ping returned NO replies from the private server. This meant security settings from my private server were blocking inbound and/or outbound ICMP traffic. Which is the traffic type of ping messages



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-0-35 ~]$ ping 10.0.1.45
PING 10.0.1.45 (10.0.1.45) 56(84) bytes of data.
█
```

Troubleshooting Connectivity

I troubleshooted this by enabling ICMP traffic in my private server's network ACL and security groups! Additionally, I also made sure the source I defined in my network ACL pointed to my public subnet.



```
       _/ _/
      _/m/'
[ec2-user@ip-10-0-0-35 ~]$ ping 10.0.1.45
PING 10.0.1.45 (10.0.1.45) 56(84) bytes of data.
64 bytes from 10.0.1.45: icmp_seq=616 ttl=127 time=0.889 ms
64 bytes from 10.0.1.45: icmp_seq=617 ttl=127 time=0.829 ms
64 bytes from 10.0.1.45: icmp_seq=618 ttl=127 time=0.814 ms
64 bytes from 10.0.1.45: icmp_seq=619 ttl=127 time=0.795 ms
64 bytes from 10.0.1.45: icmp_seq=620 ttl=127 time=0.772 ms
64 bytes from 10.0.1.45: icmp_seq=621 ttl=127 time=0.778 ms
64 bytes from 10.0.1.45: icmp_seq=622 ttl=127 time=0.802 ms
64 bytes from 10.0.1.45: icmp_seq=623 ttl=127 time=0.819 ms
64 bytes from 10.0.1.45: icmp_seq=624 ttl=127 time=0.859 ms
64 bytes from 10.0.1.45: icmp_seq=625 ttl=127 time=0.795 ms
64 bytes from 10.0.1.45: icmp_seq=626 ttl=127 time=1.34 ms
64 bytes from 10.0.1.45: icmp_seq=627 ttl=127 time=0.820 ms
64 bytes from 10.0.1.45: icmp_seq=628 ttl=127 time=0.788 ms
64 bytes from 10.0.1.45: icmp_seq=629 ttl=127 time=0.949 ms
64 bytes from 10.0.1.45: icmp_seq=630 ttl=127 time=1.13 ms
64 bytes from 10.0.1.45: icmp_seq=631 ttl=127 time=0.833 ms
64 bytes from 10.0.1.45: icmp_seq=632 ttl=127 time=0.810 ms
64 bytes from 10.0.1.45: icmp_seq=633 ttl=127 time=0.884 ms
64 bytes from 10.0.1.45: icmp_seq=634 ttl=127 time=1.85 ms
64 bytes from 10.0.1.45: icmp_seq=635 ttl=127 time=0.825 ms
64 bytes from 10.0.1.45: icmp_seq=636 ttl=127 time=0.804 ms
64 bytes from 10.0.1.45: icmp_seq=637 ttl=127 time=0.838 ms
64 bytes from 10.0.1.45: icmp_seq=638 ttl=127 time=0.782 ms
64 bytes from 10.0.1.45: icmp_seq=639 ttl=127 time=0.817 ms
64 bytes from 10.0.1.45: icmp_seq=640 ttl=127 time=0.790 ms
64 bytes from 10.0.1.45: icmp_seq=641 ttl=127 time=0.840 ms
64 bytes from 10.0.1.45: icmp_seq=642 ttl=127 time=0.837 ms
64 bytes from 10.0.1.45: icmp_seq=643 ttl=127 time=0.805 ms
64 bytes from 10.0.1.45: icmp_seq=644 ttl=127 time=0.757 ms
64 bytes from 10.0.1.45: icmp_seq=645 ttl=127 time=0.820 ms
64 bytes from 10.0.1.45: icmp_seq=646 ttl=127 time=1.03 ms
64 bytes from 10.0.1.45: icmp_seq=647 ttl=127 time=0.731 ms
64 bytes from 10.0.1.45: icmp_seq=648 ttl=127 time=0.797 ms
64 bytes from 10.0.1.45: icmp_seq=649 ttl=127 time=0.806 ms
64 bytes from 10.0.1.45: icmp_seq=650 ttl=127 time=0.732 ms
64 bytes from 10.0.1.45: icmp_seq=651 ttl=127 time=0.787 ms
64 bytes from 10.0.1.45: icmp_seq=652 ttl=127 time=0.812 ms
64 bytes from 10.0.1.45: icmp_seq=653 ttl=127 time=0.849 ms
64 bytes from 10.0.1.45: icmp_seq=654 ttl=127 time=0.782 ms
64 bytes from 10.0.1.45: icmp_seq=655 ttl=127 time=0.797 ms
```

Connectivity to the Internet

Curl is a tool to test connectivity in a network. Where ping checks if one computer can contact another (and how long messages take to travel back and forth), curl is used to transfer data to or from a server. That means on top of checking connectivity, you can use curl to grab data from, or upload data into other servers on the internet!

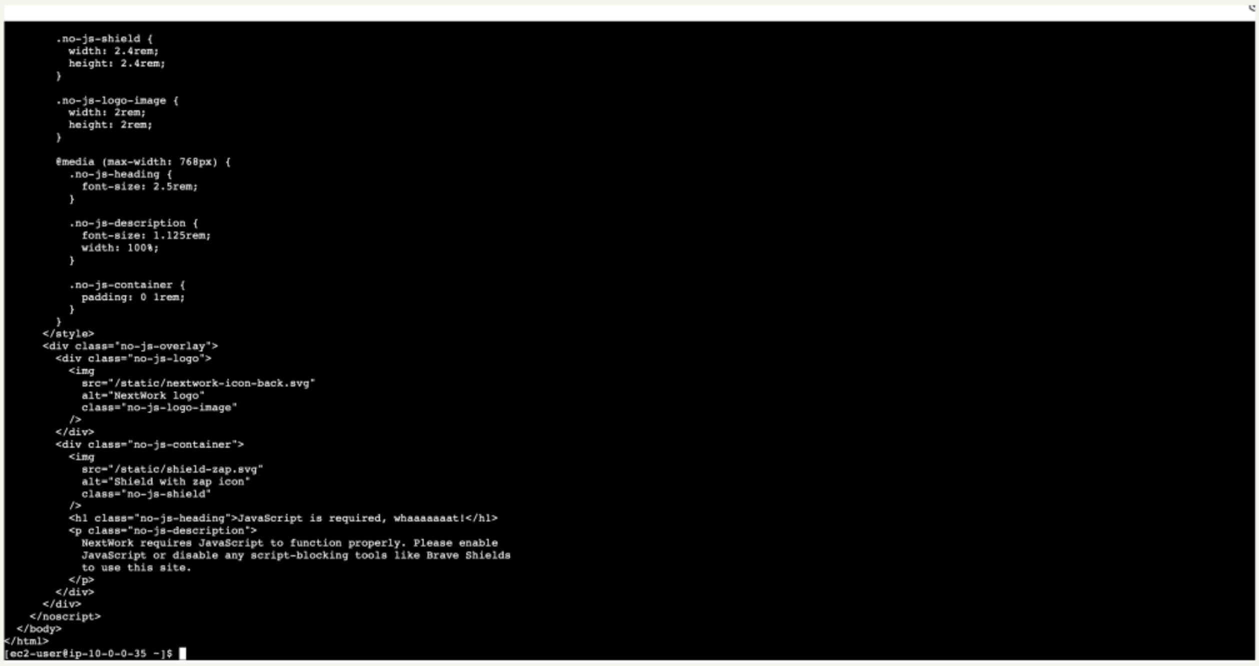
I used curl to test the connectivity between my network's public server with the public internet! This test would only be successful if my internet gateway, network ACLs, security group, and routing tables are correct.

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 with another public server.

Connectivity to the Internet

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



```
        .no-js-shield {
          width: 2.4rem;
          height: 2.4rem;
        }

        .no-js-logo-image {
          width: 2rem;
          height: 2rem;
        }

        @media (max-width: 768px) {
          .no-js-heading {
            font-size: 2.5rem;
          }

          .no-js-description {
            font-size: 1.125rem;
            width: 100%;
          }

          .no-js-container {
            padding: 0 1rem;
          }
        }
      </style>
      <div class="no-js-overlay">
        <div class="no-js-logo">
          <img
            src="/static/nextwork-icon-back.svg"
            alt="NextWork logo"
            class="no-js-logo-image"
          />
        </div>
        <div class="no-js-container">
          <img
            src="/static/shield-zap.svg"
            alt="Shield with zap icon"
            class="no-js-shield"
          />
          <h1 class="no-js-heading">JavaScript is required, whaaaaaaat!</h1>
          <p class="no-js-description">
            NextWork requires JavaScript to function properly. Please enable
            JavaScript or disable any script-blocking tools like Brave Shields
            to use this site.
          </p>
        </div>
      </div>
    </noscript>
  </body>
</html>
[ec2-user@ip-10-0-0-35 ~]$ 
```