



Testing VPC Connectivity

G Gloria

```
[ec2-user@ip-10-0-0-125 ~]$ curl https://learn.nextwork.org/projects/aws-host-a-website-on-s3
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <title>NextWork - Host a Website on Amazon S3</title>
    <meta content="Let's host your very own website on Amazon S3!" name="description" />
    <meta
        content="NextWork - Host a Website on Amazon S3"
        property="og:title"
    />
    <meta
        content="Let's host your very own website on Amazon S3!"
        property="og:description"
    />
    <meta
        content="https://cdn.prod.website-files.com/647460114a0547694b4af16/650b837078ace7d663904eee_nextwork-opengraph-image.png"
        property="og:image"
    />
    <meta
        content="NextWork - Host a Website on Amazon S3"
        property="twitter:title"
    />
    <meta
        content="Let's host your very own website on Amazon S3!"
        property="twitter:description"
    />
    <meta
        content="https://cdn.prod.website-files.com/647460114a0547694b4af16/650b837078ace7d663904eee_nextwork-opengraph-image.png"
        property="twitter:image"
    />
```

i-02654e2ade2847696 (NextWork Public Server)
PublicIPs: 44.222.83.16 PrivateIPs: 10.0.0.125

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC (Virtual Private Cloud) is a secure, isolated network in AWS where you can launch resources like EC2 instances. It's useful because it gives control over networking, such as IP ranges, subnets and security.

How I used Amazon VPC in this project

I used Amazon VPC to test connectivity between a public server and a private server. I had to troubleshoot and configure security groups and NACLs to allow traffic before it succeeded.

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

One thing I didn't expect in this project is how crucial security is in terms of allowing SSH and the use of commands such as ping to test connectivity and curl to test and extract information or data.

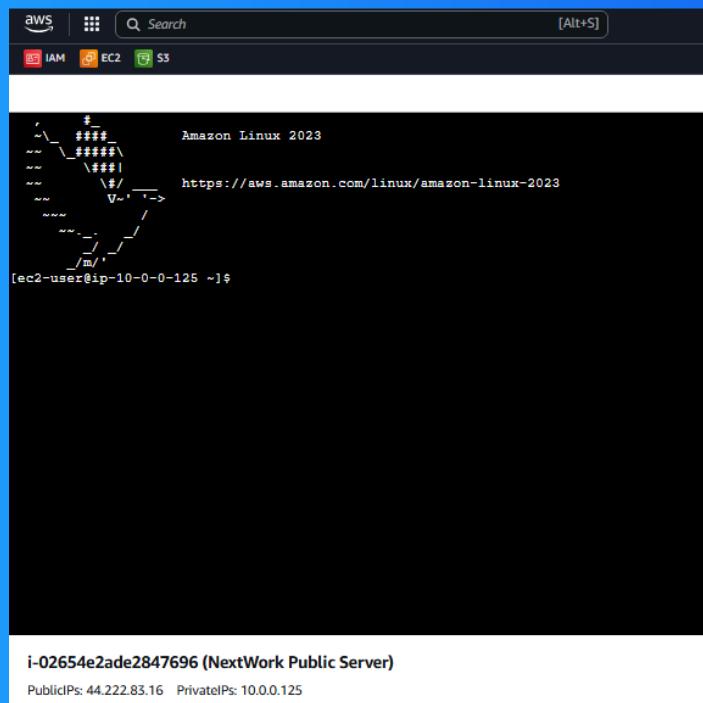
This project took me...

The whole project took 20 minutes and it was easy and quick to do so.

Connecting to an EC2 Instance

Connectivity means the ability of systems, devices, or networks to communicate and exchange data seamlessly. In the context of EC2 instances, it refers to establishing a secure and functional connection between your local device or using AWS connect

My first connectivity test was whether I could connect to the public server, ensuring it was accessible via the internet as configured through its public IP address and security group settings.



A screenshot of an AWS CloudShell terminal window. The terminal shows a successful curl command execution on an Amazon Linux 2023 instance. The output is as follows:

```
aws [Alt+S] IAM EC2 S3
.
.
.
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023
[ec2-user@ip-10-0-0-125 ~]$
```

The terminal window has a dark background and light-colored text. At the bottom, there is a status bar with the instance ID, public IP, and private IP.

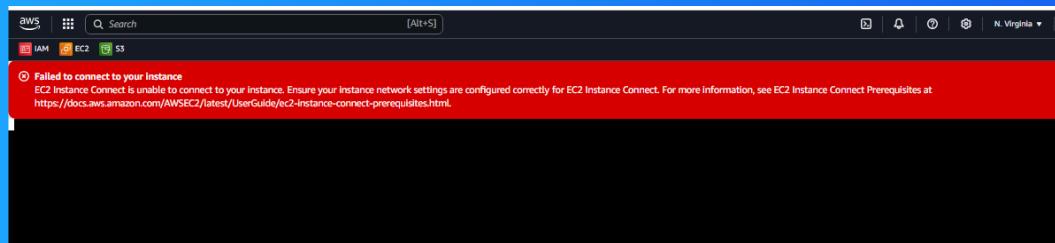
i-02654e2ade2847696 (NextWork Public Server)
PublicIPs: 44.222.83.16 PrivateIPs: 10.0.0.125

EC2 Instance Connect

I connected to my EC2 instance using EC2 Instance Connect, a feature in the AWS Management Console that provides secure SSH access without manually managing key pairs. It automatically generates and manages temporary keys, making the process quick.

My first attempt at getting direct access to my public server resulted in an error because I didn't allow SSH traffic in the security group, preventing the necessary connection for secure remote access.

I fixed this error by updating the security group to allow SSH traffic on port 22 from my anywhere as long as with the keypair, enabling secure and direct access to the public server.

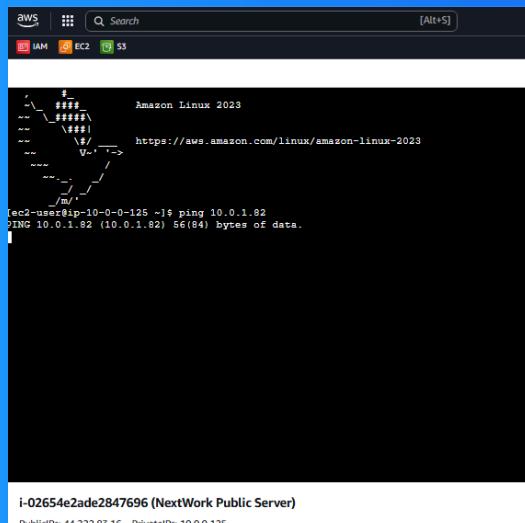


Connectivity Between Servers

Ping is a tool that checks if one device can reach another over a network. I used ping to test the connectivity between my local machine and the EC2 instance to ensure they can communicate successfully.

The ping command I ran was ping [my private IP address], which tested connectivity to my private server within the VPC to ensure the internal network configuration was correctly set up.

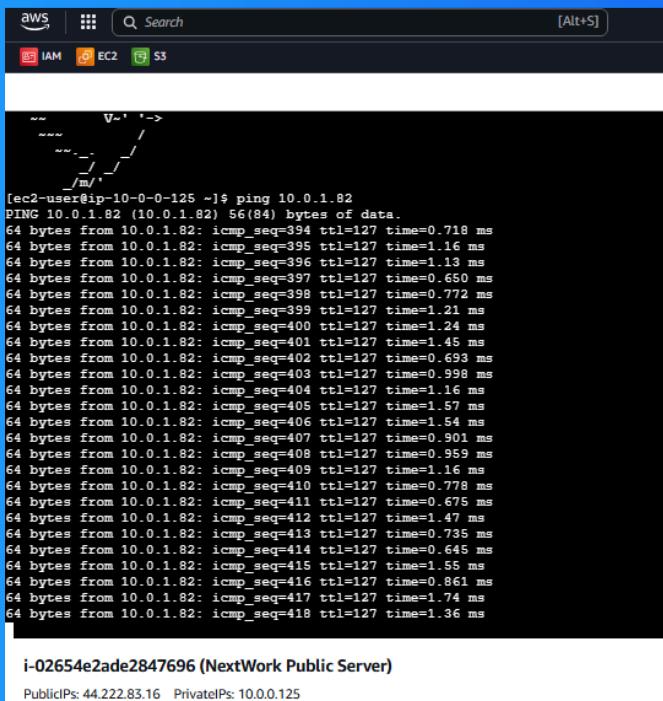
The first ping returned nothing. This meant there was no connectivity between my local machine and the EC2 instance, likely due to a misconfiguration in the security group or network ACL blocking the traffic.



A screenshot of a terminal window on an Amazon Linux 2023 system. The window title is 'Amazon Linux 2023'. The URL 'https://aws.amazon.com/linux/amazon-linux-2023' is visible at the top. The terminal prompt shows the user is in their home directory (~). The command 'ping 10.0.1.82' is run, and the output shows a failed ping attempt: 'PING 10.0.1.82 (10.0.1.82) 56(84) bytes of data.' Below the terminal window, a status bar displays the identifier 'i-02654e2ade2847696 (NextWork Public Server)' and the IP addresses 'PublicIPs: 44.222.83.16 PrivateIPs: 10.0.0.125'.

Troubleshooting Connectivity

I troubleshooted this by updating the NACL's inbound and outbound rules, as well as the security group, to allow all ICMP traffic. This ensured that ping requests could be sent and received, resolving the connectivity issue with the EC2 instance.



The screenshot shows a terminal window within the AWS CloudShell interface. The terminal output displays the results of a ping command from an EC2 instance to another host at 10.0.1.82. The ping traffic is being sent via an interface named 'eth0' with IP 10.0.0.125. The output shows multiple ICMP echo requests (ping) being sent, each with a sequence number (icmp_seq), TTL of 127, and round-trip times ranging from 0.650 ms to 1.74 ms. The terminal also shows the AWS navigation bar at the top with IAM, EC2, and S3 tabs.

```
[ec2-user@ip-10-0-0-125 ~]$ ping 10.0.1.82
PING 10.0.1.82 (10.0.1.82) 56(84) bytes of data.
64 bytes from 10.0.1.82: icmp_seq=394 ttl=127 time=0.718 ms
64 bytes from 10.0.1.82: icmp_seq=395 ttl=127 time=1.16 ms
64 bytes from 10.0.1.82: icmp_seq=396 ttl=127 time=1.13 ms
64 bytes from 10.0.1.82: icmp_seq=397 ttl=127 time=0.650 ms
64 bytes from 10.0.1.82: icmp_seq=398 ttl=127 time=0.772 ms
64 bytes from 10.0.1.82: icmp_seq=399 ttl=127 time=1.21 ms
64 bytes from 10.0.1.82: icmp_seq=400 ttl=127 time=1.24 ms
64 bytes from 10.0.1.82: icmp_seq=401 ttl=127 time=1.45 ms
64 bytes from 10.0.1.82: icmp_seq=402 ttl=127 time=0.693 ms
64 bytes from 10.0.1.82: icmp_seq=403 ttl=127 time=0.998 ms
64 bytes from 10.0.1.82: icmp_seq=404 ttl=127 time=1.16 ms
64 bytes from 10.0.1.82: icmp_seq=405 ttl=127 time=1.57 ms
64 bytes from 10.0.1.82: icmp_seq=406 ttl=127 time=1.54 ms
64 bytes from 10.0.1.82: icmp_seq=407 ttl=127 time=0.901 ms
64 bytes from 10.0.1.82: icmp_seq=408 ttl=127 time=0.959 ms
64 bytes from 10.0.1.82: icmp_seq=409 ttl=127 time=1.16 ms
64 bytes from 10.0.1.82: icmp_seq=410 ttl=127 time=0.778 ms
64 bytes from 10.0.1.82: icmp_seq=411 ttl=127 time=0.675 ms
64 bytes from 10.0.1.82: icmp_seq=412 ttl=127 time=1.47 ms
64 bytes from 10.0.1.82: icmp_seq=413 ttl=127 time=0.735 ms
64 bytes from 10.0.1.82: icmp_seq=414 ttl=127 time=0.645 ms
64 bytes from 10.0.1.82: icmp_seq=415 ttl=127 time=1.55 ms
64 bytes from 10.0.1.82: icmp_seq=416 ttl=127 time=0.861 ms
64 bytes from 10.0.1.82: icmp_seq=417 ttl=127 time=1.74 ms
64 bytes from 10.0.1.82: icmp_seq=418 ttl=127 time=1.36 ms
```

i-02654e2ade2847696 (NextWork Public Server)
PublicIPs: 44.222.83.16 PrivateIPs: 10.0.0.125



Connectivity to the Internet

Curl is a command-line tool used to transfer data to or from a server using various protocols like HTTP, HTTPS, FTP, and more. It is commonly used to test APIs, fetch web content, or interact with servers programmatically.

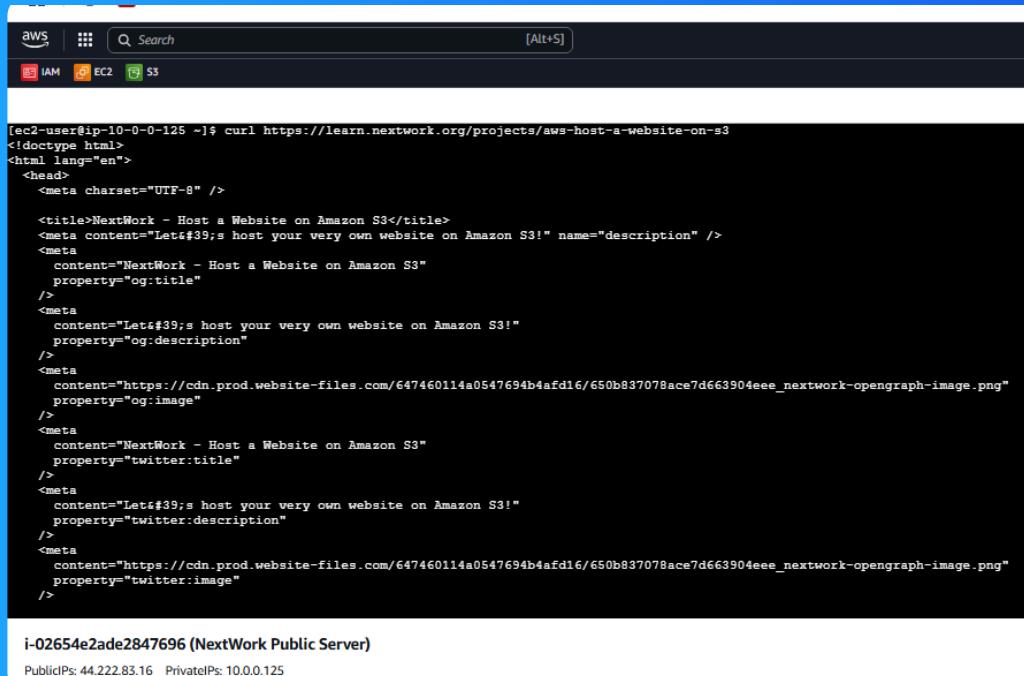
I used curl to test the connectivity between my Public Server and the internet by fetching the HTML content of a website. This confirmed that my server was properly configured to communicate with external servers and retrieve data.

Ping vs Curl

Ping and curl are different because ping checks basic connectivity between devices by sending ICMP packets, while curl tests and retrieves data from a server using application-layer protocols like HTTP, making it useful for web and API interactions.

Connectivity to the Internet

I ran the curl command `curl "https://learn.nextwork.org/projects/aws-host-a-website-on-s3"` and it returned with the complete HTML content of NextWork's web app



A screenshot of a terminal window titled 'aws' showing the output of a curl command. The terminal shows the full HTML code of a web page, which includes meta tags for title, description, and OpenGraph image, along with other standard HTML elements. The terminal interface includes a navigation bar with IAM, EC2, and S3 tabs, and a status bar at the bottom.

```
[ec2-user@ip-10-0-0-125 ~]$ curl https://learn.nextwork.org/projects/aws-host-a-website-on-s3
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>NextWork - Host a Website on Amazon S3</title>
    <meta content="Let's host your very own website on Amazon S3!" name="description" />
    <meta content="NextWork - Host a Website on Amazon S3" property="og:title" />
    <meta content="Let's host your very own website on Amazon S3!" property="og:description" />
    <meta content="https://cdn.prod.website-files.com/647460114a0547694b4af16/650b837078ace7d663904eee_nextwork-opengraph-image.png" property="og:image" />
    <meta content="NextWork - Host a Website on Amazon S3" property="twitter:title" />
    <meta content="Let's host your very own website on Amazon S3!" property="twitter:description" />
    <meta content="https://cdn.prod.website-files.com/647460114a0547694b4af16/650b837078ace7d663904eee_nextwork-opengraph-image.png" property="twitter:image" />
```

i-02654e2ade2847696 (NextWork Public Server)
PublicIPs: 44.222.83.16 PrivateIPs: 10.0.0.125



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

