



nextwork.org

Testing VPC Connectivity



Kehinde Abiuwa

```
aws Search [Option+Space] Europe (Stockholm) kehinde ▾

},
});
</script>

<script>
window.sentryOnLoad = function () {
  window.Sentry.init({
    environment: "prod",
  });
}
</script>
<script
  async
  src="https://js.sentry-cdn.com/149a6bc4cd616ff81bea862cf35e71eb.min.js"
  data-lazy="no"
  crossorigin="anonymous"
></script>

<div id="app"></div>

</body>
</html>
[ec2-user@ip-10-0-0-102 ~]$ █
```

i-0ddfb163a0b8fa8d (NextWork Public Server)
PublicIPs: 13.51.242.26 PrivateIPs: 10.0.0.102

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is a Virtual Private Cloud service provided by AWS that lets you create a logically isolated section of the AWS cloud where you can launch AWS resources, such as EC2 instances, in a customizable virtual network. You control aspects like IP address ranges, subnets, route tables, internet gateways, NAT gateways, and security groups. It is useful because it provides fine-grained control over your network environment, enhances security, supports hybrid cloud architectures (via VPN or Direct Connect), and allows you to isolate workloads, define access rules, and control traffic flow—making it essential for building secure, scalable, and compliant cloud infrastructure

How I used Amazon VPC in this project

i used Amzon vpc in todays project to test connectivity between my private and public ec2 instances and also connectivity between my public ec2 instance and the internet

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

nothing really



K

Kehinde Abiuwa
NextWork Student

nextwork.org

This project took me...

i spent 40minutes on this project

K

Kehinde Abiuwa

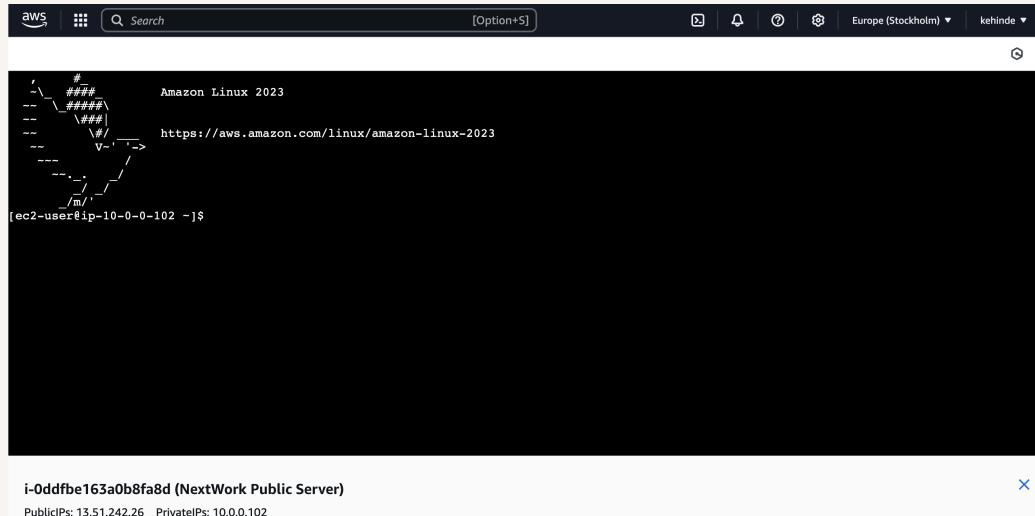
NextWork Student

nextwork.org

Connecting to an EC2 Instance

Connectivity means how well different parts of your network talk 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. Solid connectivity is the backbone of any system that relies on network interactions, making every communication and operation reliable and efficient.

My first connectivity test was whether I could connect to Nextwork public server.

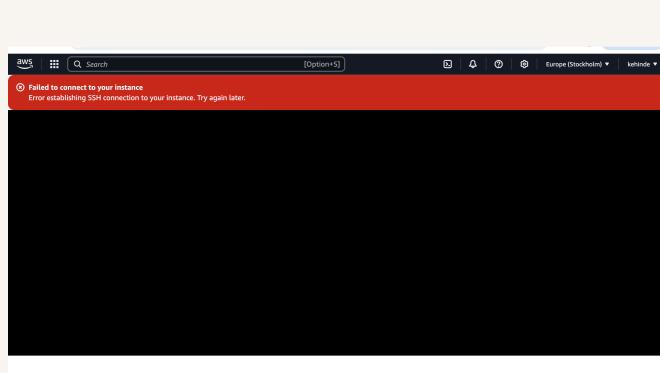


EC2 Instance Connect

I connected to my EC2 instance using EC2 Instance Connect, which is a browser-based SSH client provided by AWS that allows you to securely connect to your Amazon EC2 Linux instances directly from the AWS Management Console without needing to manage SSH keys or install a terminal application. It simplifies the connection process by using temporary, AWS-managed credentials to authenticate you for the session, making it especially useful for quick access or when you don't want to store long-term SSH keys

My first attempt at getting direct access to my public server resulted in an error, because The security group associated with NextWork Public Server lets in all inbound HTTP traffic, but this is not how I am trying to access our Public Server. I am trying to access NextWork Public Server using SSH through EC2 Instance Connect, which is a different traffic type.

I fixed this error by updating NextWork Public Server's security group so it can let in SSH traffic



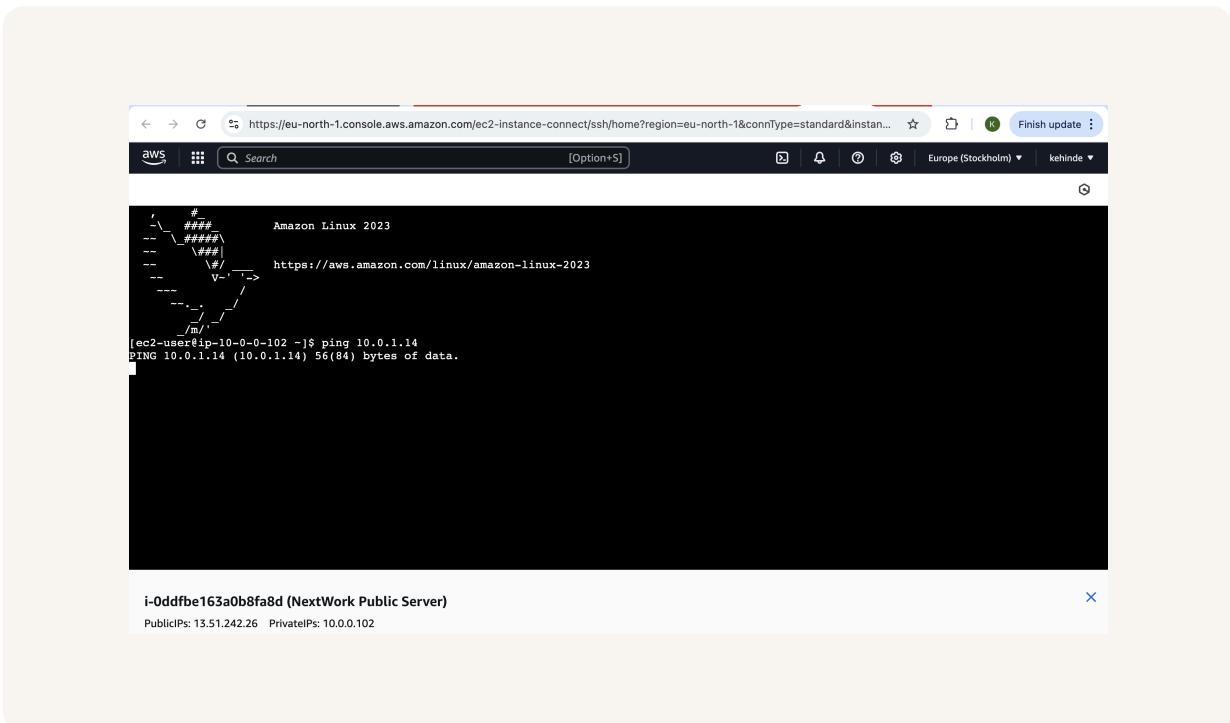
K

Connectivity Between Servers

Ping is a common computer network tool used to check whether your computer can communicate with another computer or device on a network. I used ping to test the connectivity between NextWork Public Server & NextWork Private Server by sending a small packet of data.

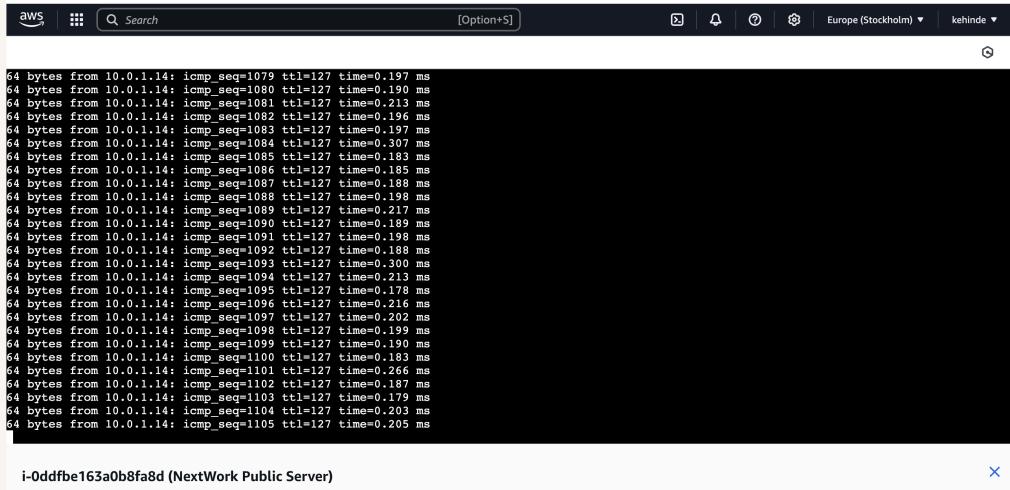
The ping command I ran was : ping 10.0.1.14

The first ping returned no response. This meant that there's a problem with the connection where NextWork Private Server or its network was blocking the type of messages used in ping, which are known as ICMP (Internet Control Message Protocol) traffic.



Troubleshooting Connectivity

I troubleshooted this by updating my route table and security groups to allow ICMP traffic



A screenshot of the AWS CloudWatch Metrics interface. The top navigation bar includes 'aws' (with a smiley face icon), a search bar, and a dropdown for 'Option+S'. On the right, it shows 'Europe (Stockholm)' and a user dropdown for 'kehinde'. Below the header, there's a list of log entries. Each entry consists of a timestamp followed by a log message: '64 bytes from 10.0.1.14: icmp_seq=1079 ttl=127 time=0.197 ms'. The log entries are repeated numerous times, indicating continuous ICMP traffic. At the bottom of the screenshot, there's a footer bar with the text 'i-0ddfbe163a0b8fa8d (NextWork Public Server)' and a small 'X' icon.

```
64 bytes from 10.0.1.14: icmp_seq=1079 ttl=127 time=0.197 ms
64 bytes from 10.0.1.14: icmp_seq=1080 ttl=127 time=0.190 ms
64 bytes from 10.0.1.14: icmp_seq=1081 ttl=127 time=0.213 ms
64 bytes from 10.0.1.14: icmp_seq=1082 ttl=127 time=0.196 ms
64 bytes from 10.0.1.14: icmp_seq=1083 ttl=127 time=0.197 ms
64 bytes from 10.0.1.14: icmp_seq=1084 ttl=127 time=0.307 ms
64 bytes from 10.0.1.14: icmp_seq=1085 ttl=127 time=0.183 ms
64 bytes from 10.0.1.14: icmp_seq=1086 ttl=127 time=0.185 ms
64 bytes from 10.0.1.14: icmp_seq=1087 ttl=127 time=0.188 ms
64 bytes from 10.0.1.14: icmp_seq=1088 ttl=127 time=0.198 ms
64 bytes from 10.0.1.14: icmp_seq=1089 ttl=127 time=0.217 ms
64 bytes from 10.0.1.14: icmp_seq=1090 ttl=127 time=0.189 ms
64 bytes from 10.0.1.14: icmp_seq=1091 ttl=127 time=0.198 ms
64 bytes from 10.0.1.14: icmp_seq=1092 ttl=127 time=0.188 ms
64 bytes from 10.0.1.14: icmp_seq=1093 ttl=127 time=0.300 ms
64 bytes from 10.0.1.14: icmp_seq=1094 ttl=127 time=0.213 ms
64 bytes from 10.0.1.14: icmp_seq=1095 ttl=127 time=0.178 ms
64 bytes from 10.0.1.14: icmp_seq=1096 ttl=127 time=0.216 ms
64 bytes from 10.0.1.14: icmp_seq=1097 ttl=127 time=0.202 ms
64 bytes from 10.0.1.14: icmp_seq=1098 ttl=127 time=0.196 ms
64 bytes from 10.0.1.14: icmp_seq=1099 ttl=127 time=0.150 ms
64 bytes from 10.0.1.14: icmp_seq=1100 ttl=127 time=0.183 ms
64 bytes from 10.0.1.14: icmp_seq=1101 ttl=127 time=0.266 ms
64 bytes from 10.0.1.14: icmp_seq=1102 ttl=127 time=0.187 ms
64 bytes from 10.0.1.14: icmp_seq=1103 ttl=127 time=0.179 ms
64 bytes from 10.0.1.14: icmp_seq=1104 ttl=127 time=0.203 ms
64 bytes from 10.0.1.14: icmp_seq=1105 ttl=127 time=0.205 ms
```

Connectivity to the Internet

curl is a tool to test connectivity in a network. 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

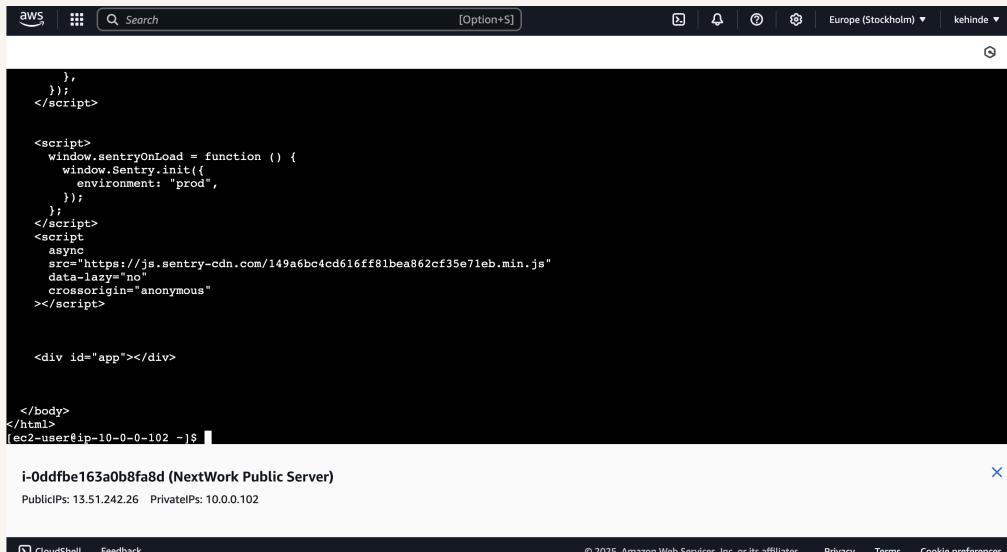
I used curl to test the connectivity between my server & the server that hosts the website (example.com). This request tells the server that I want to retrieve the HTML content of the website.

Ping vs Curl

Ping and curl are different because ping is a network diagnostic tool used to test the reachability and response time between your device and another host by sending ICMP (Internet Control Message Protocol) echo requests, while curl is a command-line tool used to transfer data to or from a server using various protocols such as HTTP, HTTPS, FTP, etc. In short, ping checks if a host is up and how fast it responds, whereas curl interacts with web servers and fetches actual content like webpages or APIs.

Connectivity to the Internet

I ran the curl command 'curl example.com'. which returned the html content of example.com



A screenshot of a terminal window titled 'aws' with a search bar and a status bar indicating 'Europe (Stockholm)' and 'kehinde'. The terminal displays the following command and its output:

```
curl https://example.com
```

```
... (redacted content)
```

```
i-0ddfbe163a0b8fa8d (NextWork Public Server)
Public IPs: 13.51.242.26 Private IPs: 10.0.0.102
```

The terminal window has a light gray background and a dark gray header bar.



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

