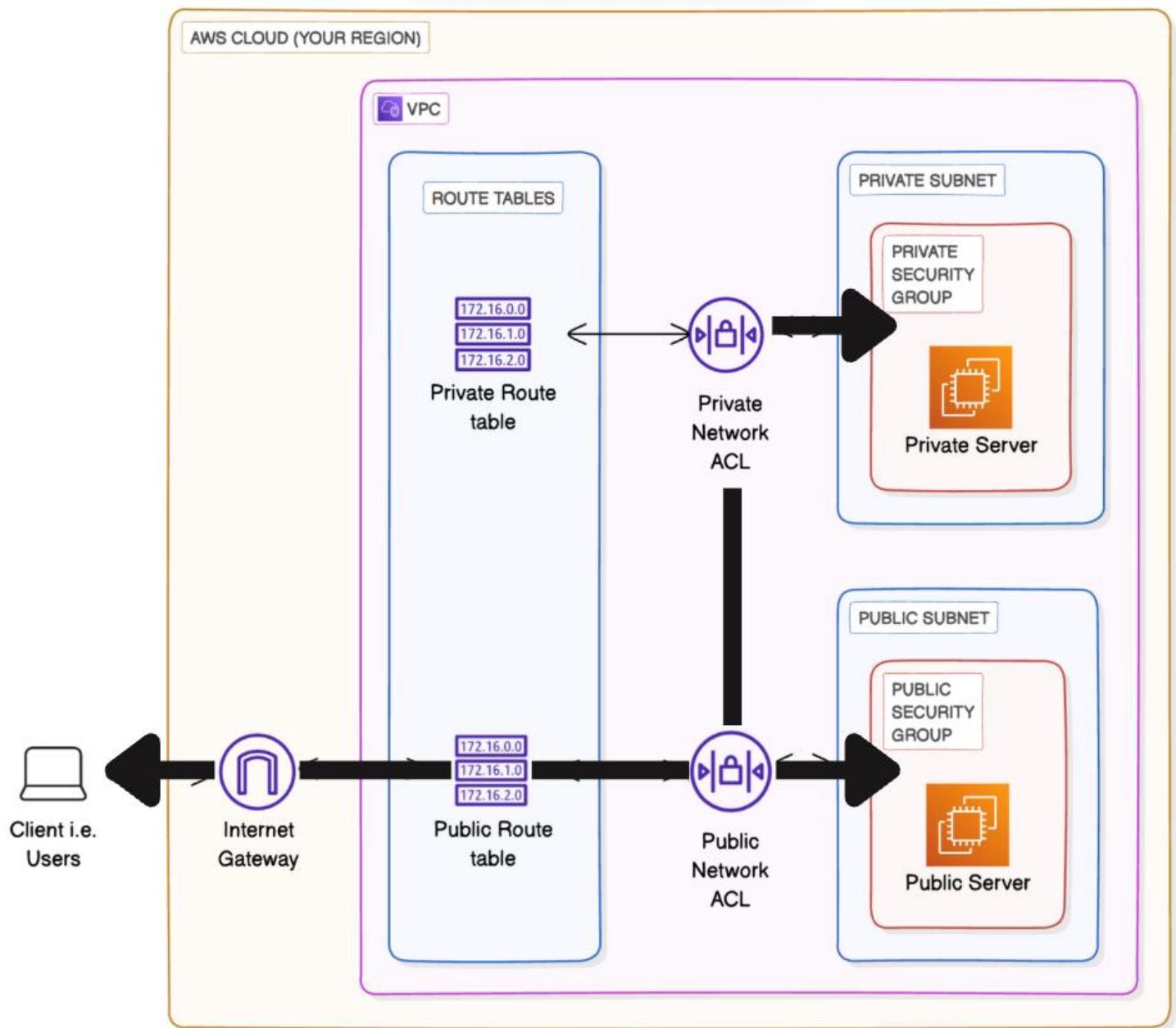


TESTING VPC CONNECTIVITY - STEP-BY-STEP DOCUMENTATION



This documentation is a detailed, step-by-step guide to testing VPC connectivity for your AWS project. It is self-contained and includes a full recap of the prior VPC setup steps, followed by the connectivity tests you'll perform between public and private EC2 instances and to the internet.

Step 0: Before we start...

Make sure you've completed the earlier projects in this VPC networking series or have the following already set up:

- An AWS account and access to the AWS Management Console.
- A VPC (NextWork VPC) with Public and Private subnets.
- An Internet Gateway attached to the VPC.

- Route tables, Security Groups and Network ACLs configured.
- One EC2 instance in the Public Subnet (NextWork Public Server) and one in the Private Subnet (NextWork Private Server).

👉 What are we here to do today? (Quick note for your documentation):

In this project I will verify connectivity between EC2 instances in my VPC and validate internet access from the public subnet.

Step 1: Create a VPC

Set up a new Virtual Private Cloud in AWS.

Actions:

- Log in to the AWS Management Console.
- Search for 'VPC' in the search bar and select it.
- In the left-hand navigation pane, choose 'Your VPCs'.
- Click 'Create VPC'.
- Select 'VPC only'.
- Enter a Name tag (e.g. NextWork VPC).
- Enter an IPv4 CIDR block (e.g., 10.0.0.0/16).
- Click 'Create VPC' to finalize.

VPC dashboard

vpc-04e56d1efd3690204 / NextWork VPC

Details		Actions	
VPC ID	vpc-04e56d1efd3690204	State	Available
DNS resolution	Enabled	Tenancy	default
Main network ACL	acl-07c912d4587db4e8b	Default VPC	No
IPv6 CIDR (Network border group)	-	Network Address Usage metrics	Disabled
Block Public Access	Off	DHCP option set	dopt-019ae45123624041f
IPv4 CIDR	10.0.0.0/16	Route 53 Resolver DNS Firewall rule groups	-
DNS hostnames	Disabled	Main route table	rtb-00a23c4fdc4c8daf4
IPv6 pool	-	Owner ID	944362433406

Resource map

- VPN: Your AWS virtual network (NextWork VPC)
- Subnets (0): Subnets within this VPC
- Route tables (1): Route network traffic to resources (rtb-00a23c4fdc4c8daf4)
- Network Connections (0): Connections to other networks

Step 2: Create Subnets

Create public and private subnets inside your VPC.

Actions:

- From the VPC Dashboard, choose 'Subnets'.
- Click 'Create subnet'.
- Choose the VPC you created (NextWork VPC).
- Enter a Subnet name (e.g., Public 1).
- Select an Availability Zone.
- Enter an IPv4 subnet CIDR block (e.g., 10.0.0.0/24).
- Click 'Create subnet'.
- Select the subnet, choose 'Edit subnet settings', and enable 'Auto-assign public IPv4 address' for the public subnet.
- Repeat to create a private subnet (e.g., 10.0.1.0/24) and leave auto-assign public IPv4 disabled.

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
Private 1	subnet-0bcf91b68f0bf4a8e	Available	vpc-04e56d1efd3690204 Next...	Off	10.0.1.0/24
-	subnet-0543cdd4ce031a9a3	Available	vpc-0dcc2a2993f222607	Off	172.31.0.0/20
-	subnet-03131cfee49ee58a7	Available	vpc-0dcc2a2993f222607	Off	172.31.16.0/20
Public 1	subnet-0b54ba4fc577c321	Available	vpc-04e56d1efd3690204 Next...	Off	10.0.0.0/24

Step 3: Create an Internet Gateway

Attach an Internet Gateway to provide internet access for public subnets.

Actions:

- In the VPC Dashboard, choose 'Internet gateways'.
- Click 'Create internet gateway'.
- Enter a Name tag (e.g., NextWork IG).
- Click 'Create internet gateway'.
- Select the gateway, choose 'Actions' → 'Attach to VPC'.
- Select your VPC (NextWork VPC) and confirm.

Name	Internet gateway ID	State	VPC ID	Owner
-	igw-0869ae9116f7be790	Attached	vpc-0dcc2a2993f222607	944362433406
NextWork IG	igw-026f3dfa162fe764b	Attached	vpc-04e56d1efd3690204 NextWork VPC	944362433406

igw-026f3dfa162fe764b / NextWork IG

Details **Tags**

Details

Internet gateway ID igw-026f3dfa162fe764b	State Attached	VPC ID vpc-04e56d1efd3690204 NextWork VPC	Owner 944362433406
--	-------------------	--	-----------------------

Step 4: Create/Configure Route Tables

Create route tables and associate them with subnets.

Actions:

- Go to 'Route tables' in the VPC Dashboard.
- Create a public route table and associate it with your Public subnet.
- Edit the routes and add Destination = 0.0.0.0/0 with Target = Internet Gateway.
- Create a private route table and associate it with your Private subnet.
- Do not add a route to the Internet Gateway for the private route table.

The screenshot shows the AWS VPC Route Tables page. The left sidebar is collapsed. The main content area displays the 'rtb-00a23c4fdc4c8daf4 / Public Route Table'. The 'Details' tab is selected, showing the route table ID, VPC, and owner ID. Under the 'Routes' tab, there are two routes: one to the internet gateway (igw-026f3dfa162fe764b) and one to the local subnet (10.0.0.0/16).

The screenshot shows the AWS VPC Route Tables page. The left sidebar is collapsed. The main content area displays the 'rtb-0f9b772ab9a96e840 / Private Route Table'. The 'Details' tab is selected, showing the route table ID, VPC, and owner ID. Under the 'Routes' tab, there is one route to the local subnet (10.0.0.0/16).

Step 5: Create Security Groups

Create Security Groups to control instance-level traffic.

Actions:

- In the VPC Dashboard, select 'Security groups'.
- Click 'Create security group'.
- Enter a Name tag (e.g., NextWork Public Security Group) and select your VPC.
- Add inbound rules for HTTP (80) and any required ports (e.g., SSH 22 for management).
- Create a separate Security Group for the Private subnet (e.g., NextWork Private Security Group) with limited inbound rules.

- Attach the appropriate security groups to your EC2 instances.

The screenshot shows the AWS VPC Security Groups page. The selected security group is 'sg-085e0775279eb18d9 - NextWork Public Security Group'. The 'Details' section shows the security group name ('NextWork Public Security Group'), security group ID ('sg-085e0775279eb18d9'), owner ('944362433406'), and descriptions ('A security group for NextWork Public Subnet'). It also shows inbound rules count (3) and outbound rules count (1). Below the details, there are tabs for 'Inbound rules', 'Outbound rules', 'Sharing - new', 'VPC associations - new', and 'Tags'. The 'Inbound rules' tab is selected, displaying three rules:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-0101a1f3fc523c46a	IPv4	HTTP	TCP	80	0.0.0.0/0
-	sgr-03c96894d3d32873c	IPv4	SSH	TCP	22	0.0.0.0/0
-	sgr-06a7105b303cf3ac1	IPv4	HTTPS	TCP	443	0.0.0.0/0

The screenshot shows the AWS VPC Security Groups page. The selected security group is 'sg-043c4824589941a42 - NextWork Private Security Group'. The 'Details' section shows the security group name ('NextWork Private Security Group'), security group ID ('sg-043c4824589941a42'), owner ('944362433406'), and descriptions ('A security Group for NextWork Private Subnet'). It also shows inbound rules count (2) and outbound rules count (1). Below the details, there are tabs for 'Inbound rules', 'Outbound rules', 'Sharing - new', 'VPC associations - new', and 'Tags'. The 'Inbound rules' tab is selected, displaying two rules:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-043d282f4fabf8bcc	IPv4	All ICMP - IPv4	ICMP	All	10.0.0.0/24
-	sgr-0a6acd488e1a89569	IPv4	SSH	TCP	22	10.0.0.0/24

Step 6: Create Network ACLs

Create Network ACLs (NACLs) to control subnet-level traffic.

Actions:

- In the VPC Dashboard, choose 'Network ACLs'.
- Create or select a Network ACL for the Public subnet and review its inbound/outbound rules.
- Rename the default NACL associated with Public subnet (optional) to 'NextWork Public NACL'.
- Create a new Network ACL for the Private subnet (NextWork Private NACL).
- Note: custom NACLs start with all traffic denied; you'll add rules to allow required traffic.

Details Info

Network ACL ID: acl-0028a09c49ca7da0b
Associated with: subnet-0c05357b6dde83c7a / Public 1
Owner: 944362433406
Default: Yes
VPC ID: vpc-023351c79befbc912 / NextWork VPC

Inbound rules (2)

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

Details Info

Network ACL ID: acl-0c902fe76cc963c85
Associated with: subnet-0885e084c9794fb2e / Private 1
Owner: 944362433406
Default: No
VPC ID: vpc-023351c79befbc912 / NextWork VPC

Inbound rules (2)

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	All traffic	All	All	10.0.0.0/24	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

Step 7: Create a Private Subnet

Add a private subnet to your VPC for internal resources.

Actions:

- In the VPC Dashboard, choose 'Subnets' → 'Create subnet'.
- Select NextWork VPC and enter Subnet name (e.g., Private 1).
- Select an Availability Zone different from the public subnet for redundancy.
- Enter an IPv4 CIDR block (e.g., 10.0.1.0/24).
- Click 'Create subnet' and leave auto-assign public IPv4 disabled.

subnet-0885e084c9794fb2e / Private 1

Details

Subnet ID	arn:aws:ec2:ap-south-1:944362433406:subnet/subnet-0885e084c9794fb2e	State	Available
IPv4 CIDR	10.0.1.0/24	IPv6 CIDR	-
Availability Zone	aps1-az3 (ap-south-1b)	Network border group	ap-south-1
Network ACL	acl-0c902fe76cc963c85 NextWork Private NACL	Default subnet	No
Auto-assign customer-owned IPv4 address	No	Customer-owned IPv4 pool	-
IPv6 CIDR reservations	-	IPv6-only	No
Resource name DNS AAAA record	Disabled	DNS64	Disabled

Flow logs

Flow logs

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 8: Create a Private Route Table

Create and associate a route table for private subnet routing.

Actions:

- In 'Route tables', click 'Create route table'.
- Enter a Name tag (e.g., NextWork Private RT) and select your VPC.
- Create the route table and go to 'Subnet associations'.
- Edit subnet associations and select your Private 1 subnet.
- Do not add routes to an Internet Gateway.

rtb-00d3252319d1dc20b / NextWork Private Route Table

Details

Route table ID	rtb-00d3252319d1dc20b	Main	No
VPC	vpc-023351c79befbc912 NextWork VPC	Owner ID	944362433406

Explicit subnet associations

subnet-0885e084c9794fb2e / Private 1

Edge associations

-

Routes

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	Create Route Table

Step 9: Create a Private Network ACL

Add a subnet-level NACL protecting your private subnet.

Actions:

- In 'Network ACLs', click 'Create network ACL'.
- Name it (e.g., NextWork Private NACL) and choose your VPC.
- Create inbound rules to allow necessary internal traffic (e.g., ICMP from public subnet) as required.
- Configure outbound rules appropriately to restrict internet-bound access.
- Associate the NACL with your Private subnet under 'Subnet associations'.

The screenshot shows the AWS VPC Network ACLs page. The top navigation bar includes 'Search' and 'Account ID: 9443-6243-3406'. The left sidebar has sections for Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, NAT gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Peering connections) and Security (Network ACLs, Security groups). The main content area shows 'acl-0c902fe76cc963c85 / NextWork Private NACL'. It displays 'Details' (Network ACL ID: acl-0c902fe76cc963c85, Associated with: subnet-0885e084c9794fb2e / Private 1, Owner: 944362433406, Default: No, VPC ID: vpc-023351c79befbc912 / NextWork VPC), 'Inbound rules' (2), and 'Outbound rules'. The 'Inbound rules' table has columns: Rule number, Type, Protocol, Port range, Source, and Allow/Deny. It contains two entries: rule 100 allowing all traffic from 10.0.0.0/24 and a wildcard entry allowing all traffic from 0.0.0.0/0.

Step 10: Launch EC2 Instance in Public Subnet

Launch a public EC2 instance with a public IP and public security group.

Actions:

- Open EC2 Console → Instances → Launch Instance.
- Enter a Name tag (e.g., NextWork Public Server).
- Select an AMI (e.g., Amazon Linux 2023 AMI) and instance type (t2.micro).
- Under Network settings, select NextWork VPC and the Public subnet.
- Enable Auto-assign Public IP.
- Choose or create a Key pair (NextWork key pair) and download the private key (.pem).
- Select the NextWork Public Security Group.
- Click 'Launch' and wait for the instance to reach 'running' state.

The screenshot shows the AWS EC2 Instances page. The top navigation bar includes 'Search' and 'Account ID: 9443-6243-3406'. The left sidebar has sections for EC2 (Dashboard, AWS Global View, Events, Instances, Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations) and Images (AMI). The main content area shows 'Instance summary for i-06c8dafaf521bcda4 (NextWork Public Server)'. It displays details like Public IPv4 address (13.232.144.25), Instance state (Running), Private IP DNS name (ip-10-0-0-41.ap-south-1.compute.internal), Instance type (t2.micro), VPC ID (vpc-023351c79befbc912), Subnet ID (subnet-0c05357b6dde83c7a), and other fields like Hostname type, Answer private resource DNS name, Auto-assigned IP address, IAM Role, and various addresses (Private IPv4, Public DNS, Elastic IP, AWS Compute Optimizer finding, Auto Scaling Group name).

Step 11: Launch EC2 Instance in Private Subnet

Launch a private EC2 instance that has no public IP.

Actions:

- EC2 Console → Launch Instance.
- Enter Name tag (e.g., NextWork Private Server).
- Choose the same AMI and instance type (t2.micro).
- Under Network settings, select NextWork VPC and the Private subnet.
- Disable Auto-assign Public IP.
- Create/select a private security group (NextWork Private Security Group).
- Click 'Launch' and wait for the instance to become 'running'. Note: connect via Bastion/Jump host from Public subnet if needed.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with 'EC2' selected, followed by links for Dashboard, AWS Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, and Images. The main content area is titled 'Instance summary for i-025e211123ae1e798 (NextWork Private Server)' and includes sections for Instance ID (i-025e211123ae1e798), IPv6 address (empty), Hostname type (IP name: ip-10-0-1-252.ap-south-1.compute.internal), Answer private resource DNS name (empty), Auto-assigned IP address (empty), IAM Role (empty), Public IPv4 address (empty), Instance state (Running), Private IP DNS name (ip-10-0-1-252.ap-south-1.compute.internal), Instance type (t2.micro), VPC ID (vpc-023351c79befbc912 (NextWork VPC)), Subnet ID (subnet-0885e084c9794fb2e (Private 1)), and various status indicators like Private IPv4 addresses (10.0.1.252), Public DNS (empty), Elastic IP addresses (empty), AWS Compute Optimizer finding (Opt-in to AWS Compute Optimizer for recommendations.), and Auto Scaling Group name (empty). A 'Connect' button is also present.

Step 12: Use Amazon VPC Wizard (Optional)

Use the VPC Wizard to create a pre-configured VPC and resources quickly.

Actions:

- In the VPC Dashboard, click 'Launch VPC Wizard' or 'Create VPC' → 'VPC and more'.
- Choose a template (e.g., VPC with Public and Private Subnets).
- Review resource map (subnets, route tables, IGW, NACLs) and configuration.
- Adjust AZ count, subnet CIDRs, and naming as needed and create the VPC.

The screenshot shows the AWS VPC Create VPC wizard. The first step, 'VPC settings', allows creating either 'VPC only' or 'VPC and more'. It includes fields for 'Name tag auto-generation' (with 'Auto-generate' checked and 'Demo' as the tag), 'IPv4 CIDR block' (CIDR 10.0.0.0/16, resulting in 65,536 IPs), and 'CIDR block size must be between /16 and /28'. The second step, 'Preview', shows a visual representation of the VPC structure. It features a central 'VPC' box labeled 'Show details' and 'Your AWS virtual network', with a 'Name tag' field containing 'Demo-vpc'. To the right, there are two boxes: 'Subnets (2)' containing 'ap-south-1a' with subnets 'Demo-subnet-public1-ap-south-1a' and 'Demo-subnet-private1-ap-south-1a'; and 'Route tables (2)' containing 'Demo-rtb-public' and 'Demo-rtb-private1-ap-south-1a'. Arrows indicate the relationships between the VPC, subnets, and route tables.

Step 13: Set up your VPC basics (verify)

Confirm that prior project resources are present and running.

Actions:

- Open the VPC console and confirm NextWork VPC exists.
- Verify Public and Private subnets are present.
- Confirm the Internet Gateway is attached to NextWork VPC.
- Check route tables (public RT has 0.0.0.0/0 → IGW).
- Ensure NACLs and Security Groups exist and are named appropriately.
- Verify EC2 instances (NextWork Public Server and NextWork Private Server) are running.

Step 14: Connect to NextWork Public Server (EC2 Instance Connect)

Use EC2 Instance Connect from the AWS Console to SSH into the public EC2 instance.

Actions:

- Open EC2 Console → Instances and select NextWork Public Server.
- Click 'Connect' → 'EC2 Instance Connect'.
- Keep default username (ec2-user) and public IP pre-filled, then click 'Connect'.
- If connection fails, inspect the NextWork Public Security Group inbound rules and ensure SSH (port 22) is allowed.
- Temporarily allow SSH from Anywhere-IPv4 to ensure EC2 Instance Connect can reach your instance (note: restrict later for security).

```
[ec2-user@ip-10-0-0-41 ~]$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
xvda    202:0   0   8G  0 disk 
└─xvda1  202:1   0   8G  0 part /
xvda127 259:0   0   1M  0 part /
xvda128 259:1   0   10M 0 part /boot/efi
[ec2-user@ip-10-0-0-41 ~]$
```

i-06c8dafaf521bcda4 (NextWork Public Server)
Public IPs: 13.232.144.25 Private IPs: 10.0.0.41

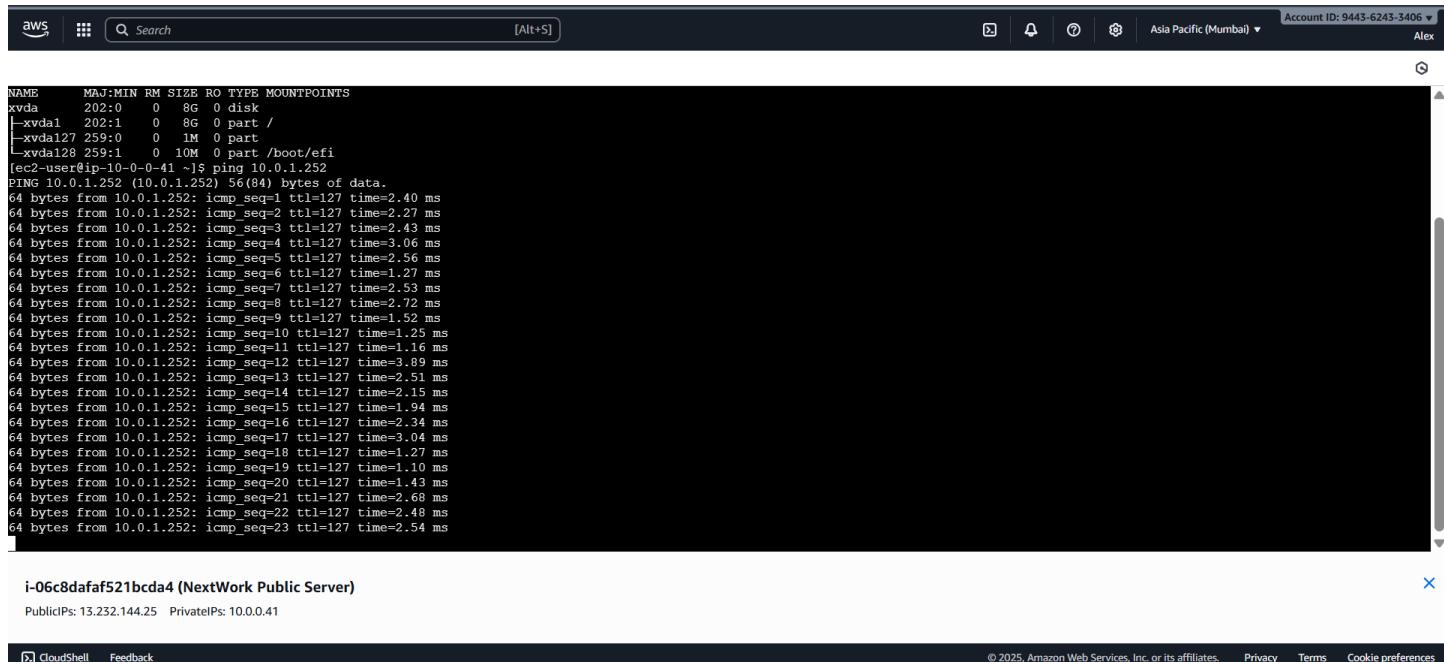
CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 15: Test connectivity between your EC2 instances (ping)

From the Public Server, ping the Private Server's private IP and troubleshoot if needed.

Actions:

- On the EC2 console, copy the Private IPv4 address of the NextWork Private Server.
- In the Public Server terminal (EC2 Instance Connect), run: ping <Private IPv4 Address>
- If you receive no replies, inspect the NextWork Private NACL rules—ensure ICMP (All ICMP - IPv4) is allowed inbound from 10.0.0.0/24 and outbound to 10.0.0.0/24.
- Inspect NextWork Private Security Group inbound rules—add All ICMP - IPv4 with Source = NextWork Public Security Group.
- After updates, run ping again and confirm multiple replies are received.



A screenshot of the AWS CloudShell interface. The terminal window shows the output of a ping command to 10.0.1.252. The output includes details about each ICMP echo request and response, such as sequence number, TTL, and time taken. The terminal also displays disk information and a user prompt at the end.

```
aws CloudShell Search [Alt+S] Account ID: 9443-6243-3406 Alex
NAME      MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
xvda     202:0    0   8G  0 disk
└─xvda1   202:1    0   8G  0 part /
  └─xvda127 259:0    0   1M  0 part
  └─xvda128 259:1    0   10M 0 part /boot/efi
[ec2-user@ip-10-0-0-41 ~]$ ping 10.0.1.252
PING 10.0.1.252 (10.0.1.252) 56(84) bytes of data.
64 bytes from 10.0.1.252: icmp_seq=1 ttl=127 time=2.40 ms
64 bytes from 10.0.1.252: icmp_seq=2 ttl=127 time=2.27 ms
64 bytes from 10.0.1.252: icmp_seq=3 ttl=127 time=2.43 ms
64 bytes from 10.0.1.252: icmp_seq=4 ttl=127 time=3.06 ms
64 bytes from 10.0.1.252: icmp_seq=5 ttl=127 time=2.56 ms
64 bytes from 10.0.1.252: icmp_seq=6 ttl=127 time=1.27 ms
64 bytes from 10.0.1.252: icmp_seq=7 ttl=127 time=2.53 ms
64 bytes from 10.0.1.252: icmp_seq=8 ttl=127 time=2.72 ms
64 bytes from 10.0.1.252: icmp_seq=9 ttl=127 time=1.52 ms
64 bytes from 10.0.1.252: icmp_seq=10 ttl=127 time=1.25 ms
64 bytes from 10.0.1.252: icmp_seq=11 ttl=127 time=1.16 ms
64 bytes from 10.0.1.252: icmp_seq=12 ttl=127 time=3.89 ms
64 bytes from 10.0.1.252: icmp_seq=13 ttl=127 time=2.51 ms
64 bytes from 10.0.1.252: icmp_seq=14 ttl=127 time=2.15 ms
64 bytes from 10.0.1.252: icmp_seq=15 ttl=127 time=1.94 ms
64 bytes from 10.0.1.252: icmp_seq=16 ttl=127 time=2.34 ms
64 bytes from 10.0.1.252: icmp_seq=17 ttl=127 time=3.04 ms
64 bytes from 10.0.1.252: icmp_seq=18 ttl=127 time=1.27 ms
64 bytes from 10.0.1.252: icmp_seq=19 ttl=127 time=1.10 ms
64 bytes from 10.0.1.252: icmp_seq=20 ttl=127 time=1.43 ms
64 bytes from 10.0.1.252: icmp_seq=21 ttl=127 time=2.68 ms
64 bytes from 10.0.1.252: icmp_seq=22 ttl=127 time=2.48 ms
64 bytes from 10.0.1.252: icmp_seq=23 ttl=127 time=2.54 ms
[1]
i-06c8dafaf521bcda4 (NextWork Public Server)
PublicIPs: 13.232.144.25 PrivateIPs: 10.0.0.41
```

Step 16: Test VPC connectivity with the internet (curl)

From the Public Server, use curl to verify external HTTP connectivity.

Actions:

- In the Public Server terminal, stop any running ping (Ctrl+C).
- Run: curl example.com and confirm you receive HTML output.
- Run: curl learn.nextwork.org and follow redirects if necessary (curl will show redirect responses).
- Use the returned URLs to curl the final page (e.g., curl https://learn.nextwork.org/projects/...).
- Confirm the HTML output appears—this verifies outbound internet connectivity from the public subnet.

AWS Search [Alt+S] Account ID: 9443-6243-3406 ▾ Alex

```
[ec2-user@ip-10-0-0-41 ~]$ lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
xvda    202:0    0 8G  0 disk
└─xvda1  202:1    0 8G  0 part /
  └─xvda127 259:0    0 1M  0 part
  └─xvda128 259:1    0 10M 0 part /boot/efi
[ec2-user@ip-10-0-0-41 ~]$ curl example.com
<!doctype html>
<html>
<head>
  <title>Example Domain</title>
  <meta charset="utf-8" />
  <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <style type="text/css">
    body {
      background-color: #f0f0f2;
      margin: 0;
      padding: 0;
      font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
    }
    div {
      width: 600px;
      margin: 5em auto;
      padding: 2em;
      background-color: #fdfdff;
      border-radius: 0.5em;
      box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
    }
  </style>
</head>
<body>
  <div>
    <h1>Hello World</h1>
    <p>This is a simple example web page. It has a <a href="#">link</a>, some <code>HTML</code>, and a <code>CSS</code> file.
    <p>The <code>lsblk</code> command shows we have an 8G xvda volume with a 1M /boot/efi partition and a 10M / partition.
    <p>The <code>curl example.com</code> command shows the contents of this page.
  </div>
</body>
</html>
```

i-06c8dafaf521bcda4 (NextWork Public Server)

Public IPs: 13.232.144.25 Private IPs: 10.0.0.41



CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences