



Building a Virtual Private Cloud



Kanika Mathur
github.com/KanikaGenesis



VPC settings

Resources to create [Info](#)

Create only the VPC resource or the VPC and other networking resources.

VPC only

VPC and more

Name tag - *optional*

Creates a tag with a key of 'Name' and a value that you specify.

NextWork VPC

IPv4 CIDR block [Info](#)

IPv4 CIDR manual input

IPAM-allocated IPv4 CIDR block

IPv4 CIDR

10.0.0.0/16

CIDR block size must be between /16 and /28.

My first VPC

VPCs are isolated sections of AWS cloud which keeps my AWS resources private and secure.

There was already a default VPC in my account ever since my AWS account was created. This is because AWS has setup a default VPC to allow me to deploy resources like EC2 instances/RDS databases right away (without having to create my own VPC from scratch).

To set up my VPC, I had to define an IPv4 CIDR block, which means a range of IP addresses that my VPC can allocate to the resources deployed into my VPC.

My VPC setup page

VPC settings

Resources to create [Info](#)
Create only the VPC resource or the VPC and other networking resources.

VPC only VPC and more

Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.
NextWork VPC

IPv4 CIDR block [Info](#)

IPv4 CIDR manual input
 IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/16

CIDR block size must be between /16 and /28.

Subnets

- Subnets are subsections of my VPC, just like how neighborhoods are subsections of a city.
- There are already subnets existing in my account, one for every Availability Zone in the Region that I've set up in my VPC in. Since my region is N. Virginia (Us-east-1), which has six Availability Zones, I have six default subnets already.
- I named my subnet Public 1, but that doesn't automatically make my subnet a public subnet. For a subnet to be considered public, it has to have a route to an internet gateway.

My created subnet!



Subnets (1/7) Info				
<input type="text"/> Find resources by attribute or tag				
-	Name	Subnet ID	State	VPC
<input type="checkbox"/>	-	subnet-07ec14466cc3c1065	Available	vpc-0d07efa809c
<input type="checkbox"/>	-	subnet-0f941aec08737ba28	Available	vpc-0d07efa809c
<input type="checkbox"/>	-	subnet-050dcbf67389b5d05	Available	vpc-0d07efa809c
<input type="checkbox"/>	-	subnet-0bf412ff3d8331cd5	Available	vpc-0d07efa809c
<input type="checkbox"/>	-	subnet-0dbeb29b91e7216fa	Available	vpc-0d07efa809c
<input type="checkbox"/>	-	subnet-0e8aa553ee3421fb0	Available	vpc-0d07efa809c
<input checked="" type="checkbox"/>	Public 1	subnet-06424db0173e8d0ec	Available	vpc-00513cd1352

Internet gateways

- Internet gateways are the key VPC components that allows internet access for the resources in my VPC/subnet. An internet gateway is also how users in the public can access my resources in a public subnet.
- Attaching an internet gateway means resources in your VPC can now access the internet. The EC2 instances with public IP addresses also become accessible to users, so your applications hosted on those servers become public too.
- While I've set up an internet gateway and attached it to a VPC, I still have to set up route tables. Route tables will help EC2 instances/resources in my VPC to find their way to the internet gateway attached to my VPC- otherwise, even if that connection has been established between my VPC and the internet gateway, my EC2 instances would still struggle to access the internet.

My created internet gateway!

Internet gateways (2) [Info](#)

 [Search](#)

<input type="checkbox"/>	Name	▼	Internet gateway ID	▼	State
<input type="checkbox"/>	-		igw-05564877bc849532a		Attached
<input type="checkbox"/>	NextWork IG		igw-0038695238fa35652		Attached



VPC Traffic Flow and Security



Kanika Mathur
github.com/KanikaGenesis



sg-0770ac2579dbd44e7 - NextWork Security Group

Details

Security group name NextWork Security Group	Security group ID sg-0770ac2579dbd44e7	Description A Security Group for the NextWork VPC.	VPC ID vpc-1
Owner 991380288324	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

[Inbound rules](#) [Outbound rules](#) [Tags](#)

Inbound rules (1)

<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range
<input type="checkbox"/>	-	sgr-0810f8ecfcfa0b1c5	IPv4	HTTP	TCP	80

Route tables

- Route tables are like the GPS that directs traffic within my VPC to the correct destination.
- Routes tables are needed to make a subnet public because a subnet needs to have a route to an internet gateway in order to be considered public. A route table is the only way to establish this connection.
- A route table is made up of routes, which are defined by its destination and target:
 - The **destination** is the range of IP addresses that traffic in my VPC is trying to reach.
 - The **target** is the road/path that the traffic will use to get to their destination.
- The route in my route table that directed internet-bound traffic to my internet gateway had a **destination** of 0.0.0.0/0 and a **target** of my NextWork IG (Internal gateway).

This route directs Internet-bound traffic to my internet gateway

A screenshot of the AWS Route Table configuration interface. The table has two columns: 'Destination' and 'Target'. There are two rows of data:

Destination	Target	Status
10.0.0.0/16	local	Active
0.0.0.0/0	Internet Gateway igw-0038695238fa35652	Active

A blue arrow points from the text "This route directs Internet-bound traffic to my internet gateway" to the 'Destination' column of the second row. The 'Add route' button is visible at the bottom left.

Security groups

- Security groups are like security guards that monitor both inbound and outbound traffic **at the resource level, i.e. every single resource in a subnet/VPC has a security group.**
- Security groups control traffic flow using two types of rules:
 - **Inbound rules** are the rules that monitor/restrict inbound traffic, e.g. users visiting a web app I'm hosting.
 - **Outbound rules** are the rules that monitor/restrict outbound traffic, e.g. my web app requesting data from a public source.
- By default, an outbound rule will allow all outbound traffic.
- I also configured an inbound rule that allows all inbound HTTP traffic.

My configured security group

The screenshot shows the AWS Security Groups console for a security group named "sg-0770ac2579dbd44e7 - NextWork Security Group". A blue circle highlights the group name. The "Details" section shows the following information:

Security group name	sg-0770ac2579dbd44e7	Description	A Security Group for the NextWork VPC.
Owner	991380288324	Inbound rules count	1 Permission entry
VPC ID	vpc-0	Outbound rules count	1 Permission entry

The "Inbound rules" tab is selected, showing one rule:

Name	Security group rule...	Type	Protocol	Port range	
-	sgr-0810f8ecfcfa0b1c5	IPv4	HTTP	TCP	80



Network ACLs

- Network ACLs are like community watchmen that secures my network at a subnet level.
- The difference between a security group and a network ACL is their scope i.e a security group secures resources at the resource level (so every resource in my VPC is associated with a security group), while network ACLS secures my network at the **subnet** level (every single subnet in my VPC is associated with a network ACL)
- Having both network ACLs and security groups is a good security best practice. It creates a dual layer of security that makes sure inbound/outbound traffic go through at least two checks.
- Similar to security groups, network ACLs use inbound and outbound rules:
 - A default network ACL's inbound rule is set up to allow all incoming traffic.
 - A default network ACL's outbound rule is set up to allow all outgoing traffic.
 - In contrast, a custom ACL's inbound and outbound rules are automatically set to deny all incoming/ outgoing traffic.

Kanika Mathur
github.com/KanikaGenesis

NextWork.org

My network ACL's inbound rules

Inbound rules (2)							Edit inbound rules
Filter inbound rules							< 1 > ⌂
Rule number	Type	Protocol	Port range	Source	Allow/Deny	Action	
00	All traffic	All	All	0.0.0.0/0	<input checked="" type="checkbox"/> Allow		
	All traffic	All	All	0.0.0.0/0	<input type="checkbox"/> Deny		

My network ACL's outbound rules

Outbound rules (2)							Edit outbound rules
Filter outbound rules							< 1 > ⌂
Rule number	Type	Protocol	Port range	Destination	Allow/Deny	Action	
100	All traffic	All	All	0.0.0.0/0	<input checked="" type="checkbox"/> Allow		
*	All traffic	All	All	0.0.0.0/0	<input type="checkbox"/> Deny		



Creating a Private Subnet



Kanika Mathur
github.com/KanikaGenesis

VPC

VPC ID
Create subnets in this VPC.
vpc-00513cd135254f395 (NextWork VPC)

Associated VPC CIDRs

IPv4 CIDRs
10.0.0.0/16

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
NextWork Private Subnet
The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US East (N. Virginia) / us-east-1b

IPv4 VPC CIDR block [Info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
10.0.0.0/16

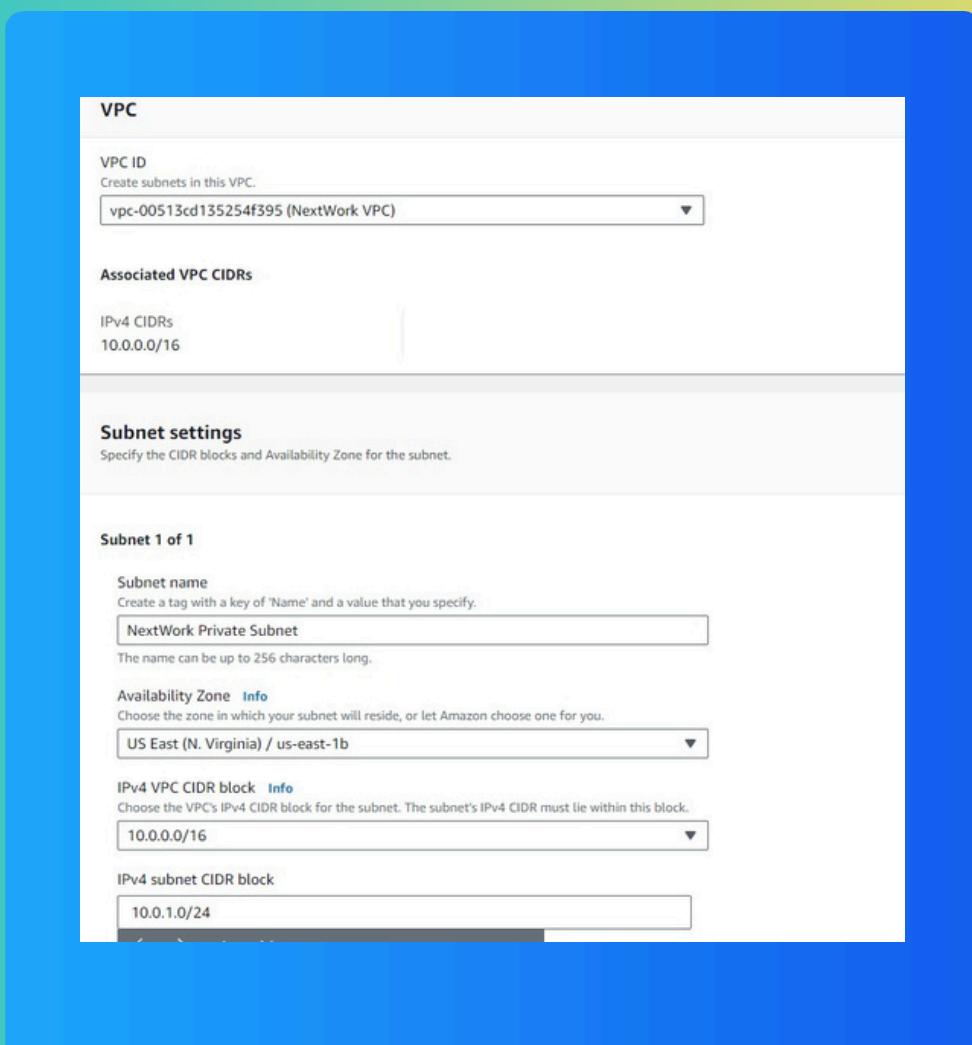
IPv4 subnet CIDR block
10.0.1.0/24
< > ^ v

Private vs Public Subnets

The difference between public and private subnets is that public subnets are accessible by and can access the internet, while private subnets are completely isolated from the internet by default.

Having private subnets are useful because keeping resources away from the internet is extremely important for the security of confidential resources/data.

My private and public subnets cannot have the same IPv4 CIDR block i.e. the same range of IP addresses. The CIDR block for every subnet must be unique and cannot overlap with another subnet.



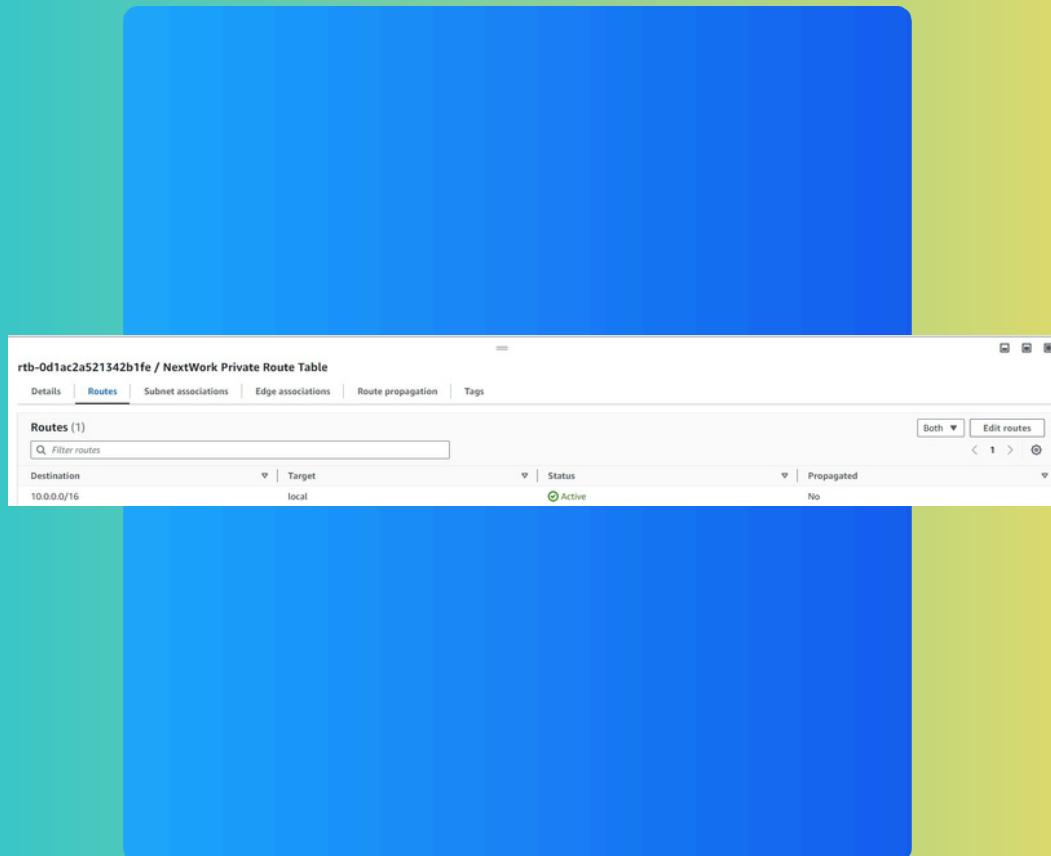


A dedicated route table

By default, my private subnet is associated with the default route table i.e. a route table that has a route to an internet gateway.

To make my subnet private, I had to set up a new route table because my subnet can't have a route to an internet gateway.

My private subnet's dedicated route table only has one inbound and one outbound rule that allows internal communication i.e with a destination of another resource within my VPC.



Kanika Mathur
github.com/KanikaGenesis

NextWork.org

A new network ACL

By default, my private subnet is associated with the default network ACL that's set up for every VPC created in my AWS account.

I set up a dedicated network ACL for my private subnet because a network ACL becomes crucial in the event of security breaches where traffic that has compromised my public subnet can get access to my private subnet if I have network ACL rules.

My new network ACL has two simple rules - deny all inbound and deny all outbound traffic.

The screenshot shows the AWS Network ACL Inbound Rules table. It has a header row with columns: Rule number, Type, Protocol, Port range, Source, and Allow/Deny. There is one data row with the following values:

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



Launching VPC Resources



Kanika Mathur
github.com/KanikaGenesis

The screenshot shows the AWS VPC settings configuration page. On the left, the 'VPC settings' section includes fields for 'Resources to create' (set to 'VPC and more'), 'Name tag auto-generation' (set to 'Auto-generate' with 'network' as the prefix), 'IPv4 CIDR block' (set to '10.0.0.0/16'), 'IPv6 CIDR block' (set to 'No IPv6 CIDR block'), 'Tenancy' (set to 'Default'), 'Number of Availability Zones (AZs)' (set to 3), and 'Number of public subnets' (set to 1). On the right, the 'Preview' section displays a network diagram for the 'nextwork-vpc' VPC. The diagram shows the VPC structure with associated Subnets (us-east-1a and us-east-1b), Route Tables (rtt-public and rtt-private1-us-east-1a), and Network connections (igw and vpc-s3).



Setting Up Direct VM Access

Directly accessing a virtual machine means "logging into" the EC2 instance (instead of managing it at a higher level with the AWS Management Console). This includes operations like installing software and changing my EC2 instance's configurations.

SSH is a key method for directly accessing a VM

SSH means Secure Shell, and it is both a protocol and a traffic type. It is the protocol that matches key pairs and enables direct VM access, and once a connection is set up, it is a traffic type that encrypts communication/ data being transferred.

To enable direct access, I set up key pairs

Key pairs are tools that help engineers authenticate themselves when trying to get access to a virtual machine, e.g. an EC2 instance. Key pairs work by having two private keys-a private key for the VM and a matching private key for the resource/user.

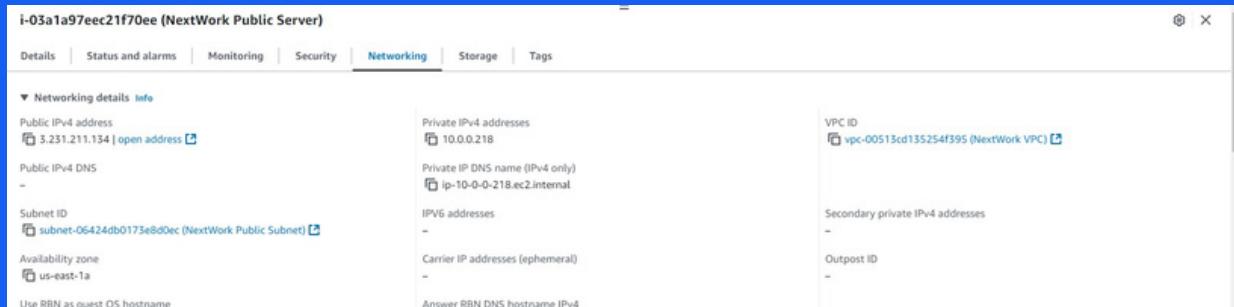
A private key's file format means the file type that my key is stored in. My private key's file format was .pem, which is a widely accepted file format that most servers will be able to read/use.

Kanika Mathur
github.com/KanikaGenesis

NextWork.org

Launching a public server

I had to change my EC2 instance's networking settings by changing the VPC and the subnet my EC2 instance was going to be launched in! I updated both to my NextWork VPC and my Public Subnet respectively. I also used my existing Public security group.



Kanika Mathur
github.com/KanikaGenesis

NextWork.org

Launching a private server

My private server has its own dedicated security group because the NextWork public security group allows in ALL HTTP traffic which would leave our private server much more vulnerable to security attacks/ risks.

My private server's security group's source is my NextWork Security Group, which means only SSH traffic coming from resources associated with that security group would be allowed.

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, sg-0770ac2579dbd44e7) Remove

Type Info ssh	Protocol Info TCP	Port range Info 22
Source type Info Custom	Source Info <input type="text"/> Add CIDR, prefix list or security group sg-0770ac2579dbd44e7 X	Description - optional Info e.g. SSH for admin desktop

[Add security group rule](#)

Speeding up VPC creation

I used an alternative way to set up an Amazon VPC! This time, I used the "VPC and more" option, which gives me a VPC resource map to use when creating the VPC and all of its components, e.g. security groups, route tables and internet gateways.

A VPC resource map is a visual diagram that maps out my VPC's components and their relationship/ connections between them. A resource map is interactive i.e. it will highlight the connections relevant to a resource that I highlight/ hover over.

My new VPC has a CIDR block of 10.0.0.0/16. It is possible for my new VPC to have the same IPv4 CIDR block as my existing VPC (NextWork VPC) because VPCs are already isolated from each other. Still, this is not best practice if we need VPC peering.



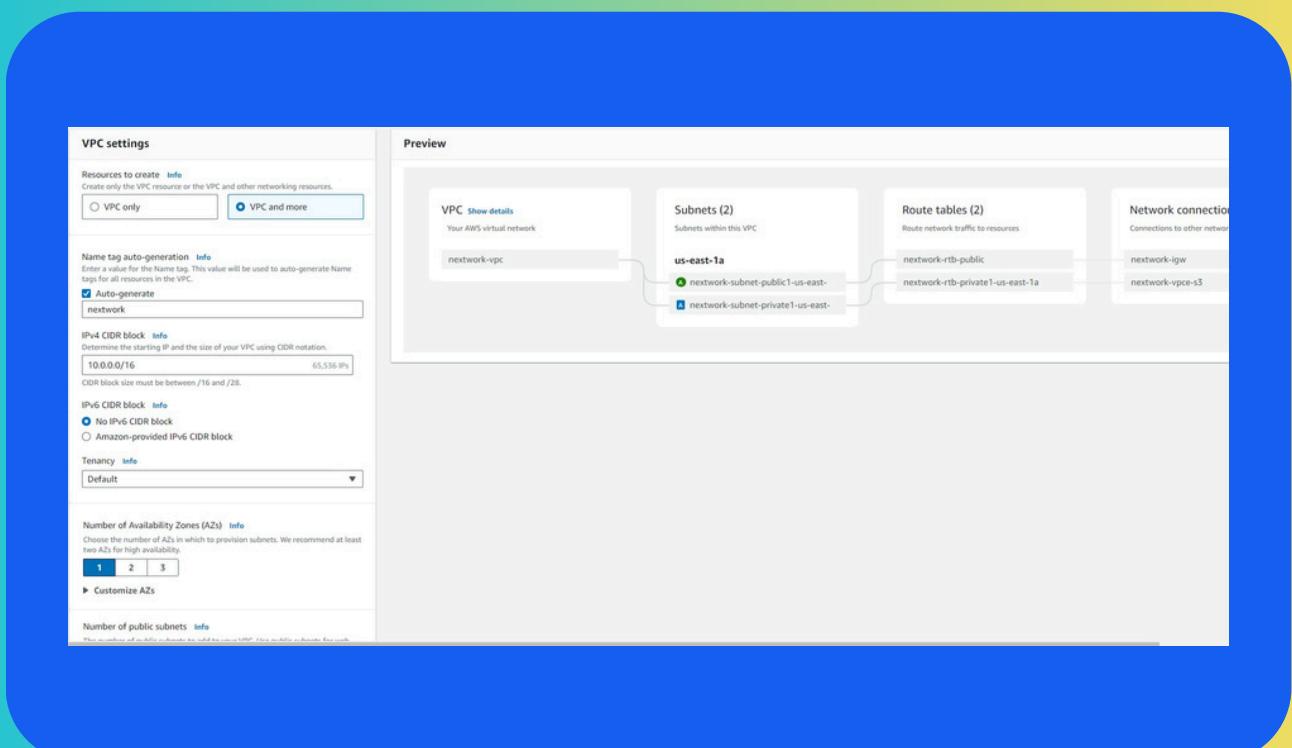


Speeding up VPC creation

Tips for using the VPC resource map

When determining the number of public subnets in my VPC, I only had two options: either none or one in each Availability Zone for my VPC. This was because it is best practice (improves redundancy and high availability) to have at least subnet/AZ.

The setup page also offered to create NAT gateways, which are connectors/gateways that will let resources in private subnet get access to the internet (e.g. for security updates) while still blocking off incoming traffic from the internet.





Testing VPC Connectivity



Kanika Mathur
github.com/KanikaGenesis

```
src="https://js.sentry-cdn.com/149a6bc4cd616ff81bea862cf35e7leb.min.js"
data-lazy="no"
crossorigin="anonymous"
></script>

<project-app
  project="{"id": "aws-host-a-website-on-s3", "metadata": [{"category": "Storage", "concepts": ["AWS Lambda", "Amazon S3", "AWS CloudFront", "AWS CloudWatch Metrics", "AWS CloudWatch Logs", "AWS Lambda@Edge", "AWS Lambda triggers"], "cost": "$0", "description": "Let's start your very own website on Amazon S3!", "difficulty": "Easy peasy", "icon": "static/icon3.png", "needs": "Create one AWS account - \u003ca href=\u0034;https://docs.aws.amazon.com/accounts/latest/reference/manage-acct-creating.html\u0034;\u003c/a\u003eCreate one!", "order": 1, "shareTemplate": true}, {"category": "AWS Lambda", "concepts": ["AWS Lambda", "AWS Lambda triggers"], "cost": "$0", "description": "Just wrapped up a thrilling AWS Lambda challenge mode engaged! 🎉", "difficulty": "Medium", "icon": "static/icon4.png", "needs": "Tackled public access settings and fixed an interesting challenge with the website's visibility.", "order": 2, "shareTemplate": true}, {"category": "Amazon S3", "concepts": ["Amazon S3", "AWS CloudFront", "AWS CloudWatch Metrics", "AWS CloudWatch Logs", "AWS Lambda@Edge", "AWS Lambda triggers"], "cost": "$0", "description": "Created and configured an Amazon S3 bucket, complete with ACLs, versioning, and public access.", "difficulty": "Medium", "icon": "static/icon5.png", "needs": "Uploaded website content, diving deep into static websites function and how to host them on S3.", "order": 3, "shareTemplate": true}, {"category": "AWS CloudFront", "concepts": ["AWS CloudFront", "AWS CloudWatch Metrics", "AWS CloudWatch Logs", "AWS Lambda@Edge", "AWS Lambda triggers"], "cost": "$0", "description": "Tackled public access settings and fixed an interesting challenge with the website's visibility.", "difficulty": "Medium", "icon": "static/icon6.png", "needs": "My journey from creating buckets to deploying a fully functional static website in my documentation below.", "order": 4, "shareTemplate": true}, {"category": "AWS CloudWatch Metrics", "concepts": ["AWS CloudWatch Metrics", "AWS CloudWatch Logs", "AWS Lambda@Edge", "AWS Lambda triggers"], "cost": "$0", "description": "Shoutout to all AWS learners - let's connect, are tips, and keep improving!", "difficulty": "Medium", "icon": "static/icon7.png", "needs": "Big thanks to @NextWork for setting up this engaging challenge. Ready for the next one! link.nextwork.org/linkedIn", "order": 5, "shareTemplate": false}, {"category": "AWS CloudWatch Logs", "concepts": ["AWS CloudWatch Logs", "AWS Lambda@Edge", "AWS Lambda triggers"], "cost": "$0", "description": "Leaves most of up to you. Great for those looking for a challenge.", "difficulty": "Medium", "icon": "static/icon8.png", "needs": "Low Touch", "order": 6, "shareTemplate": false}, {"category": "AWS Lambda@Edge", "concepts": ["AWS Lambda@Edge", "AWS Lambda triggers"], "cost": "$0", "description": "Gives p-by-step instructions for every part. Great for beginners!", "difficulty": "Medium", "icon": "static/icon9.png", "needs": "High Touch", "order": 7, "shareTemplate": false}, {"category": "AWS Lambda triggers", "concepts": ["AWS Lambda triggers"], "cost": "$0", "description": "INCOMPLETE", "difficulty": "Medium", "icon": "static/icon10.png", "needs": "High Touch", "order": 8, "shareTemplate": false}], "selectedTrack": ""}>
</project-app>

</body>
</html>
```

A circular portrait of a young woman with long dark hair, smiling. She is wearing a light blue denim jacket over a white shirt. The background shows shelves filled with books, indicating a library environment.

Kanika Mathur
github.com/KanikaGenesis

NextWork.org

Connecting to an EC2 Instance

Connectivity means getting resources in our network to communicate with each other and how well they can communicate/deliver data to each other. Without connectivity, resources in our network cannot communicate eg users cannot access our application.

My first connectivity test was whether I could connect to my network's Public Server (an EC2 instance)

```
~\### Amazon Linux 2023
~~\###\
~~\###|
~~ \### https://aws.amazon.com/linux/amazon-linux-2023
~~ V~'.'->
~~ ~~~ /-
~~~ \_/
~~~ \_/
~~~ \_m/ \
[ec2-user@ip-10-0-0-218 ~]$
```



Kanika Mathur
github.com/KanikaGenesis

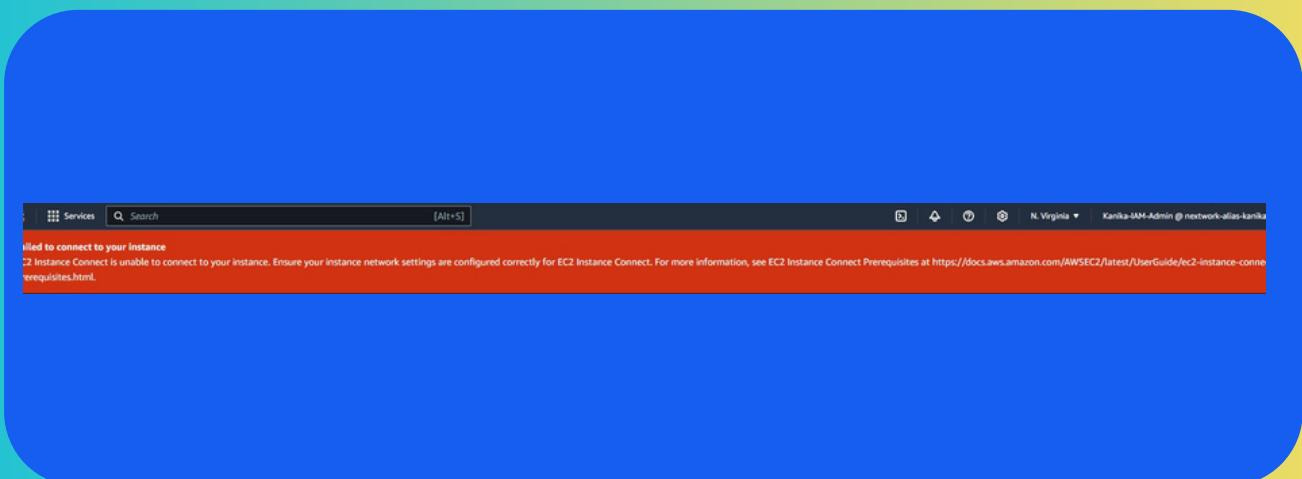
NextWork.org

EC2 Instance Connect

I connected to my EC2 instance using EC2 Instance Connect, which is a tool provided by Amazon EC2 that allows us to directly access an EC2 instance using AWS Management Console! We no longer need to manage key pairs or use a SSH client to get access.

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 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.





Kanika Mathur
github.com/KanikaGenesis

NextWork.org

Connectivity Between Servers

Ping is a tool to test the connectivity between two servers and also the response time, 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 10.0.1.106', where 10.0.1.106 is the private IPv4 address of my Private Server.

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

```
Administrator: ~ [Alt+L] N: Virginia • Kanika-MH-Admin @ network-alien-kar
$ ping 10.0.1.106
PING 10.0.1.106(10.0.1.106) 56(84) bytes of data.
from 10.0.1.106 icmp_seq=69 ttl=127 time=0.47 ms
from 10.0.1.106 icmp_seq=70 ttl=127 time=0.44 ms
from 10.0.1.106 icmp_seq=71 ttl=127 time=0.09 ms
from 10.0.1.106 icmp_seq=72 ttl=127 time=0.09 ms
from 10.0.1.106 icmp_seq=73 ttl=127 time=0.55 ms
from 10.0.1.106 icmp_seq=74 ttl=127 time=0.23 ms
from 10.0.1.106 icmp_seq=75 ttl=127 time=0.09 ms
from 10.0.1.106 icmp_seq=76 ttl=127 time=0.70 ms
from 10.0.1.106 icmp_seq=77 ttl=127 time=0.01 ms
from 10.0.1.106 icmp_seq=78 ttl=127 time=0.09 ms
from 10.0.1.106 icmp_seq=79 ttl=127 time=0.09 ms
from 10.0.1.106 icmp_seq=80 ttl=127 time=0.95 ms
from 10.0.1.106 icmp_seq=81 ttl=127 time=0.41 ms
from 10.0.1.106 icmp_seq=82 ttl=127 time=0.41 ms
from 10.0.1.106 icmp_seq=83 ttl=127 time=0.16 ms
from 10.0.1.106 icmp_seq=84 ttl=127 time=0.11 ms
from 10.0.1.106 icmp_seq=85 ttl=127 time=0.41 ms
from 10.0.1.106 icmp_seq=86 ttl=127 time=0.41 ms
from 10.0.1.106 icmp_seq=87 ttl=127 time=0.42 ms
from 10.0.1.106 icmp_seq=88 ttl=127 time=0.73 ms
from 10.0.1.106 icmp_seq=89 ttl=127 time=0.41 ms
from 10.0.1.106 icmp_seq=90 ttl=127 time=0.41 ms
from 10.0.1.106 icmp_seq=91 ttl=127 time=0.78 ms
from 10.0.1.106 icmp_seq=92 ttl=127 time=0.09 ms
from 10.0.1.106 icmp_seq=93 ttl=127 time=0.00 ms
from 10.0.1.106 icmp_seq=94 ttl=127 time=0.00 ms
from 10.0.1.106 icmp_seq=95 ttl=127 time=0.01 ms
from 10.0.1.106 icmp_seq=96 ttl=127 time=0.78 ms
from 10.0.1.106 icmp_seq=97 ttl=127 time=0.01 ms
from 10.0.1.106 icmp_seq=98 ttl=127 time=0.536 ms
from 10.0.1.106 icmp_seq=99 ttl=127 time=0.43 ms
from 10.0.1.106 icmp_seq=100 ttl=127 time=0.452 ms
from 10.0.1.106 icmp_seq=101 ttl=127 time=0.00 ms
from 10.0.1.106 icmp_seq=102 ttl=127 time=0.90 ms
from 10.0.1.106 icmp_seq=103 ttl=127 time=0.20 ms
from 10.0.1.106 icmp_seq=104 ttl=127 time=0.04 ms
from 10.0.1.106 icmp_seq=105 ttl=127 time=0.962 ms
from 10.0.1.106 icmp_seq=106 ttl=127 time=0.19 ms
from 10.0.1.106 icmp_seq=107 ttl=127 time=0.15 ms
from 10.0.1.106 icmp_seq=108 ttl=127 time=0.15 ms
```



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 in my network ACL correctly pointed to my Public Subnet.

A screenshot of the AWS CloudWatch Metrics interface. The top navigation bar shows 'AWS' and 'Services'. The search bar contains 'Q. Search [Alt+S]'. On the right, it shows 'N. Virginia' and 'Kanika-IMM-Admin @ network-alias-kap'. The main area displays a log stream with the following content:

```
# bytes from 10.0.1.106: icmp_seq=69 ttl=127 time=1.47 ms
# bytes from 10.0.1.106: icmp_seq=70 ttl=127 time=1.44 ms
# bytes from 10.0.1.106: icmp_seq=71 ttl=127 time=1.09 ms
# bytes from 10.0.1.106: icmp_seq=72 ttl=127 time=1.28 ms
# bytes from 10.0.1.106: icmp_seq=73 ttl=127 time=1.23 ms
# bytes from 10.0.1.106: icmp_seq=74 ttl=127 time=1.23 ms
# bytes from 10.0.1.106: icmp_seq=75 ttl=127 time=1.17 ms
# bytes from 10.0.1.106: icmp_seq=76 ttl=127 time=1.70 ms
# bytes from 10.0.1.106: icmp_seq=77 ttl=127 time=1.44 ms
# bytes from 10.0.1.106: icmp_seq=78 ttl=127 time=1.99 ms
# bytes from 10.0.1.106: icmp_seq=79 ttl=127 time=1.09 ms
# bytes from 10.0.1.106: icmp_seq=80 ttl=127 time=1.95 ms
# bytes from 10.0.1.106: icmp_seq=81 ttl=127 time=1.44 ms
# bytes from 10.0.1.106: icmp_seq=82 ttl=127 time=1.41 ms
# bytes from 10.0.1.106: icmp_seq=83 ttl=127 time=2.16 ms
# bytes from 10.0.1.106: icmp_seq=84 ttl=127 time=1.03 ms
# bytes from 10.0.1.106: icmp_seq=85 ttl=127 time=1.44 ms
# bytes from 10.0.1.106: icmp_seq=86 ttl=127 time=1.42 ms
# bytes from 10.0.1.106: icmp_seq=87 ttl=127 time=1.71 ms
# bytes from 10.0.1.106: icmp_seq=88 ttl=127 time=1.48 ms
# bytes from 10.0.1.106: icmp_seq=89 ttl=127 time=1.44 ms
# bytes from 10.0.1.106: icmp_seq=90 ttl=127 time=1.78 ms
# bytes from 10.0.1.106: icmp_seq=91 ttl=127 time=1.77 ms
# bytes from 10.0.1.106: icmp_seq=92 ttl=127 time=1.00 ms
# bytes from 10.0.1.106: icmp_seq=93 ttl=127 time=1.44 ms
# bytes from 10.0.1.106: icmp_seq=94 ttl=127 time=1.45 ms
# bytes from 10.0.1.106: icmp_seq=95 ttl=127 time=1.01 ms
# bytes from 10.0.1.106: icmp_seq=96 ttl=127 time=1.78 ms
# bytes from 10.0.1.106: icmp_seq=97 ttl=127 time=1.05 ms
# bytes from 10.0.1.106: icmp_seq=98 ttl=127 time=1.936 ms
# bytes from 10.0.1.106: icmp_seq=99 ttl=127 time=1.43 ms
# bytes from 10.0.1.106: icmp_seq=100 ttl=127 time=0.652 ms
# bytes from 10.0.1.106: icmp_seq=101 ttl=127 time=1.44 ms
# bytes from 10.0.1.106: icmp_seq=102 ttl=127 time=1.39 ms
# bytes from 10.0.1.106: icmp_seq=103 ttl=127 time=1.20 ms
# bytes from 10.0.1.106: icmp_seq=104 ttl=127 time=1.76 ms
# bytes from 10.0.1.106: icmp_seq=105 ttl=127 time=0.962 ms
# bytes from 10.0.1.106: icmp_seq=106 ttl=127 time=1.13 ms
# bytes from 10.0.1.106: icmp_seq=107 ttl=127 time=1.15 ms
# bytes from 10.0.1.106: icmp_seq=108 ttl=127 time=1.15 ms
```

A circular profile picture of a young woman with long dark hair, wearing a light blue shirt, standing in front of a bookshelf.

Kanika Mathur
github.com/KanikaGenesis

NextWork.org

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 ACL, Route Tables and Security Groups were set up correctly.

Ping vs Curl

Ping and curl are different because they return different response to my 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 server!



Kanika Mathur
github.com/KanikaGenesis

NextWork.org

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.

```
src="https://js.sentry-cdn.com/149a6bc4cd616ff81bea862cf35e71eb.min.js"
data-lazy="no"
crossorigin="anonymous"
></script>

<project-app
  project="{id:id,title:title,category:category,storage:storage,concepts:concepts,description:description,difficulty:difficulty,icon:icon,needs:needs,track:track,order:order,shareTemplate:shareTemplate,public:public,status:status,incomplete:incomplete}"
  target="\u003ca href="#" data-lazy="no" crossorigin="anonymous">
  <p>Create your very own website on Amazon S3! It's easy peasy!</p>
  <p>AWS account - \u003ca href="#" data-lazy="no" crossorigin="anonymous">Create one</a>!<br/>
  <p>Just wrapped up a thrilling Amazon S3 project-challenge mode engaged!<br/>
  <p>Created and configured an Amazon S3 bucket, complete with ACLs, versioning, and public access.<br/>
  <p>Uploaded website content, diving deep into static websites function and how to host them on S3.<br/>
  <p>Tackled public access settings and fixed an interesting challenge with the website visibility.<br/>
  <p>My journey from creating buckets to deploying a fully functional static website in my documentation below.<br/>
  <p>Big thanks to NextWork for setting up this engaging challenge. Ready for the next one!<br/>
  <p>Leaves most of the tips, and keep improving!<br/>
  <p>Great for those looking for a challenge. Great for beginners!<br/>
  <p>Host a Website on Amazon S3<br/>
  <p>tracks#34;:[{description:#34;:#34;#1 - Low Touch#34;}, {description:#34;:#34;#2 - High Touch#34;}], visibility:#34;:#34;
  selectedTrack:"#34;
></project-app>

</body>
</html>
cc2-user@lin-10-0-0-218 ~15 □
```



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for more projects

