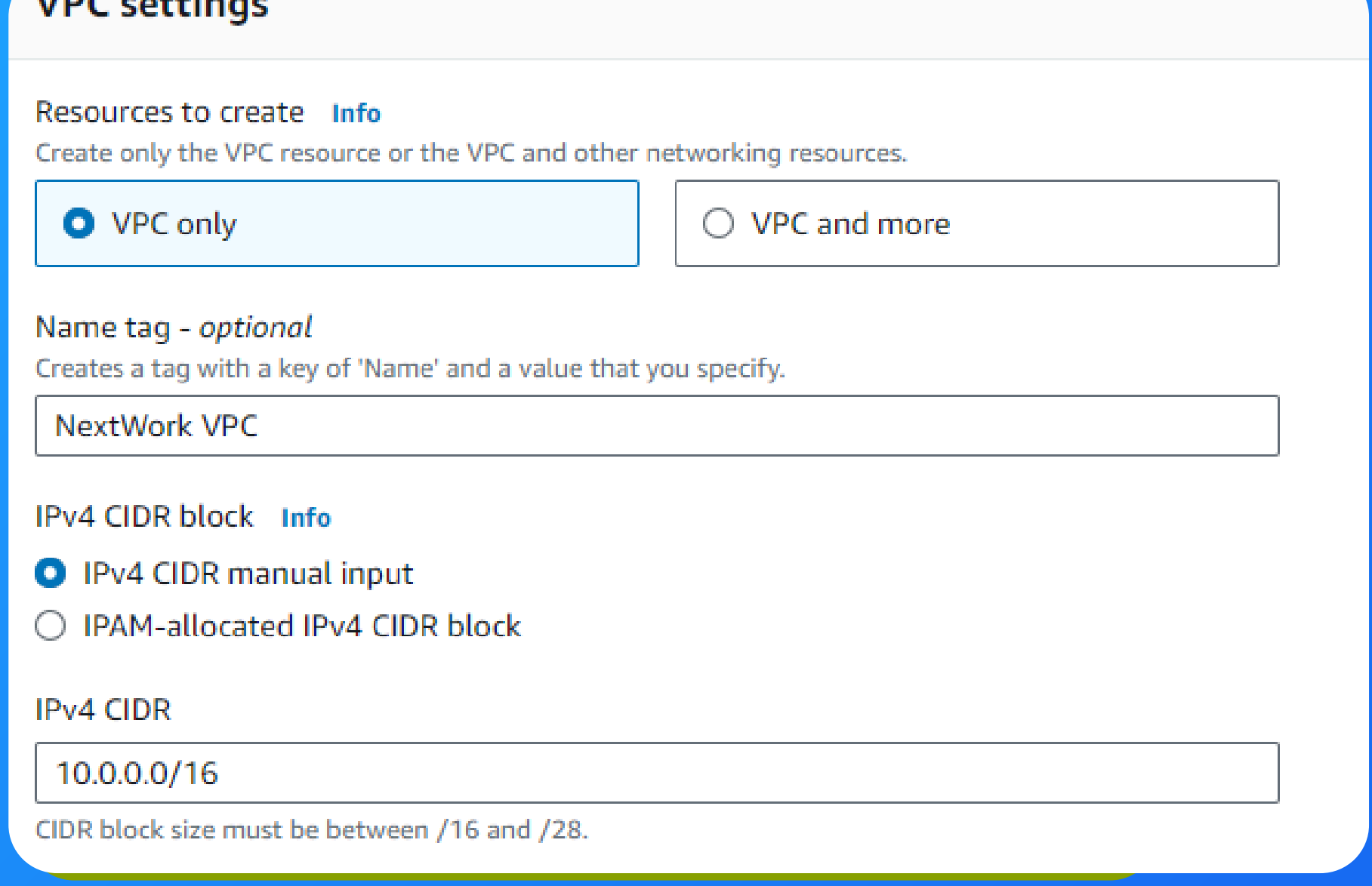
NextWork.org

# Building a Virtual Private Cloud


Kanika Mathur
github.com/KanikaGenesis


Kanika Mathur
github.com/KanikaGenesis

NextWork.org

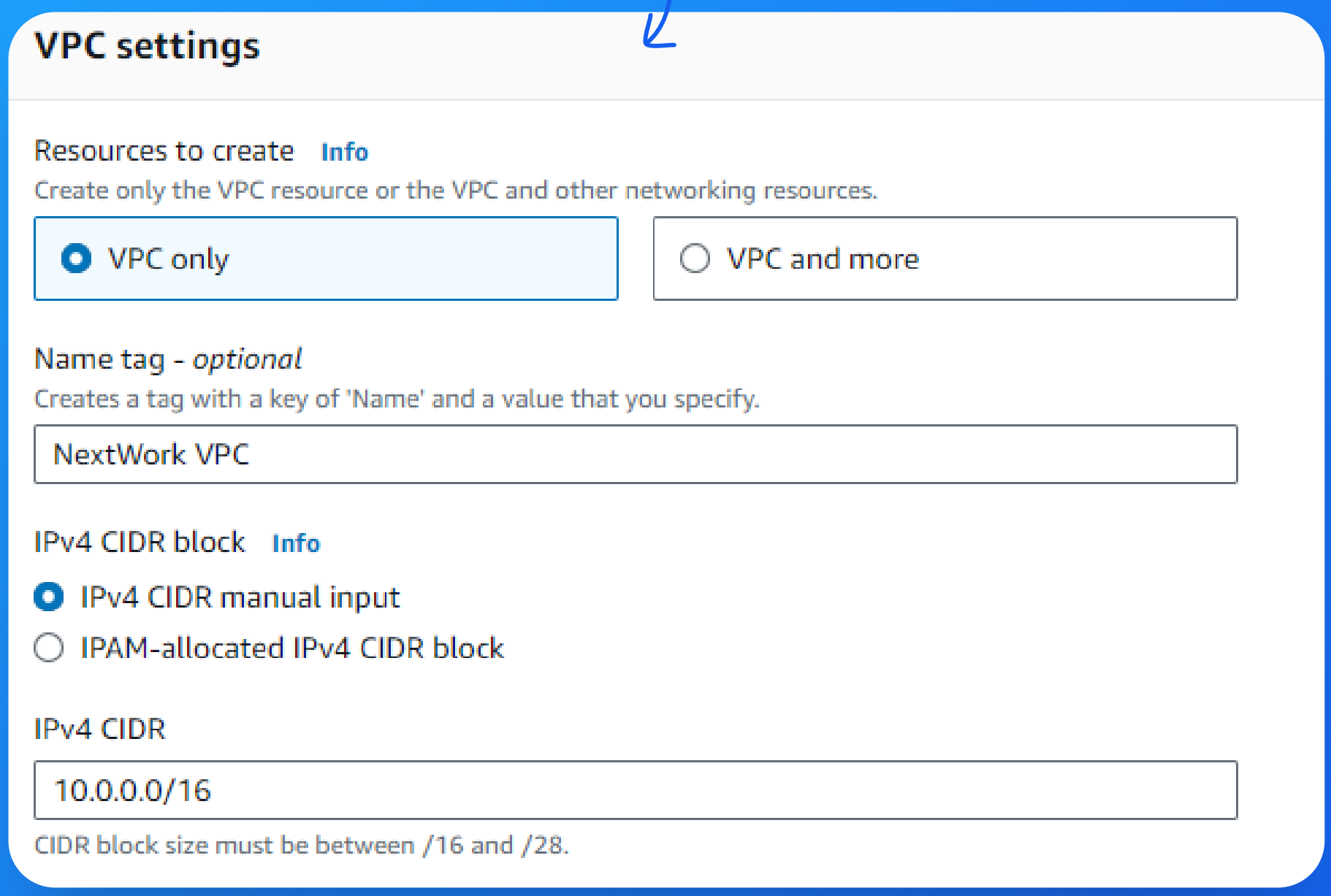
# My first VPC

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



Kanika Mathur
github.com/KanikaGenesis



# 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

Find resources by attribute or tag

| Name | Subnet ID | State | VPC |
|---|---|---|---|
| – | subnet-07ec14466cc3c1065 | Available | vpc-0d07efa809c |
| – | subnet-0f941aec08737ba28 | Available | vpc-0d07efa809c |
| – | subnet-050dcbf67389b5d05 | Available | vpc-0d07efa809c |
| – | subnet-0bf412ff3d8331cd5 | Available | vpc-0d07efa809c |
| – | subnet-0dbeb29b91e7216fa | Available | vpc-0d07efa809c |
| – | subnet-0e8aa553ee3421fb0 | Available | vpc-0d07efa809c |
| Public 1 | subnet-06424db0173e8d0ec | Available | vpc-00513cd1352 |

Kanika Mathur
github.com/KanikaGenesis

NextWork.org

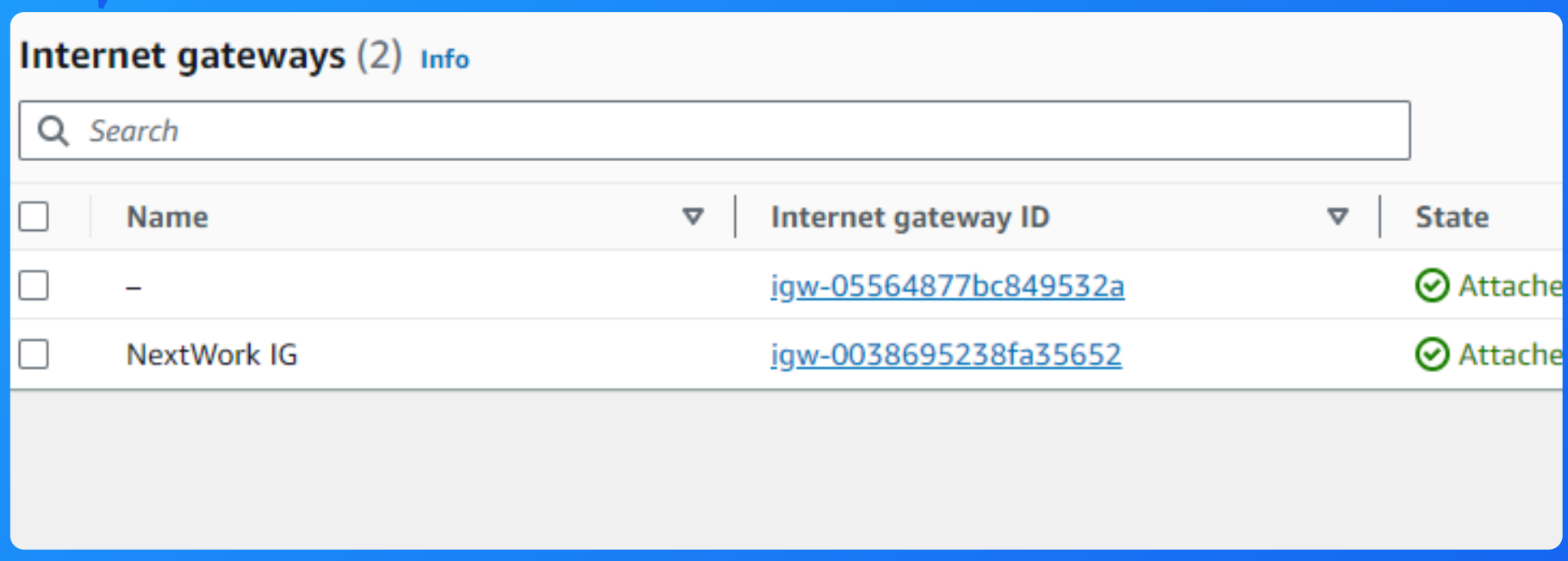
# Internet gateways

- 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 EC@ 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.

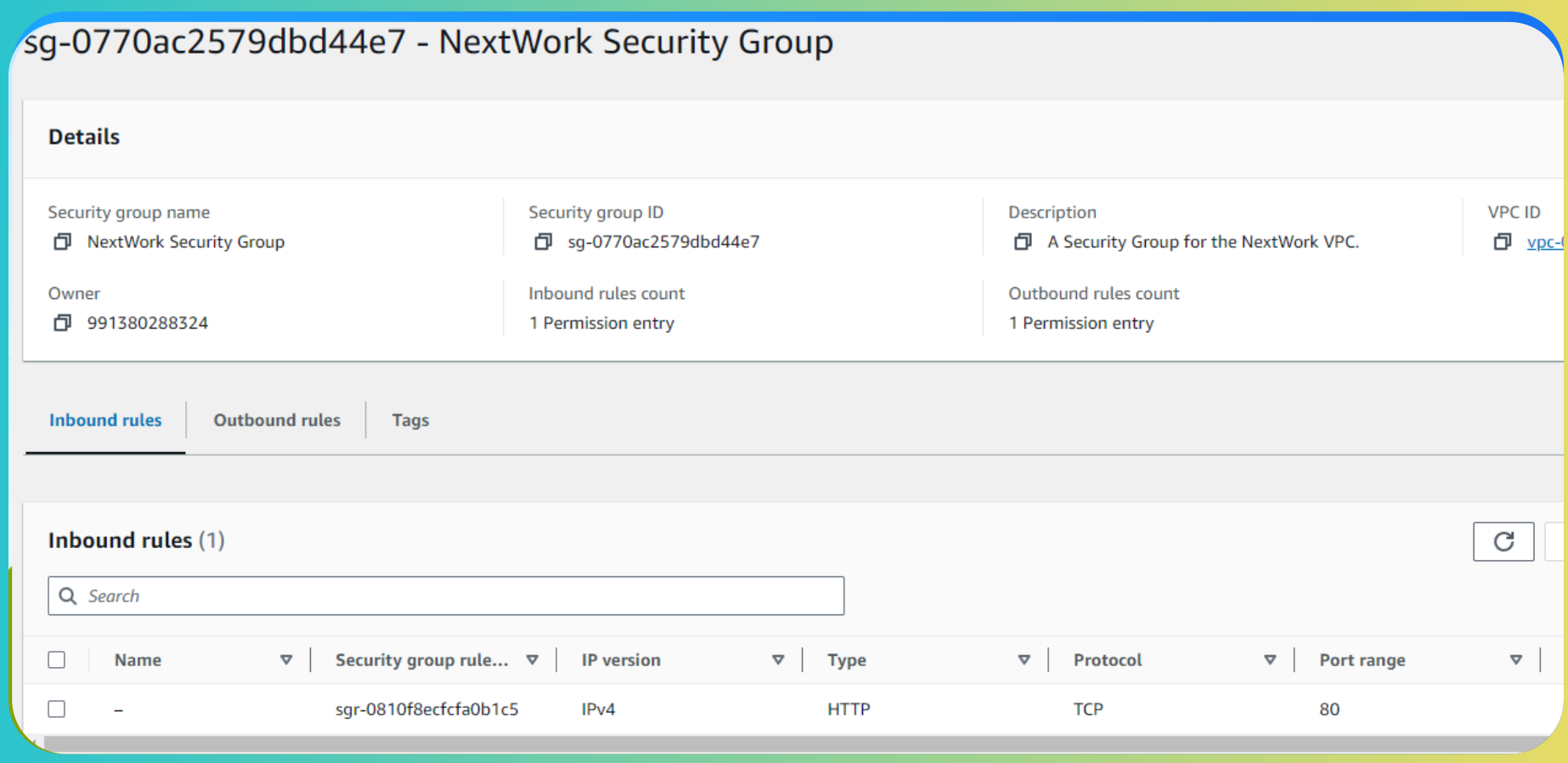
My created internet gateway!



NextWork.org

# VPC Traffic Flow and Security

Kanika Mathur
github.com/KanikaGenesis




Kanika Mathur
github.com/KanikaGenesis



# Route tables

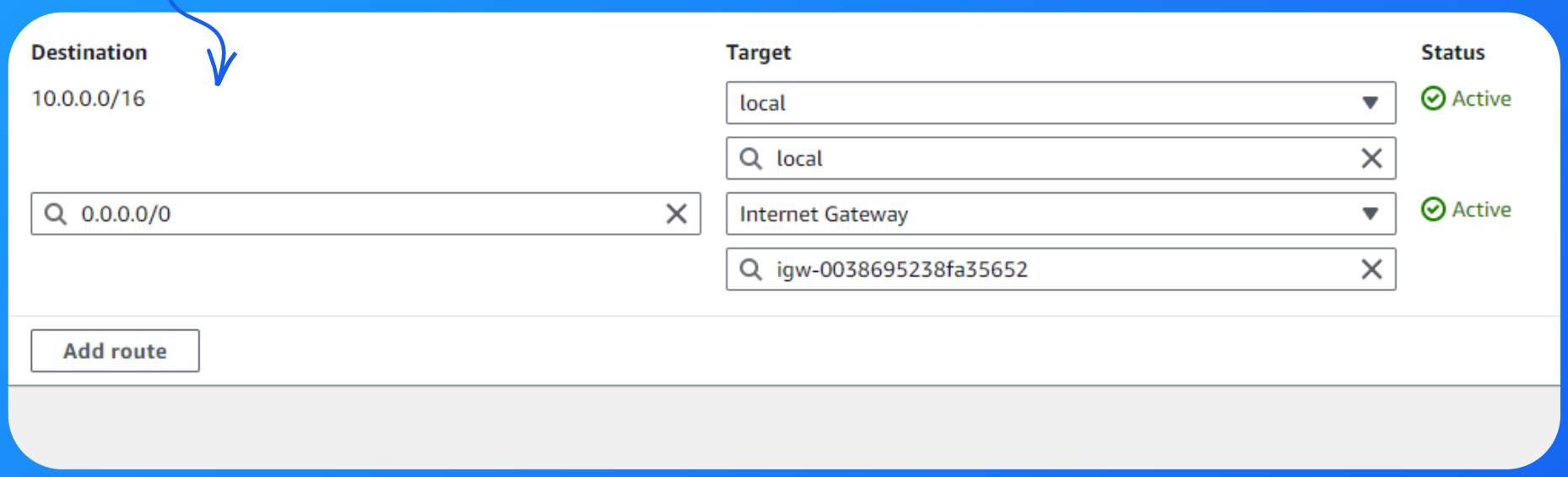
- Route tables are like the GPS that directs traffic within my VPC to the correct destination.

- 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




Kanika Mathur
github.com/KanikaGenesis



# Security groups

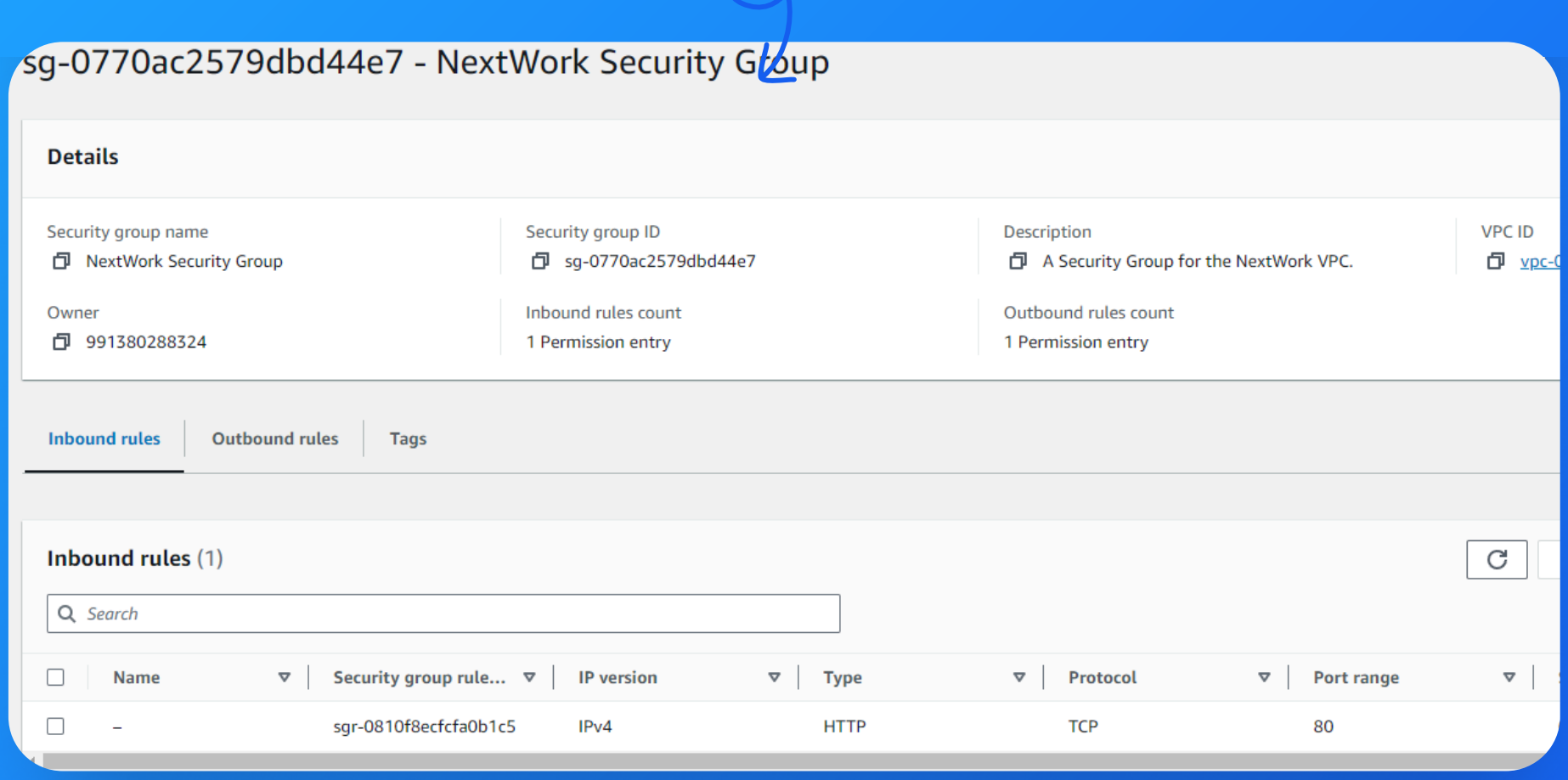
- Security groups are like security guards that monitor both inbound and outbound traffic **at the resource level, i.e. every single resource in a subnet/VPC has a security group.**

- 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




Kanika Mathur
github.com/KanikaGenesis



# 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)

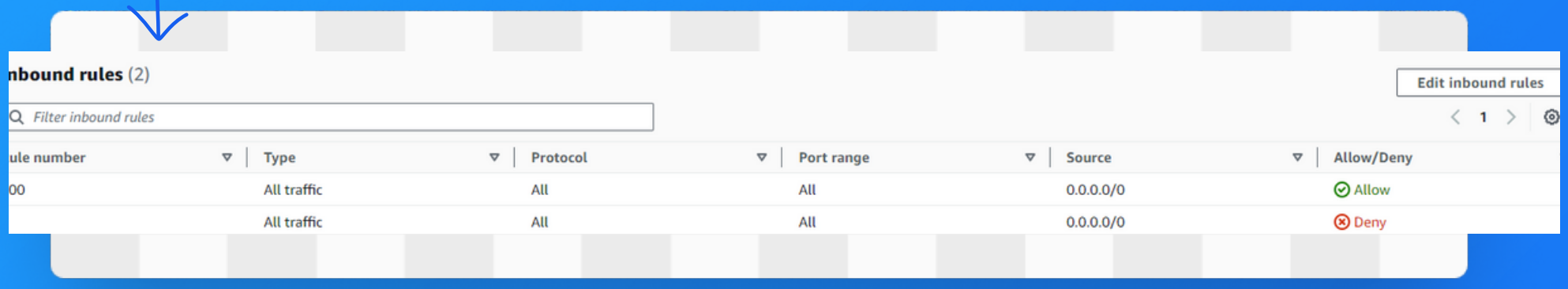
- Having both network ACLs and security groups is a good security best practice. It creates a dual layer of security that makes sure inbound/outbound traffic go through at least two checks.

- 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.

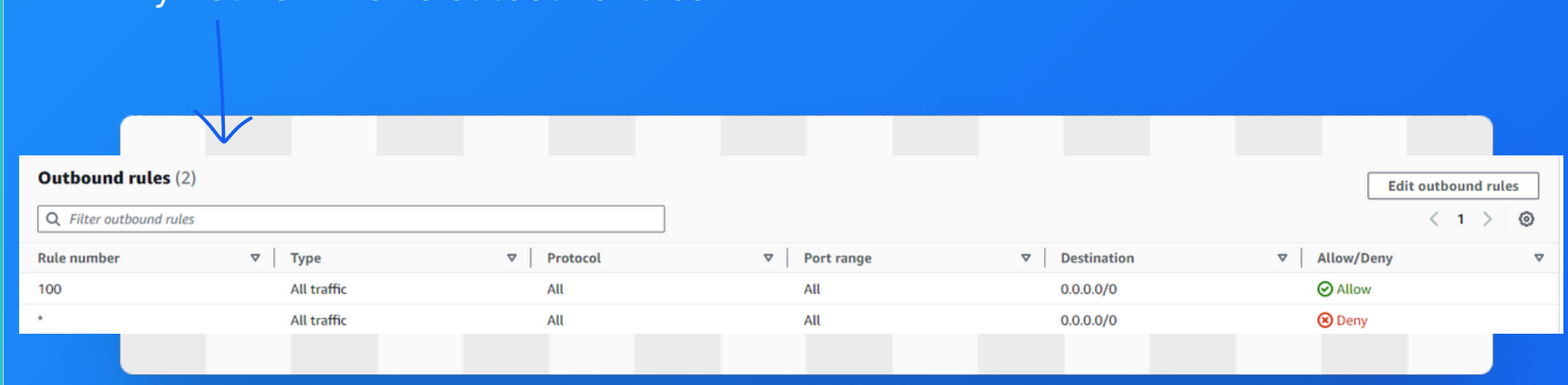
Kanika Mathur
github.com/KanikaGenesis

NextWork.org

My network ACL’s inbound rules



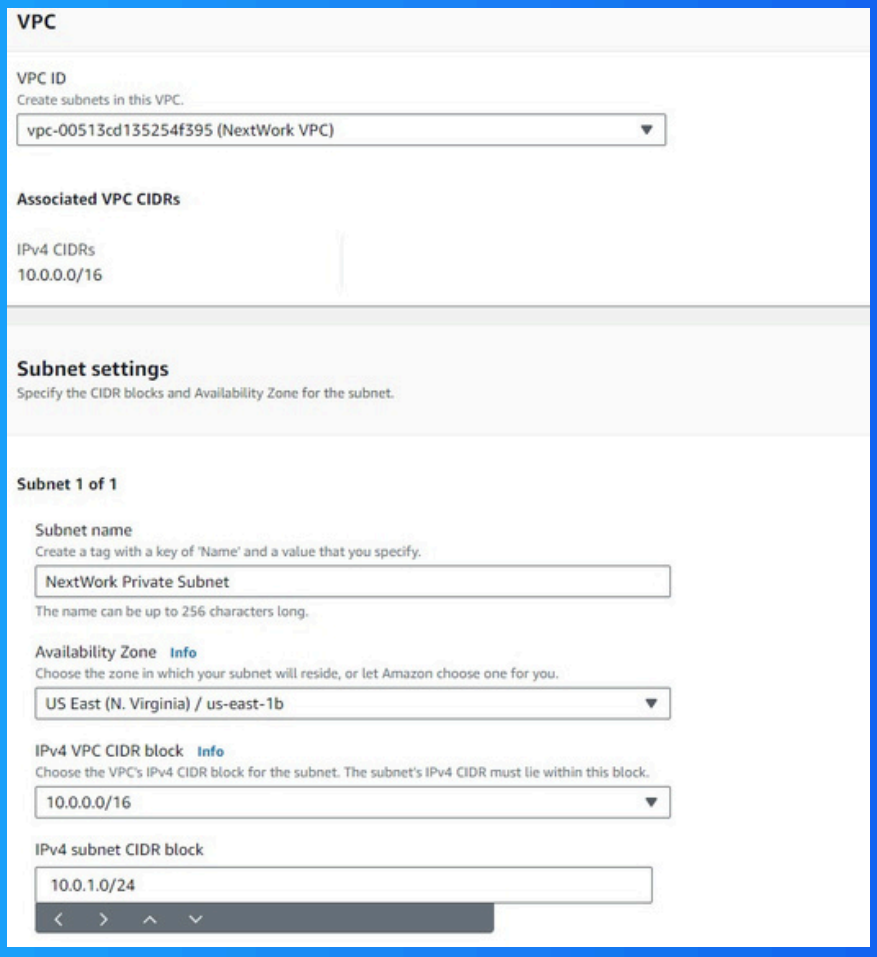

My network ACL’s outbound rules



NextWork.org

# Creating a Private Subnet

Kanika Mathur
github.com/KanikaGenesis



Kanika Mathur
github.com/KanikaGenesis

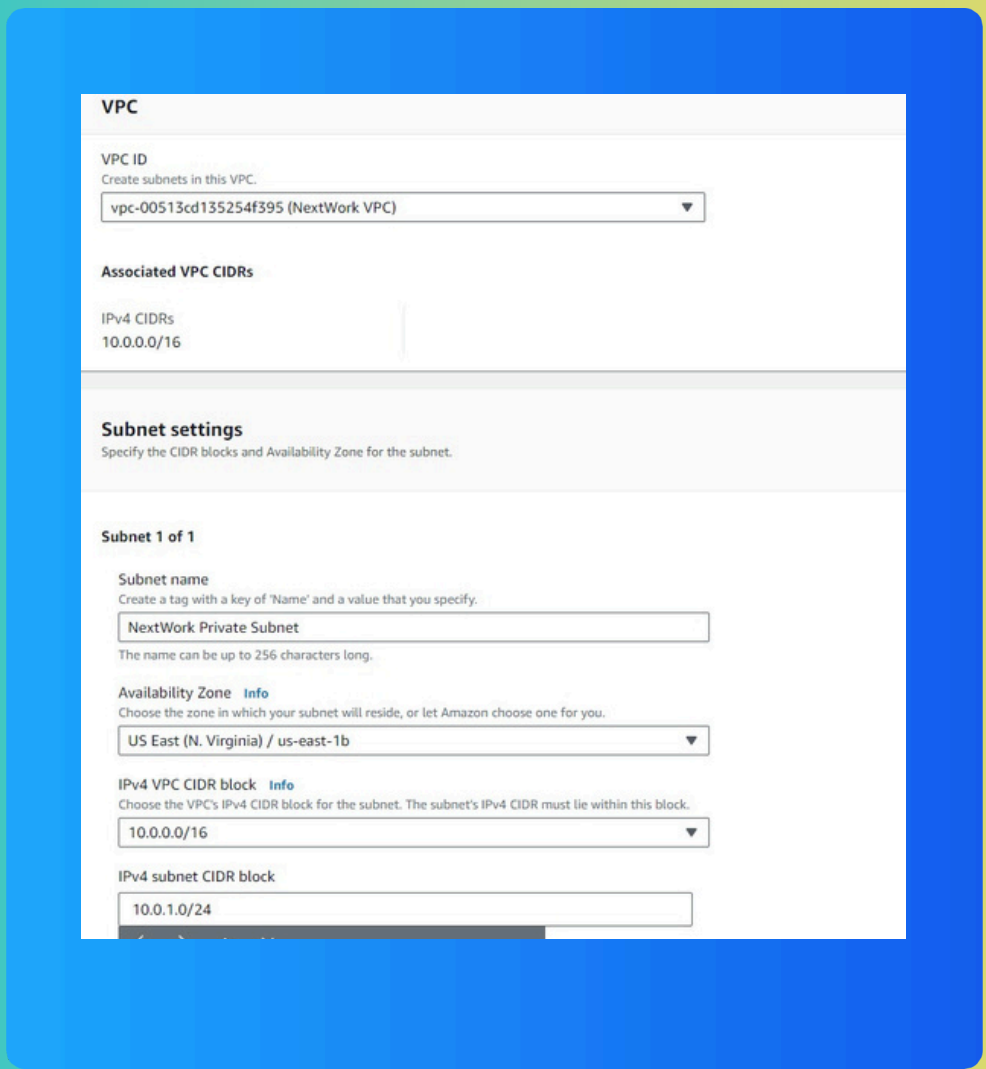
NextWork.org

# 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.



Kanika Mathur
github.com/KanikaGenesis

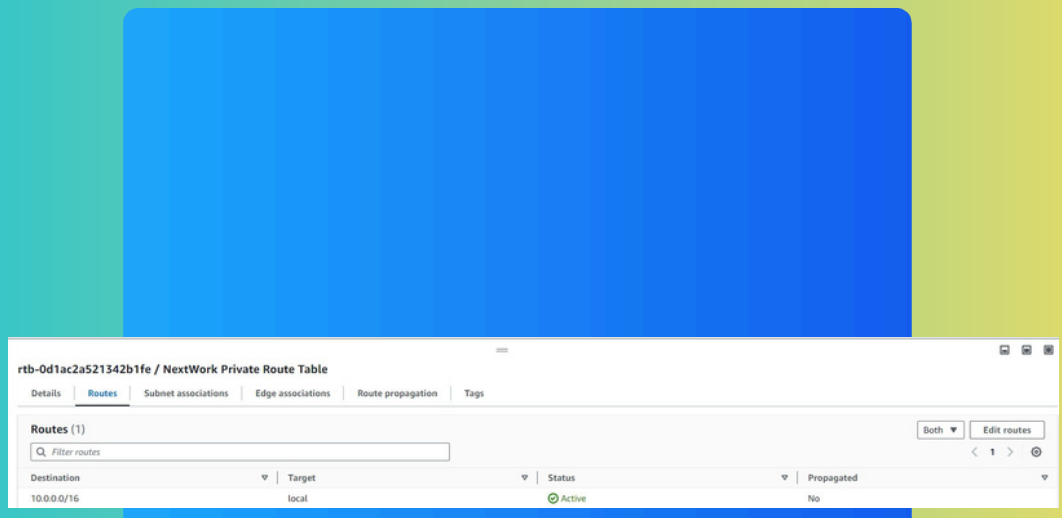
NextWork.org

# 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

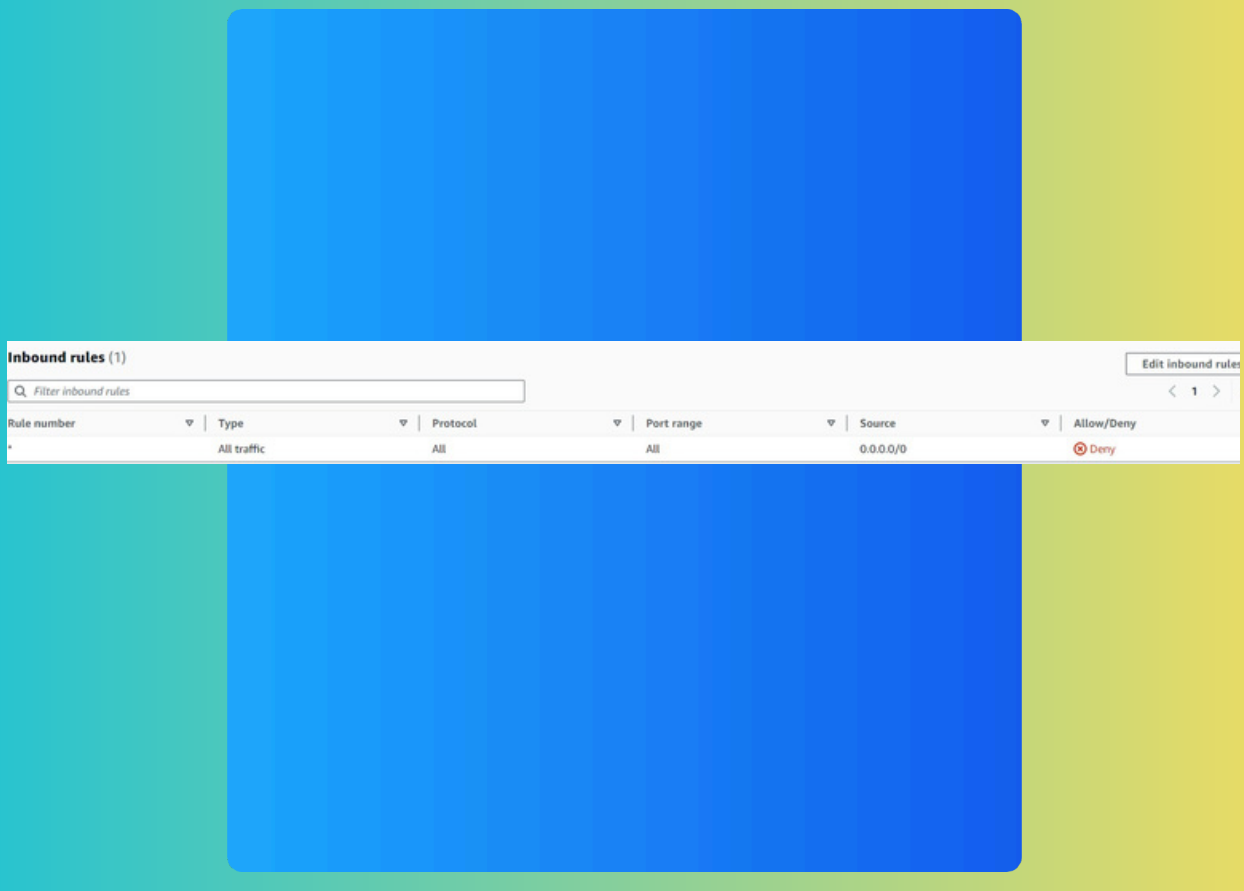
NextWork.org

# 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.

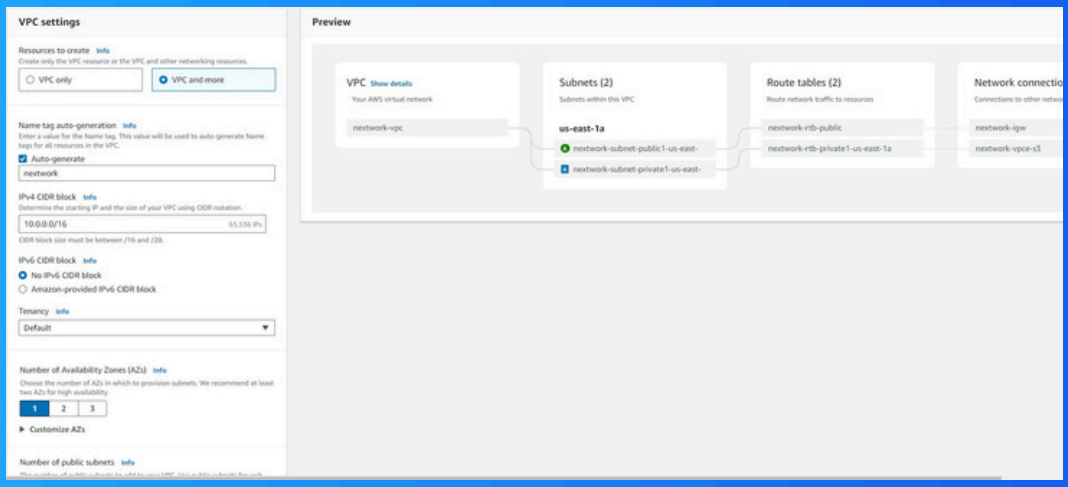
My new network ACL has two simple rules - deny all inbound and deny all outbound traffic.



NextWork.org

# Launching VPC Resources

Kanika Mathur
github.com/KanikaGenesis




Kanika Mathur
github.com/KanikaGenesis



# 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 Piblic security group.

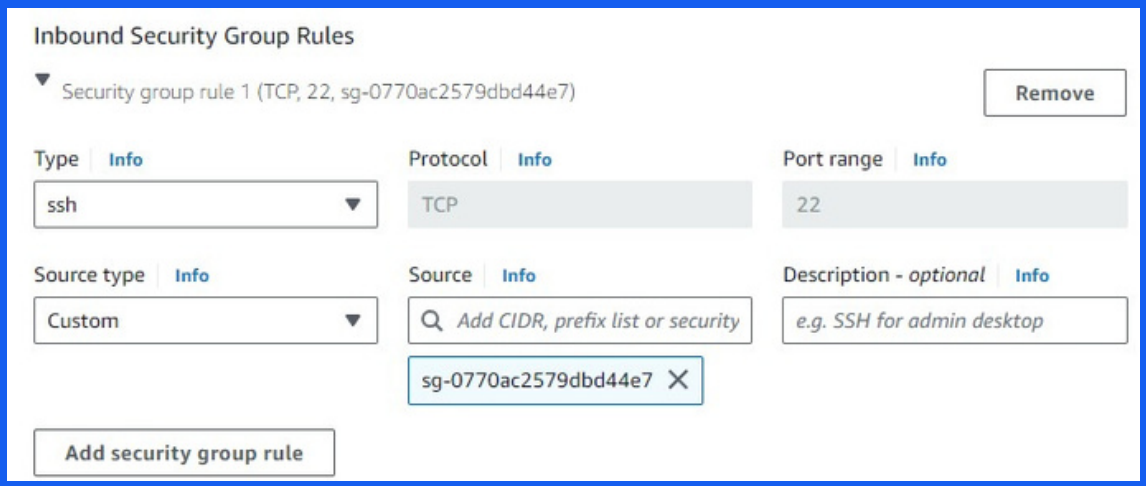
Kanika Mathur
github.com/KanikaGenesis

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.



Kanika Mathur
github.com/KanikaGenesis

NextWork.org

# 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.

Kanika Mathur
github.com/KanikaGenesis

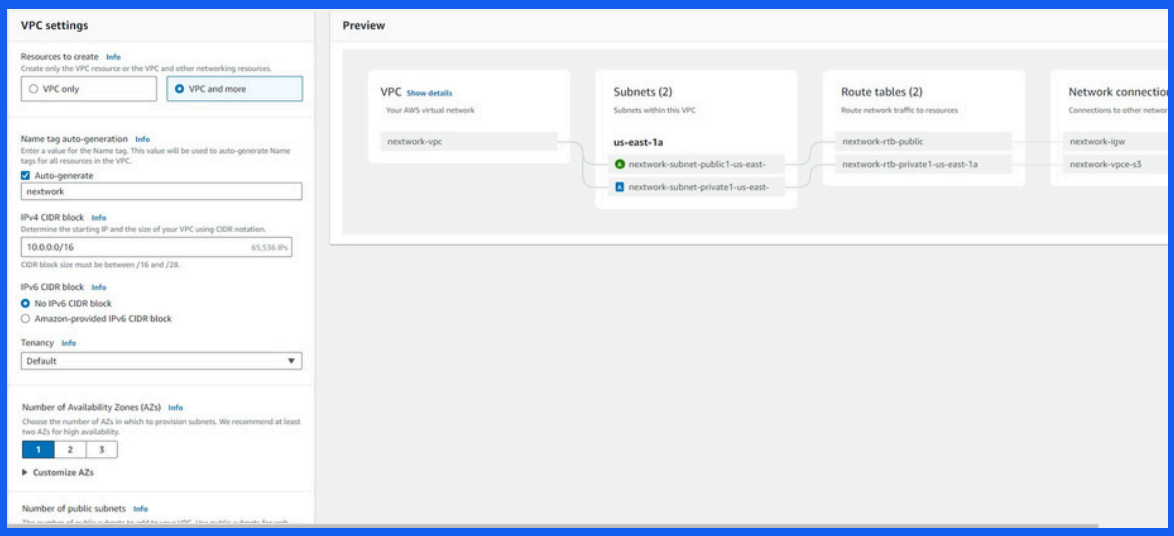
NextWork.org

# 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.

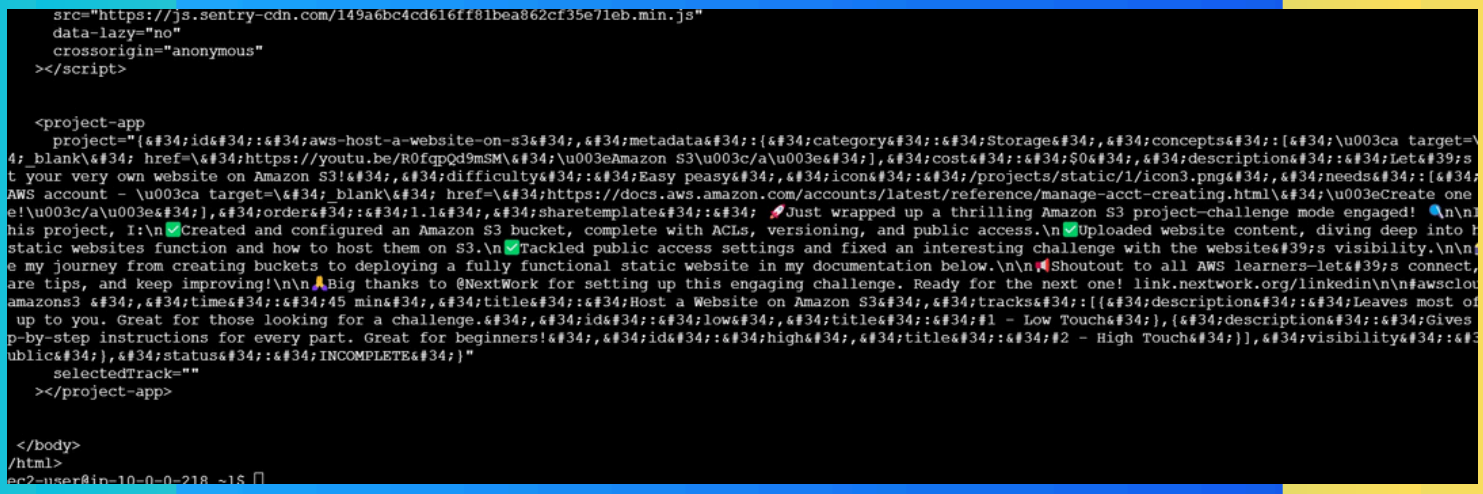
The setup page also offered to create NAT gateways, which are connectors/gateways that will let resources in private subnet get access to the internet (e.g. for security updates) while still blocking off incoming traffic from the internet.



NextWork.org

# Testing VPC Connectivity

Kanika Mathur
github.com/KanikaGenesis



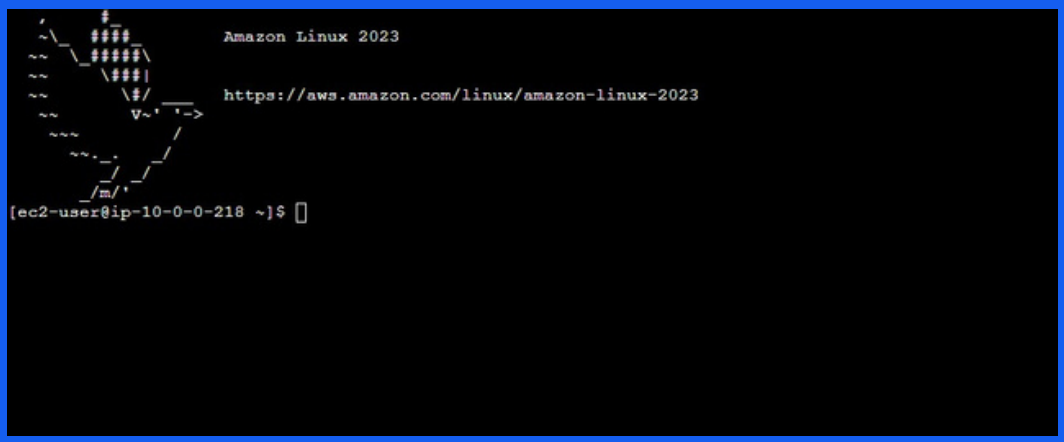
Kanika Mathur
github.com/KanikaGenesis

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 )



Kanika Mathur
github.com/KanikaGenesis

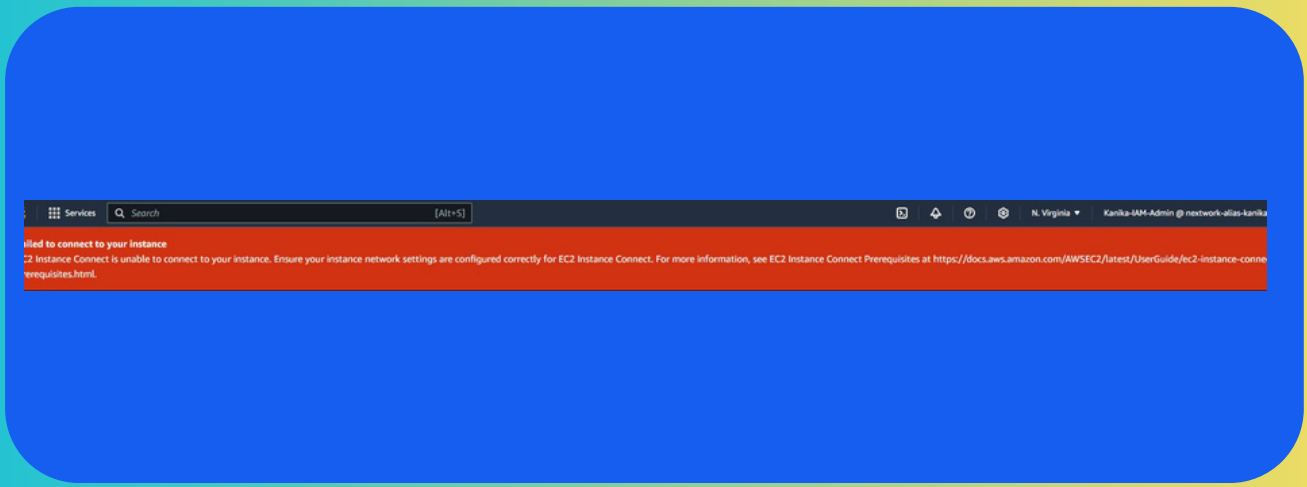
NextWork.org

# 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

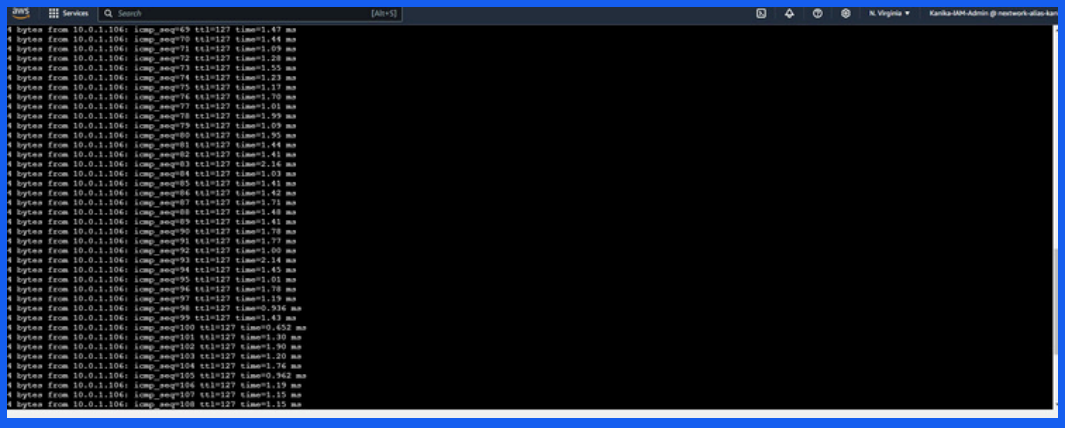


# Connectivity Between Servers

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 theprivate 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.

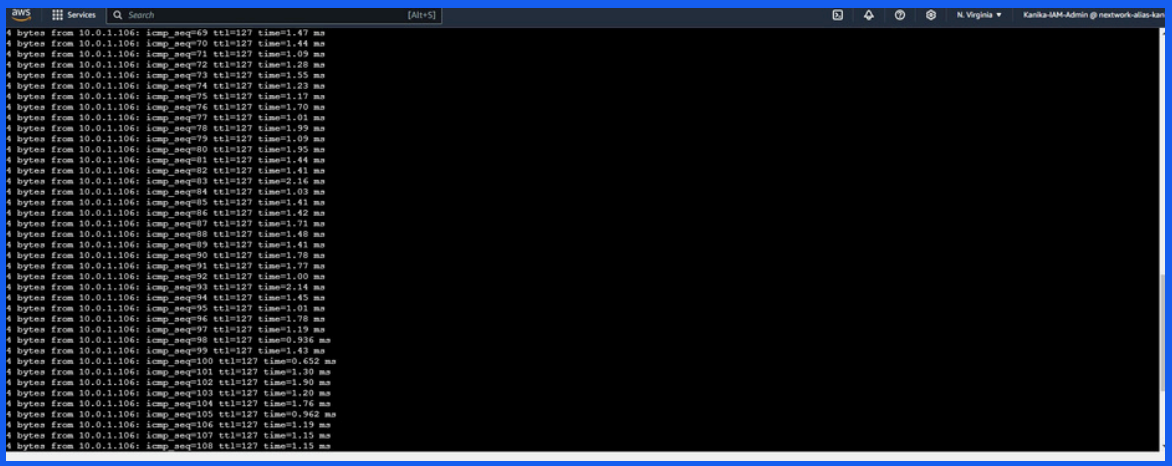


Kanika Mathur
github.com/KanikaGenesis

NextWork.org

# 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.




Kanika Mathur
github.com/KanikaGenesis



# 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

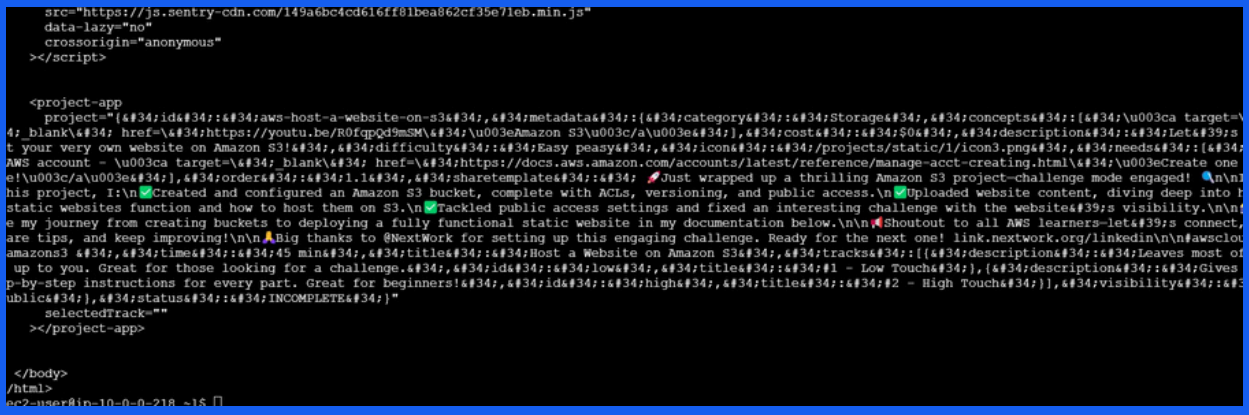
Ping and curl are different because they return different response to my Server's terminal. ping responds with a report on the performance of connectivity with my Private Server, curl responded with HTML data from another public server!

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.



NextWork.org

# Everyone should be in a job they love.

Check out nextwork.org for more projects