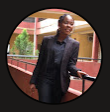
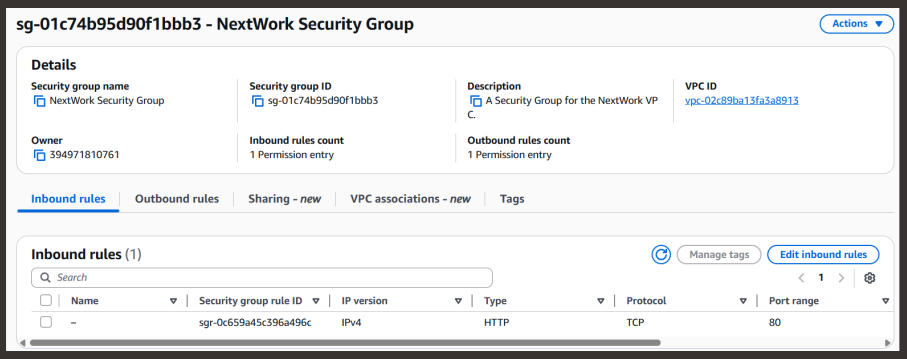
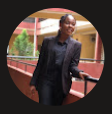
nextwork.org

VPC Traffic Flow and Security



Joba Amunigun
https://www.linkedin.com/in/dvoice




Joba Amunigun
NextWork Student

Introducing Today's Project!

What is Amazon VPC?

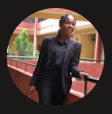
Amazon VPC (Virtual Private Cloud) is a secure, isolated section of the AWS cloud where I can launch and manage resources like EC2 instances, databases, and subnets. It’s useful because it gives me full control over my network — including IP ranges, routing, and access — so I can decide who can connect, how they connect, and what they can reach.

How I used Amazon VPC in this project

In today’s project, I used Amazon VPC to build and secure a cloud network environment from scratch. I created a VPC, added a subnet, set up route tables, configured internet access with an internet gateway, and secured resources using security groups and network ACLs. This hands-on setup helped me understand how cloud traffic flows and how to control who can access what — a critical part of designing secure and reliable cloud systems.

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

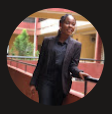
One thing I didn’t expect was how tricky it can be to properly delete resources using the AWS CLI in CloudShell. I assumed removing resources would be quick, but I had to carefully follow the right order — like detaching gateways before deleting a VPC — or I'd run into errors.



Joba Amunigun
NextWork Student

This project took me...

This project took me about 90 minutes — including time spent working through the CLI steps, exploring EC2 Global View, and troubleshooting the resource deletion process. I gave myself room to slow down and understand how each part of the VPC setup connects — not just to build, but to build with clarity.

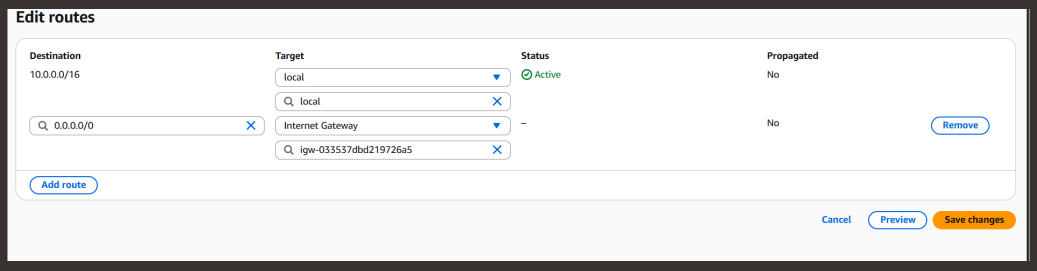


Joba Amunigun
NextWork Student

Route tables

Route tables are like GPS systems for your subnet — they tell traffic where to go. Each route table contains a set of rules (called routes) that decide how data moves in and out of a subnet. For example, if I want an EC2 instance to receive traffic from the internet, its subnet must be linked to a route table that knows how to direct that traffic. Without a route table, even a properly set up instance won't know where to send or receive data — it's like having a phone with no signal path.

Route tables are needed to make a subnet public because they tell the subnet how to reach the internet. For a subnet to be considered public, its route table must have a route that sends all internet-bound traffic (0.0.0.0/0) to an internet gateway. Without that route, even if the subnet has an internet gateway attached, it won't know how to reach the outside world — and the internet won't be able to reach it either

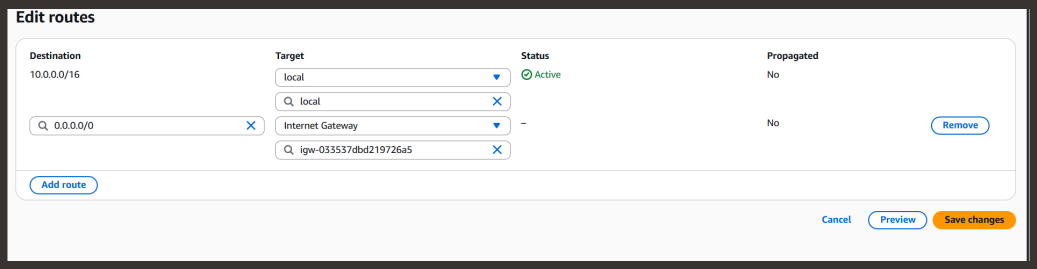


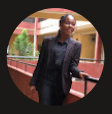
Joba Amunigun
NextWork Student

Route destination and target

Routes are defined by their destination and target, which mean where the traffic is going and how it should get there. The destination is the range of IP addresses the traffic is trying to reach (e.g. 0.0.0.0/0 means “anywhere on the internet”). The target is the next step in the path — like an internet gateway, NAT gateway, or another network interface — that the traffic should be sent through to reach that destination. It’s like saying, “To reach this address (destination), go through this road (target).”

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 igw-033537dbd219726a5. This means any traffic heading to the internet from my subnet will go through the specified internet gateway, making the subnet public and accessible from outside the VPC. In short: all outgoing traffic (destination) uses this internet gateway (target) to get there.




Joba Amunigun
NextWork Student

Security groups

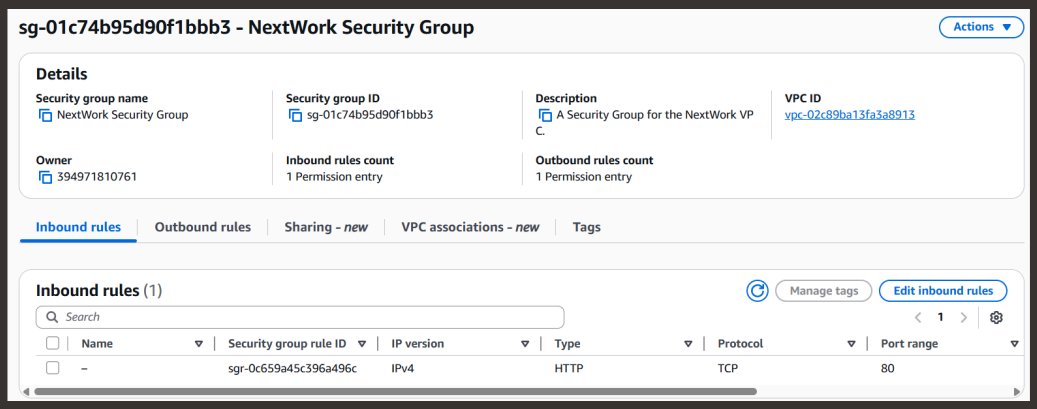
Security groups are like security checkpoints for individual cloud resources. They control what kind of traffic is allowed in or out of a specific resource — like an EC2 instance — based on things like IP address, protocol, and port number. Unlike subnets or VPCs, security groups don't attach to whole neighborhoods — they're assigned to each resource directly, acting like the personal guard at the door. If the VPC is the city, and the subnet is the neighborhood, the security group is the guard at your building's front door — deciding who can visit, and who gets turned away.

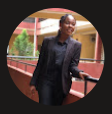
Inbound vs Outbound rules

Inbound rules are the settings that control which types of data or traffic are allowed to enter a resource — like letting users visit a website hosted on an EC2 instance. I configured an inbound rule that allows traffic from anywhere (0.0.0.0/0) on port 80, which is the standard port for HTTP web traffic. This ensures that users can access the website I'm hosting from any browser, globally. Think of it like opening the front door for guests — but only if they knock on the correct door (port).

Outbound rules are the settings that control what kind of traffic your resource is allowed to send out to the internet or other services. By default, my security group's outbound rule allows all traffic to all destinations, which means my EC2 instance can freely connect to the internet — So unless i specify otherwise, any resource associated with the security group can access and send data to any IP address - whether it's in my VPC, other VPCs (if you have the right permissions) and on the public internet! It's like letting someone inside your building place calls or send messages without restriction.

Joba Amunigun
NextWork Student



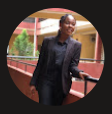

Joba Amunigun
NextWork Student

Network ACLs

Network ACLs are like traffic cops stationed at the borders of a subnet — inspecting every piece of data (called data packets) that tries to enter or exit. They apply broad security rules to all resources inside a subnet. For example, I can use a Network ACL to block an entire range of IP addresses or restrict certain ports from sending or receiving traffic. While security groups protect individual buildings, Network ACLs guard the entire neighborhood — giving my VPC an extra layer of security.

Security groups vs. network ACLs

The difference between a security group and a network ACL is that a security group controls traffic at the resource level, while a network ACL manages traffic at the subnet level.- Network ACLs are used to set broad traffic rules that apply to an entire subnet, while allow for more granular control, managing access to individual resource Security groups are like guards at the door of each EC2 instance — they control access to that specific resource. Network ACLs, on the other hand, act like checkpoint officers at the entrance and exit of an entire subnet — checking all traffic going in or out. Using both creates layered security: broad rules for the subnet with ACLs, and detailed rules for each resource with security groups.


Joba Amunigun
NextWork Student

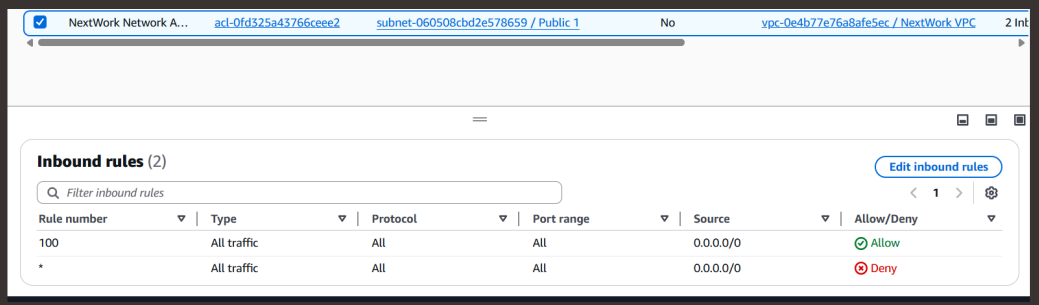
Default vs Custom Network ACLs

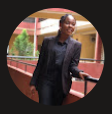
Similar to security groups, network ACLs use inbound and outbound rules

By default, a network ACL's inbound and outbound rules will use inbound and outbound rules to decide which data packets are allowed to enter or leave subnets: For this project -Rule 100 Inbound allows all inbound traffic into the Public Subnet. - Rule 100 Outbound allows all traffic out of the Public Subnet. - The second line in each ruleset shows an asterisk (*) that acts as a catch-all rule in case traffic does not match any of the earlier rules. This means default network ACLs allow all inbound and outbound traffic, unless customized.

In contrast, a custom ACL’s inbound and outbound rules are automatically set to deny all traffic until specific rules are added to allow certain types of traffic in or out. This means nothing can enter or leave the subnet unless I explicitly permit it through custom rule entries — giving me full control over what’s allowed. It’s like setting up a new checkpoint where nobody gets in or out unless their name is on the list.

Joba Amunigun
NextWork Student



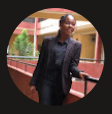

Joba Amunigun
NextWork Student

Tracking VPC Resources

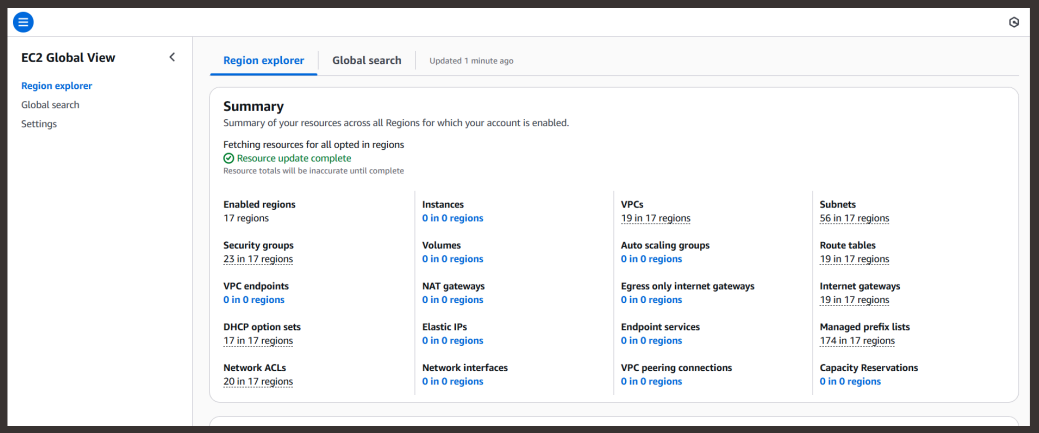
I created additional resources using the AWS CLI — a VPC, Internet Gateway, and Security Group — but this time in a different AWS region (us-east-1) instead of my default region. Teams use multiple regions to: – Improve latency for end users by hosting resources closer to their physical location – Increase resilience — if one region faces an outage or disruption, others can keep the system running

EC2 Global View is a tool where you can find all your EC2 resources across every AWS region in one place — including VPCs, instances, subnets, security groups, and more. I could even narrow down my search by region, resource type, or account, which made it easier to track exactly where everything was deployed. Without EC2 Global View, you'd have to switch between regions manually in the AWS Console just to see what you've created — which can be confusing and time-consuming.

Now that I've learnt about EC2 Global View, I'd use it again to keep track of all my resources across different AWS regions — especially when managing multi-region deployments. It's a great way to quickly audit what exists, avoid duplication, and stay organized without switching regions manually every time.



Joba Amunigun
NextWork Student



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects