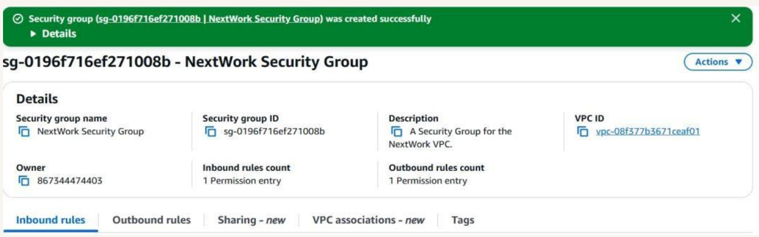
VPC Traffic Flow and Security

Anas Oubassou



Introducing Today's Project!

What is Amazon VPC?

Amazon VPC creates a private network environment within AWS where you can launch resources like EC2 instances with complete control. I find it particularly valuable because it allows you to manage network traffic patterns effectively, implement strong security through customizable rules, and organize your infrastructure using subnets. This organization provides better scalability as your applications grow and maintains proper isolation between different parts of your application.

How I used Amazon VPC in this project

What surprised me during this implementation was discovering that custom network ACLs deny all traffic by default, which contrasts with default ACLs that allow all traffic. This meant I needed to explicitly add rules to permit the traffic I wanted, rather than just blocking unwanted traffic. This experience highlighted how important it is to thoroughly understand security defaults when setting up VPC components, as these subtle differences can significantly impact your network's accessibility and security posture.

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

I was surprised to discover that custom network ACLs deny all traffic by default, unlike default ACLs that allow all traffic. This necessitated adding explicit rules to permit traffic, highlighting the importance of careful configuration when setting up VPC security components

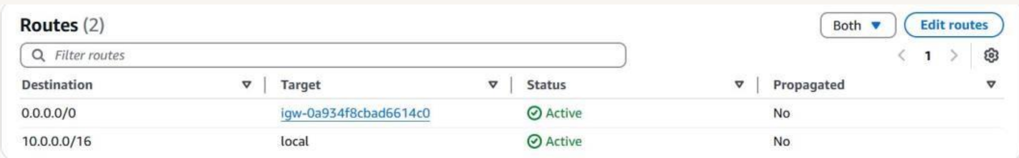
This project took me...

The project required approximately 60 minutes to complete from start to finish. During this time, I worked through several key phases: creating the VPC infrastructure, setting up and configuring the route table for proper traffic flow, implementing the security group with appropriate access rules, and finally adding the network ACL layer for additional protection. Throughout each step, I followed the instructions carefully while taking time to verify that each component was functioning correctly before moving forward.

Route tables

Route tables are essential components that control network traffic flow within a VPC. They consist of a collection of rules, called routes, that determine where data packets should travel based on their destination IP address range. Each route specifies a target—the gateway or connection through which traffic should flow, such as an internet gateway for external traffic or the local VPC infrastructure for internal communications.

Route tables play a crucial role in making a subnet public. To enable internet access, a subnet's route table must include a specific route that directs all internet-bound traffic (identified by the destination CIDR block 0.0.0.0/0) to an internet gateway. Without this configuration, resources in the subnet would remain isolated from the internet regardless of other settings, effectively keeping them private.



Routes (2)

Both | Edit routes

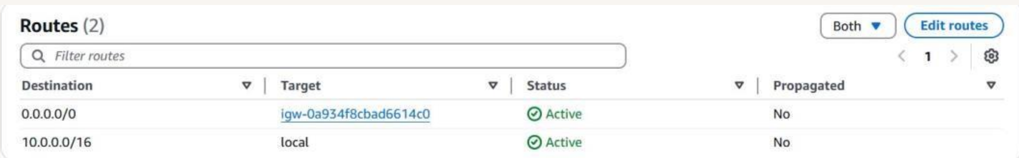
Filter routes

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0a934f8cbad6614c0	Active	No
10.0.0.0/16	local	Active	No

Anas Oubassou
EMI Engineering Student

Route destination and target

When configuring routes, understanding the relationship between destination and target is fundamental. The destination identifies which IP address range the route applies to—essentially defining "where is this traffic trying to go?" Meanwhile, the target specifies the AWS resource that will handle traffic matching that destination—answering "how should this traffic get there?"
In my project implementation, I configured an important route with a destination of 0.0.0.0/0, which represents all possible IPv4 addresses anywhere on the internet. I set the target for this route to my NextWork internet gateway (displayed as NextWork IG in the AWS console). This configuration meant that any traffic from my subnet destined for any external IP address would be directed through my internet gateway, enabling communication with the wider internet while maintaining VPC security.



Routes (2)

Both | Edit routes

Filter routes

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0a934f8cbad6614c0	Active	No
10.0.0.0/16	local	Active	No

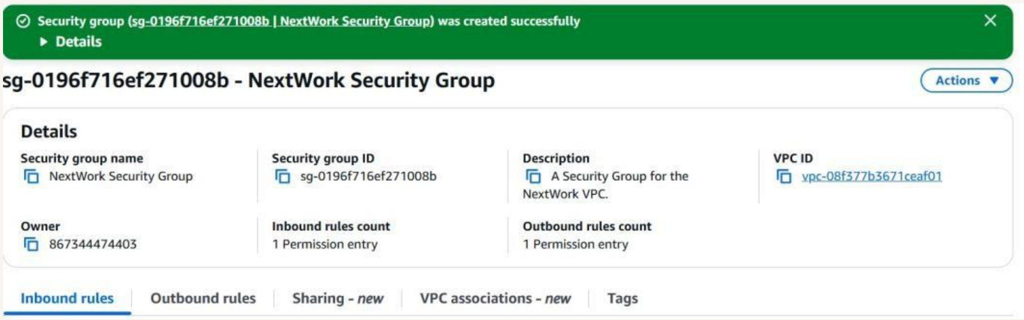
Security groups

Security groups function as virtual firewalls that protect individual resources within your VPC. They control both incoming and outgoing traffic through rule-based filtering that evaluates IP addresses, protocols (such as HTTP or SSH), and port numbers. Unlike traditional firewalls that often filter solely based on network attributes, security groups in AWS provide more granular control at the resource level, allowing different resources within the same subnet to have different security policies.

Inbound vs Outbound rules

Inbound rules govern which external traffic can reach your protected resources. For my project, I configured an inbound rule that allowed HTTP traffic (port 80) from any source IP address (0.0.0.0/0). This configuration enabled public internet users to access web resources hosted in my NextWork VPC—a typical requirement for public-facing web applications.

Outbound rules determine how your resources can communicate with the external world. In my security group configuration, I maintained the default outbound rule that permits all outbound traffic to any destination. This allows resources within the security group to freely initiate connections to external services, which is often necessary for software updates, API calls, and other outbound communications.


Security group (sg-0196f716ef271008b | NextWork Security Group) was created successfully
Details
sg-0196f716ef271008b - NextWork Security Group
Actions
Details
Security group name
NextWork Security Group
Security group ID
sg-0196f716ef271008b
Description
A Security Group for the NextWork VPC.
VPC ID
vpc-08f377b3671ceaf01
Owner
867344474403
Inbound rules count
1 Permission entry
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Sharing - new
VPC associations - new
Tags

Network ACLs

Network ACLs provide an additional security layer that operates at the subnet level. Unlike sec groups that protect specific resources, network ACLs filter all traffic entering or exiting an entir They examine data packets against ordered rules to determine whether to allow or deny acces on IP ranges, protocols, and ports. Their stateless nature means they evaluate inbound and ou traffic separately, requiring explicit rules for both directions.

Security groups vs. network ACLs

While security groups and network ACLs both control traffic flow, they differ in several importa Security groups apply to individual resources and are stateful—remembering previous connecti automatically allowing return traffic. Network ACLs, however, apply broadly to all resources in and are stateless—treating each packet independently without tracking connection state. This fundamental difference means security groups provide more granular protection for specific re while network ACLs offer broader traffic filtering at the subnet boundary.

Default vs Custom Network ACLs

Network ACLs, like security groups, use both inbound and outbound rules to control traffic. De network ACLs come with a permissive stance—their inbound and outbound rules allow all traffi any IP address. Typically, this is implemented through a rule (often numbered as Rule 100) that all protocols and ports, with a catch-all (*) rule at the end to deny any traffic not explicitly matc Custom network ACLs, however, take a more restrictive approach. When you create a custom A inbound and outbound rules are automatically set to deny all traffic by default. This requires yo explicitly define rules (such as allowing traffic from 0.0.0.0/0) to permit the specific data packe want to allow through. This "deny by default" approach aligns with security best practices but more careful planning to ensure necessary traffic isn't accidentally blocked.

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	Custom TCP	TCP (6)	0	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny