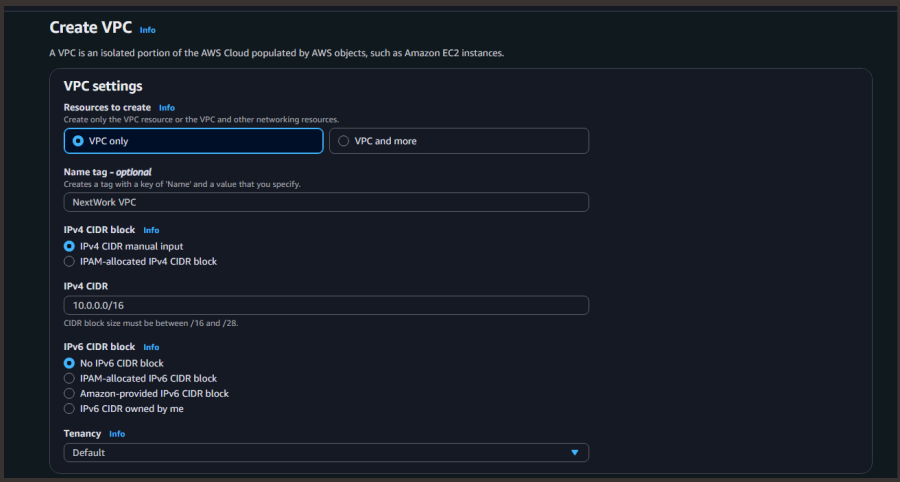
nextwork.org

Build a Virtual Private Cloud (VPC)

Louis



Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is a virtual private cloud service that lets you create an isolated, logically separated network within AWS where you can launch and manage your cloud resources with complete control over IP addressing, subnets, routing, and security settings, and it is useful because it provides secure network isolation for your applications, allows you to control exactly who and what can access your resources, and enables you to design custom network architectures that meet specific security and compliance requirements while connecting securely to your on-premises infrastructure if needed.

How I used Amazon VPC in this project

In today's project, I used Amazon VPC to create a custom isolated network environment by exploring both the AWS Console and AWS CloudShell. I started by using the Console's graphical interface to understand how VPC resources are structured and connected, then switched to CloudShell to create the same infrastructure using CLI commands. I built the foundational networking infrastructure by creating a VPC with a defined CIDR block, adding a subnet within that VPC to segment the network space, and attaching an Internet Gateway to enable communication between resources in my VPC and the internet. This hands-on experience taught me how cloud networking components work together and gave me exposure to both GUI-based and command-line approaches to provisioning infrastructure.

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

One thing I didn't expect in this project was how specific and precise the CLI commands needed to be. I thought creating a subnet would be straightforward, but the error message taught me that AWS requires exact parameters like the CIDR block, and missing even one required parameter prevents the command from executing. This showed me that while CLI is powerful, it requires more attention to detail than the Console.

This project took me...

This project took me approximately 1-2 hours to complete, including time to explore the AWS Console interface, learn CloudShell commands, troubleshoot the missing CIDR block error, and document my process and learnings.

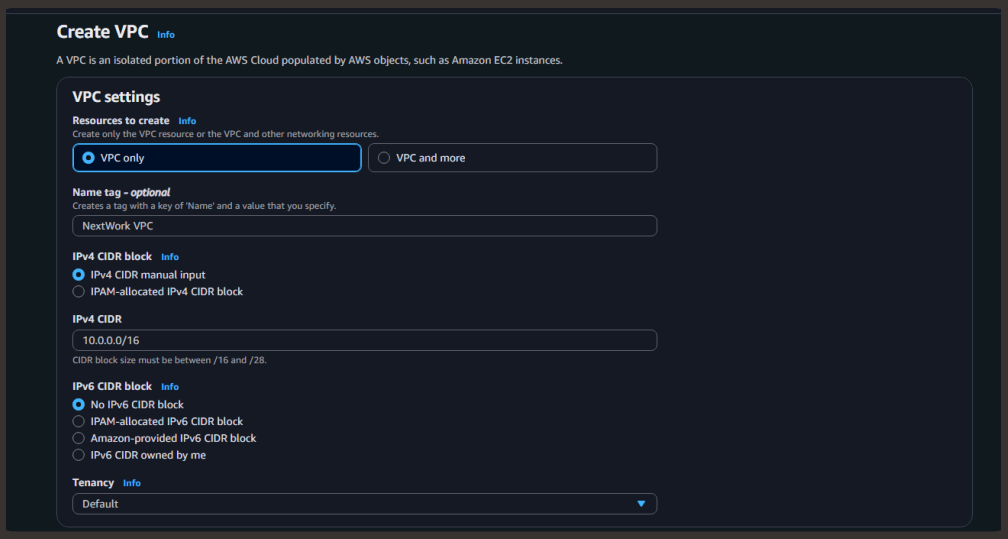
Virtual Private Clouds (VPCs)

VPCs are software-defined networks in AWS that provide logical isolation, allowing you to define your own IP address range, create subnets, and configure route tables and network gateways

There was already a default VPC in my account ever since my AWS account was created. This is because AWS pre-configures a default VPC with basic networking components (subnets, internet gateway, route tables) to reduce the barrier to entry for users who want to deploy resources right away.

To set up my VPC, I had to define an IPv4 CIDR block, which is a notation that specifies a range of IP addresses available for my VPC. The number after the slash (/16) indicates how many IP addresses are available - in my case, 10.0.0.0/16 provides 65,536 IP addresses (from 10.0.0.0 to 10.0.255.255).

Louis
NextWork Student



Subnets

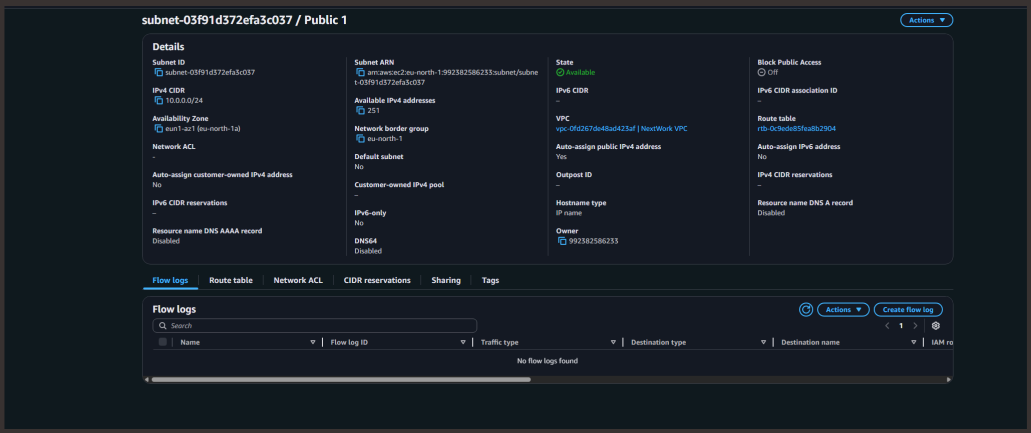
Subnets are smaller divisions or segments within a VPC that allow you to organise resources into separate network spaces, each with its own IP address range from the VPC's CIDR block. There are already subnets existing in my account, one for every Availability Zone in the region, because AWS automatically created them as part of the default VPC.

Once I created my subnet, I enabled auto-assign public IPv4 addresses. This setting makes sure that network interfaces created in this subnet automatically receive a public IPv4 address from AWS's pool, so that instances can be reached from the internet and can initiate outbound connections.

The difference between public and private subnets is their routing configuration and internet accessibility. Public subnets have a route to an internet gateway, allowing resources to communicate with the internet, while private subnets do not have direct internet access. For a subnet to be considered public, it has to have a route table with a route that directs internet-bound traffic (0.0.0.0/0) to an internet gateway.

Louis
NextWork Student

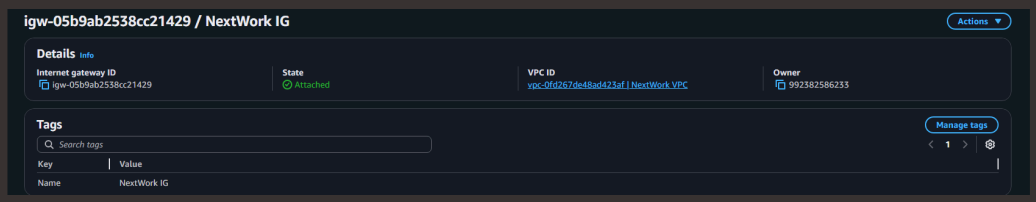
nextwork.org



Internet gateways

Internet gateways are fully managed AWS components that attach to your VPC and serve as the target for internet-routable traffic, translating between your resources' private IP addresses and their public IP addresses to enable internet communication.

Attaching an internet gateway to a VPC means that my VPC now has a pathway to communicate with the public internet, enabling resources with public IP addresses to send and receive traffic outside of AWS. If I missed this step, my resources would remain isolated and unable to access the internet or be accessed from the internet, even if they have public IP addresses and are in a public subnet.



Using the AWS CLI

VPC resources can also be created with CloudShell, which is a browser-based, pre-authenticated shell environment provided by AWS that comes with the AWS CLI and other tools pre-installed, eliminating the need to install or configure anything locally. CLI is a Command Line Interface that allows users to interact with AWS services through text-based commands instead of clicking through the graphical console, enabling automation, scripting, and faster infrastructure management.

To set up a VPC or a subnet, you can use the command "aws ec2 create-subnet", but you must include the required parameters. To avoid errors, it needs to include the "--cidr-block" parameter, which specifies the IP address range for your subnet.

Compared to using the AWS Console, an advantage of using commands is that they can be easily documented, repeated, and automated through scripts, making it faster to recreate identical infrastructure and enabling version control of your infrastructure setup. An advantage of using the Console is that it provides a visual, user-friendly interface with dropdown menus and helpful tooltips that guide you through the process, making it easier for beginners to understand the relationships between resources and catch configuration errors before they happen.

```
~ $ aws ec2 create-internet-gateway --query InternetGateway.InternetGatewayId --output text
igw-0b0ab58231ac464f7
~ $ aws ec2 attach-internet-gateway --vpc-id vpc-0a87105803b5887f9 --internet-gateway-id igw-0b0ab58231ac464f7
~ $
```

nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects