



Build a Virtual Private Cloud (VPC)

A

Andrei Boboc

The screenshot shows the 'Create VPC' wizard on the AWS VPC console. The current step is 'VPC settings'. The 'Resources to create' section is set to 'VPC only'. A 'Name tag - optional' field contains 'NextWork VPC'. The 'IPv4 CIDR block' field is set to '10.0.0.0/16'. Under 'IPv6 CIDR block', 'None' is selected. In the 'Tenancy' dropdown, 'Default' is chosen. The 'VPC encryption control' section has 'None' selected. At the bottom, there are 'Tags' fields for assigning labels to the VPC.



Andrei Boboc

NextWork Student

nextwork.org

Introducing Today's Project!

In this project, I will demonstrate how to design and deploy a basic cloud networking architecture by creating a Virtual Private Cloud (VPC), subnets, and an internet gateway. I am doing this project to practice the fundamentals of cloud networking, resource isolation, and secure connectivity in a cloud environment.

What is Amazon VPC?

Amazon VPC (Virtual Private Cloud) is a logically isolated virtual network within AWS that allows you to launch and manage resources in a defined IP address space. It is useful because it gives you full control over networking components such as subnets, routing, and security, enabling you to build secure, scalable, and customizable cloud architectures.

In today's project, I used Amazon VPC to create a custom virtual network where I defined an IP address range, created subnets, and configured internet access using an internet gateway. This allowed me to understand how AWS networking components work together to securely deploy resources.

Personal reflection

This project took me about an hour to complete.

placeholder

One thing I didn't expect in this project was how important it is to choose the correct VPC CIDR size from the beginning. Using a CIDR block that was too small made subnet creation difficult and caused conflicts, which showed me how early design decisions can impact the entire network setup.



Andrei Boboc

NextWork Student

nextwork.org

Virtual Private Clouds (VPCs)

What I did in this step

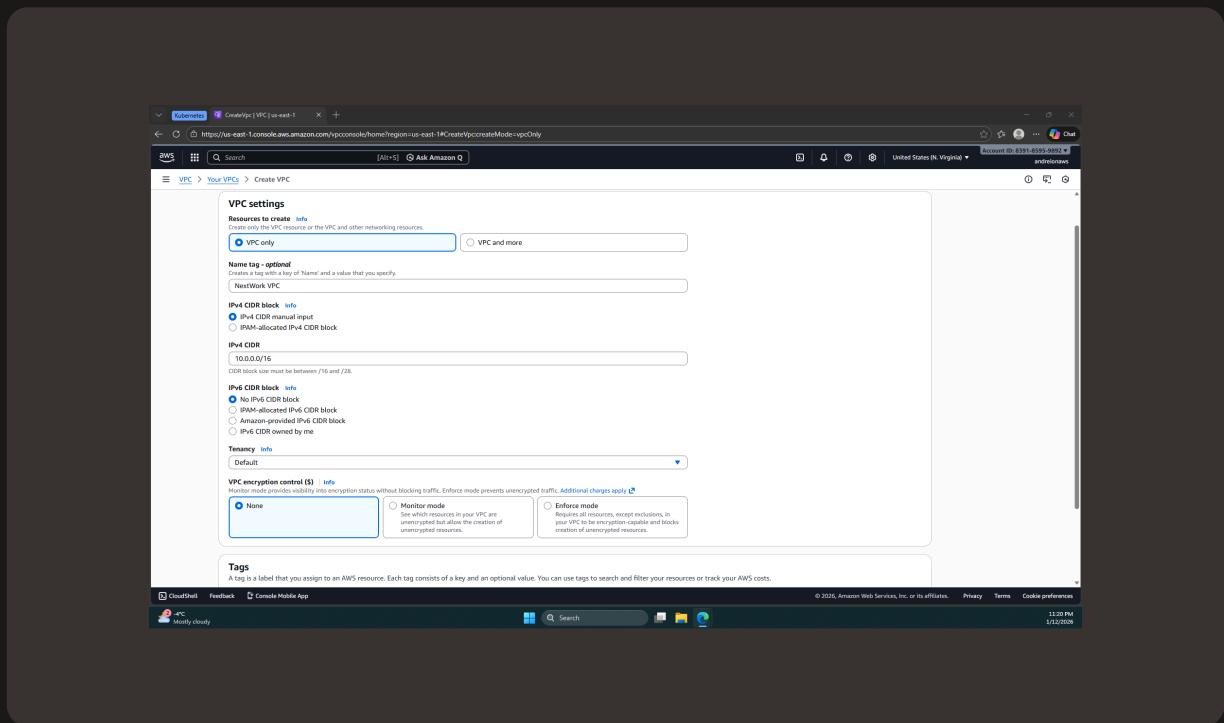
In this step, I will create a Virtual Private Cloud (VPC) with a defined CIDR range because the VPC provides an isolated network environment where all other cloud resources will be deployed and securely communicate.

How VPCs work

VPCs (Virtual Private Clouds) are logically isolated virtual networks within AWS that allow you to define your own IP address ranges, subnets, routing, and security controls for deploying cloud resources.

Why there is a default VPC in AWS accounts

There was already a default VPC in my account when it was created because AWS automatically provisions a default VPC in each region to allow users to quickly launch resources without needing to manually configure networking components.



Defining IPv4 CIDR blocks

To set up my VPC, I had to define an IPv4 CIDR block, which is a range of IP addresses written in CIDR notation that determines how many private IP addresses are available within the VPC.

A**Andrei Boboc**

NextWork Student

nextwork.org

Subnets

What I did in this step

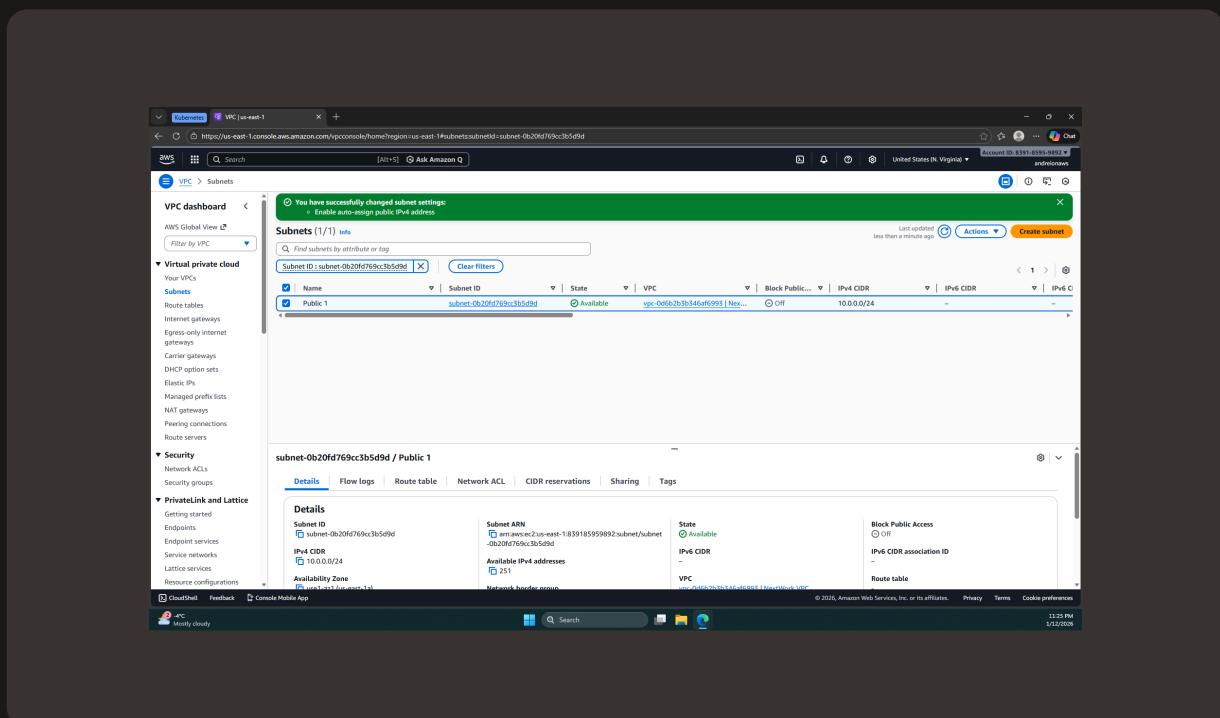
In this step, I will create subnets within the VPC because subnets divide the VPC's IP address range into smaller networks, allowing resources to be organized, isolated, and placed in specific availability zones.

Creating and configuring subnets

Subnets are smaller networks created within a VPC by dividing its CIDR block into smaller IP ranges. There are already subnets existing in my account, one for each availability zone, because AWS automatically creates default subnets in the default VPC to simplify resource deployment.

Public vs private subnets

The difference between public and private subnets is how they route traffic to the internet. For a subnet to be considered public, it must have a route to an internet gateway (IGW) in its route table and allow resources to receive public IP addresses. This subnet is not public yet because it is not associated with a route table that routes traffic to an internet gateway.



Auto-assigning public IPv4 addresses

Once I created my subnet, I enabled auto-assign public IPv4 addresses. This setting makes sure that any EC2 instance launched in this subnet automatically receives a public IP address, so that it can communicate directly with the internet without requiring manual public IP assignment.

A**Andrei Boboc**

NextWork Student

nextwork.org

Internet gateways

What I did in this step

In this step, I will create and attach an internet gateway to the VPC because an internet gateway allows resources within public subnets to communicate with the internet.

Setting up internet gateways

Internet gateways are AWS-managed components that allow communication between a VPC and the internet by enabling inbound and outbound traffic for resources with public IP addresses through routing rules.

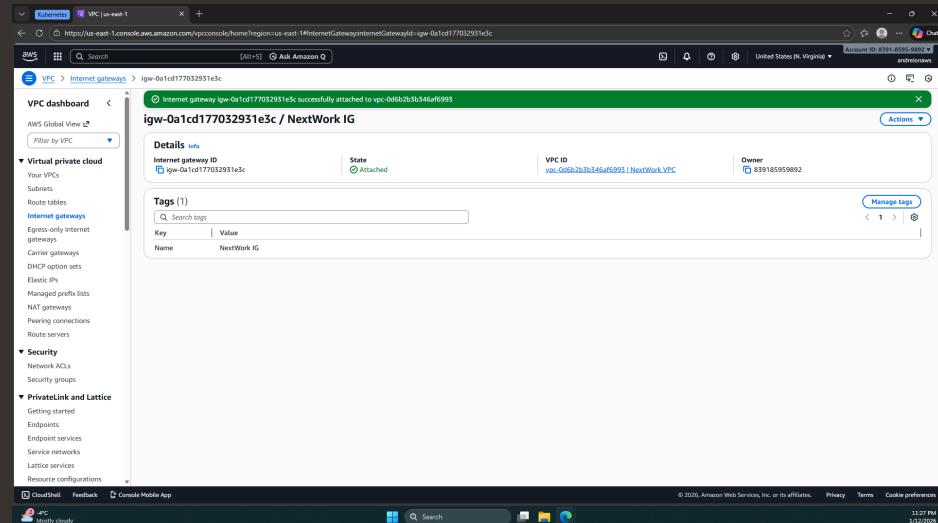
Attaching an internet gateway to a VPC means the VPC is now capable of sending and receiving traffic from the internet. If I missed this step, any subnets in the VPC would be unable to provide internet access to resources, even if they had public IP addresses and correct route table entries.

A

Andrei Boboc

NextWork Student

nextwork.org



Using the AWS CLI

Debugging my setup

I ran into an error because the command was missing required parameters or referenced resources that did not yet exist, such as a VPC ID, subnet CIDR block, or region. To avoid errors, it's important to include all required flags, verify resource IDs, and ensure the command is run in the correct region.

```
18 aws ec2 create-vpc --cidr-block 10.0.0.0/24 --query Vpc.VpcId --output text
19 aws ec2 create-tags --resources=vpc-0a7ff591cb106cf4e --tags Key=Name,Value="NextWork VPC 2"
20 aws ec2 create-subnet --vpc-id vpc-0a7ff591cb106cf4e --cidr-block 10.0.0.24/25
21 clear
22 aws ec2 create-internet-gateway
23 aws ec2 attach-internet-gateway --internet-gateway-id igw-0debb194b67ef3480 --vpc-id vpc-0a7ff591cb106cf4e
```

Comparing CloudShell vs AWS Console

Compared to using the AWS Console, an advantage of setting up VPC resources in CloudShell using CLI commands is that it allows for faster, repeatable, and more precise configuration, which is especially useful for automation and scripting. An advantage of using the Console is its visual interface, which makes it easier to understand resource relationships and reduces the risk of mistakes for beginners.



A

Andrei Boboc

NextWork Student

nextwork.org

Overall, I preferred using the AWS Console for learning, while CloudShell is better suited for automation and advanced workflows.



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

