



# Build a Virtual Private Cloud

G Gloria

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

## VPC settings

### Resources to create [Info](#)

Create only the VPC resource or the VPC and other networking resources.

VPC only

VPC and more

### Name tag - *optional*

Creates a tag with a key of 'Name' and a value that you specify.

NextWork VPC

### IPv4 CIDR block [Info](#)

- IPv4 CIDR manual input
- IPAM-allocated IPv4 CIDR block

### IPv4 CIDR

10.0.0.0/16

CIDR block size must be between /16 and /28.

### IPv6 CIDR block [Info](#)

- No IPv6 CIDR block
- IPAM-allocated IPv6 CIDR block
- Amazon-provided IPv6 CIDR block
- IPv6 CIDR owned by me

# Introducing Today's Project!

## What is Amazon VPC?

Amazon VPC (Virtual Private Cloud) is a secure, isolated network in AWS where you can launch resources like EC2 instances. It's useful because it gives control over networking, such as IP ranges, subnets and security.

## How I used Amazon VPC in this project

Today, I used Amazon VPC to create an isolated network for my resources. I set up subnets and attached an internet gateway for public access, enabling my instances to communicate with the internet.

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

One thing I didn't expect in this project was how setting up subnets and attaching an internet gateway impacts public accessibility ,it highlighted the importance of properly configuring resources for connectivity.

## This project took me...

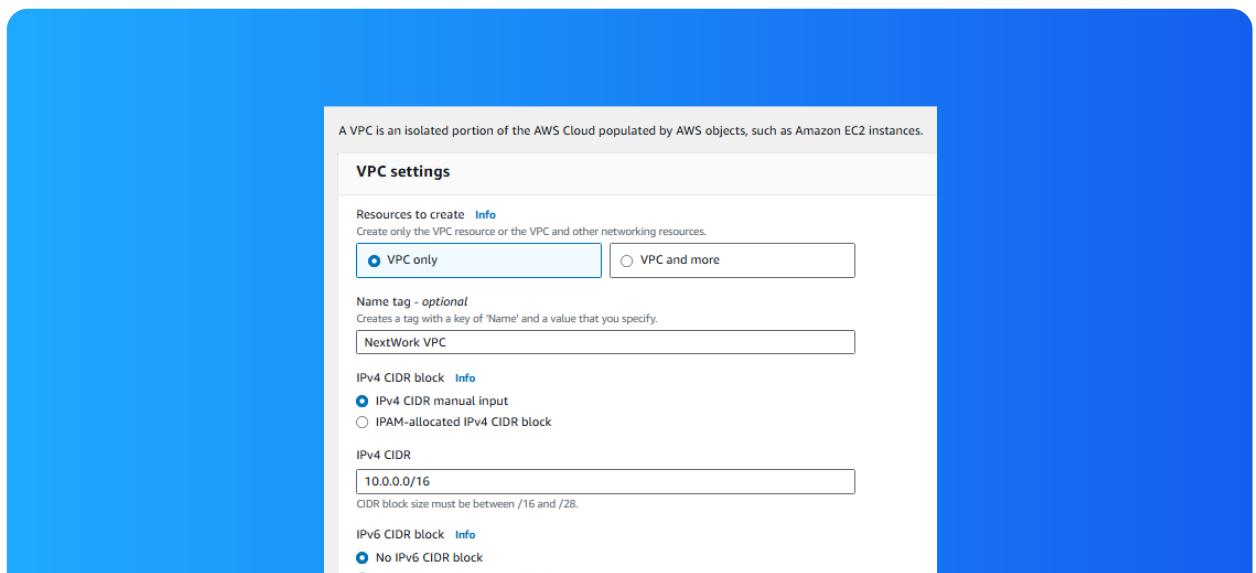
This project took less than 10 minutes to complete. The AWS interface made everything easy and faster to provision.

# Virtual Private Clouds (VPCs)

VPCs are Virtual Private Clouds that provide isolated cloud networks for securely managing resources like servers and databases, with control over IP ranges, subnets, and access. It controls access to EC2 instances, databases and more.

There was already a default VPC in my account ever since my AWS account was created. This is because AWS provides it to make using AWS from the start easier and beginner-friendly, enabling quick resource deployment without needing custom VPC setup.

To set up my VPC, I had to define an IPv4 CIDR block, which is a range of IP addresses, like 10.0.0.0/16. It specifies the network's starting IP and the size of the range using subnet masks. Less number in the CIDR block means higher IP addresses.

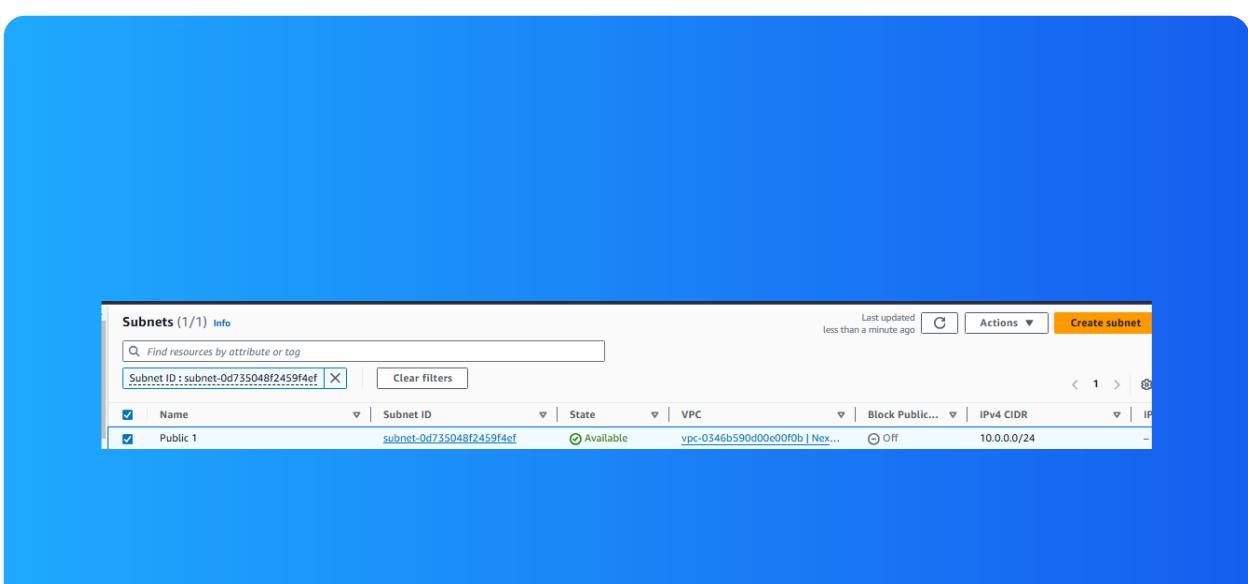


# Subnets

Subnets are smaller sections of a network in a VPC. They help organize and control resources. There are already subnets existing in my account, one for every availability zone, making it easier to manage and distribute resources across the zones.

Once I created my subnet, I enabled auto-assign public IPv4 addresses. This setting makes sure instances launched in the subnet automatically get public IPs, so they can communicate with the internet without needing manual IP assignment.

The difference between public and private subnets is their internet accessibility. For a subnet to be considered public, it has to be connected to an internet gateway. Without this connection, even if labeled "Public," it remains private.

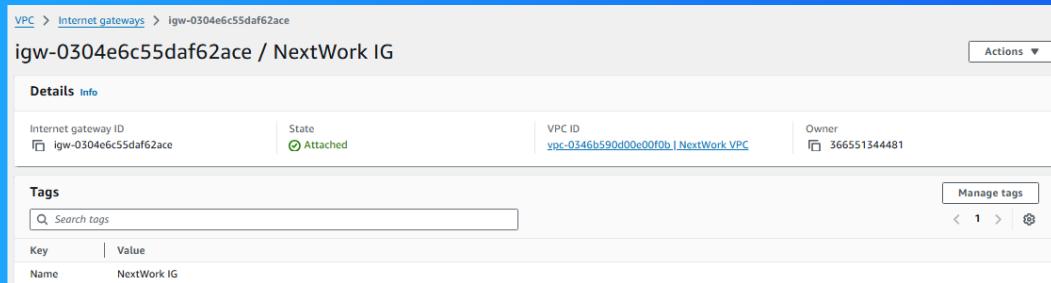


Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IP
Public 1	subnet-0d735048f2459f4ef	Available	vpc-0346b590d00e00f0b   Nex...	Off	10.0.0.0/24	-

# Internet gateways

Internet gateways are resources that allow communication between instances in a VPC and the internet. They provide a path for data to flow in and out of public subnets, enabling instances to send and receive traffic over the internet.

Attaching an internet gateway to a VPC means enabling resources in public subnets to communicate with the internet. If I missed this step, instances in public subnets wouldn't be able to send or receive traffic over the internet.





NextWork.org

# Everyone should be in a job they love.

Check out nextwork.org for  
more projects

