



nextwork.org

Build a Virtual Private Cloud (VPC)

AN

antcruz567@gmail.com

Create VPC Info

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

Tenancy Info
Default

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is a service provided that allows you to create a logically isolated network within the AWS cloud. Essentially, it's your own private data center in the cloud, where you have full control over your network configuration.

How I used Amazon VPC in this project

I created A VPC in the mangement console and the powershell then created subnets and attached an internet gateway to the VPC.

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

I did not expect the PowerShell CLI to be so fluid and fast when creating resources such as the internet gateway, and subnets.

This project took me...

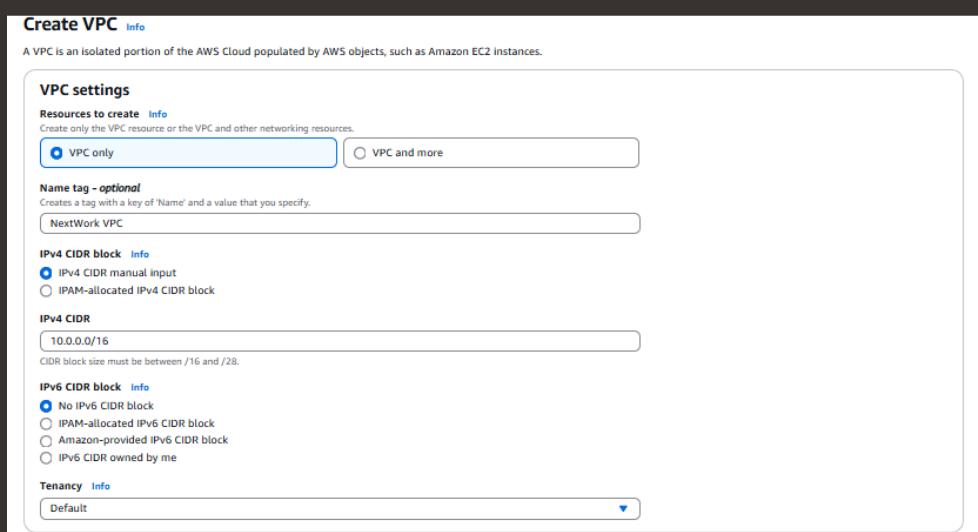
This project took me 2 hours due to troubleshooting wrong code within the powershell.

Virtual Private Clouds (VPCs)

If we imagine an AWS Region as a country, a Virtual Private Cloud (VPC) is like your own private city inside that country. We can design neighborhoods (subnets), traffic rules, and security measures to control how resources, like EC2 instances and databases, connect and work together.

There was already a default VPC in my account ever since my AWS account was created. This is because AWS automatically sets up a default VPC for me. This default VPC is why I could launch resources (e.g. EC2 instances) and connect services from Day 1 of using AWS.

To set up my VPC, I had to define an IPv4 CIDR block, (Classless Inter-Domain Routing). This is a way to assign a whole block of IP addresses to my VPC.



Subnets

Subnets are like different neighborhoods inside your city. You use subnets to group resources with similar access rules and restrictions. Some subnets might be public areas that all resources can access (public subnets), while others are private areas with limited access (private subnets). There are already subnets existing in my account, one for every availability zone.

Once I created my subnet, I enabled the option to auto assign public IPv4 addresses. This setting makes sure any EC2 instance launched in my subnet will instantly get a public IP address so I will not have to create one manually.

The difference between public and private subnets is that a public subnet is connected to the internet. Resources inside a public subnet can communicate with external networks. A private subnet does not have direct internet access. It is used for internal resources that don't need to be publicly accessible. For a subnet to be considered public, it has to have an internet gateway.

AN

antcruz567@gmail.com
NextWork Student

NextWork.org

You have successfully changed subnet settings:
Enable auto-assign public IPv4 address

Last updated less than a minute ago

Actions Create subnet

Subnets (1/1) Info

Find resources by attribute or tag Subnet ID: subnet-0702fc840753876fb X Clear filters

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR	IPv6 CIDR association ID
Public 1	subnet-0702fc840753876fb	Available	vpc-09069d5e7e70d3697 Next... Off 10.0.0.0/24 - -				

subnet-0702fc840753876fb / Public 1

Details Flow logs Route table Network ACL CIDR reservations Sharing Tags

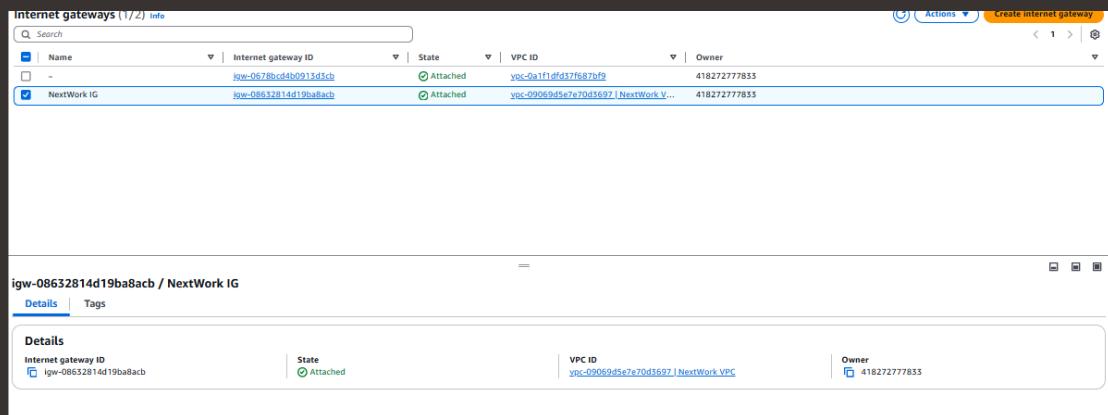
Details

Subnet ID	subnet-0702fc840753876fb	Subnet ARN	arn:aws:ec2:us-east-1:4182777833:subnet/subnet-0702fc840753876fb	State	Available
IPv4 CIDR	10.0.0.0/24	Available IPv4 addresses	251	IPv6 CIDR	-
Availability Zone	us-east-1a	Availability Zone ID	use1-az6	Network border group	us-east-1
Route table	-	Network ACL	-	Default subnet	No
Auto-assign IPv6 address	No	Auto-assign customer-owned IPv4 address	No	Customer-owned IPv4 pool	-
IPv4 CIDR reservations	None	IPv6 CIDR reservations	None	IPv6-only	No
Block Public Access	Off	IPv6 CIDR association ID	-	VPC	vpc-09069d5e7e70d3697 NextWork VPC
Auto-assign public IPv4 address	Yes	Outpost ID	-	Hostname type	IP name

Internet gateways

Internet gateways connect VPCs to the Internet.

Attaching an internet gateway to a VPC means my EC2 instances can access the internet. If I missed this step, resources in my VPC would not be able to access the internet.

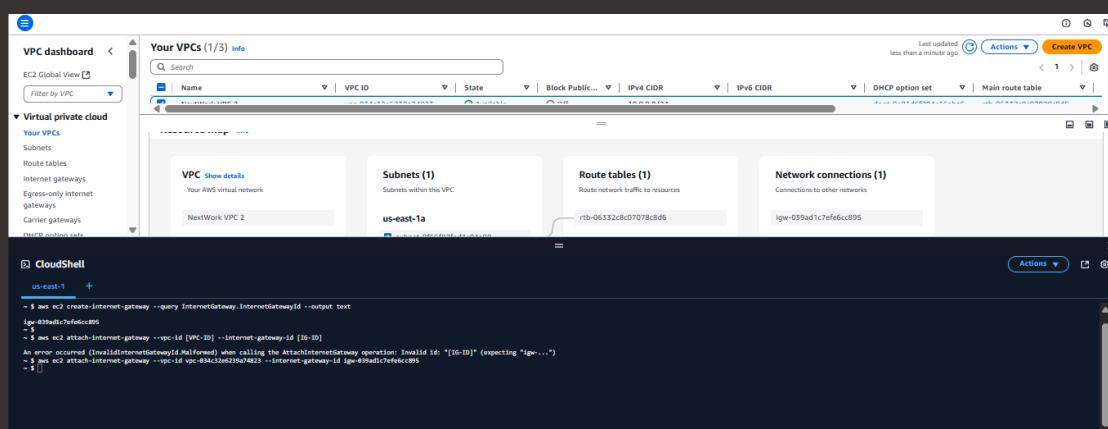


Using the AWS CLI

VPC resources could also be created with CloudShell, which is a unique way to automate tasks using scripts, making the CLI essential for managing your cloud environment in an efficient way.

To set up a VPC or a subnet, you can use the command `aws ec2 create-vpc` or `create-subnet`. Make sure to avoid errors by including the CIDR block and correctly inputting the VPC ID.

Compared to using the AWS Console, an advantage of using commands is that they are more productive and faster when making and organizing resources. Overall, I preferred cloudshell BUT will need to practice more to become more efficient. This project was a good first step.



The screenshot shows the AWS CloudShell interface integrated into the AWS Management Console. The top navigation bar includes 'Your VPCs (1/3) info' and 'Actions' with a 'Create VPC' button. Below this, there are sections for 'Subnets', 'Route tables', 'Network connections', and 'Main route table'. On the left, a sidebar lists 'Virtual private cloud' options like Subnets, Route tables, Internet gateways, Egress-only Internet gateways, Carrier gateways, and Network interface sets. A 'CloudShell' tab is open, showing a terminal window with the following AWS CLI commands:

```
$ aws ec2 create-internet-gateway --query InternetGateway.Id --output text
igw-039ad1c7fe6cc895
$ aws ec2 attach-internet-gateway --vpc-id [VPC-ID] --internet-gateway-id [IG-ID]
An error occurred (InvalidInternetGatewayId.NotFound) when calling the AttachInternetGateway operation: Invalid id: "[IG-ID]" (expecting "igw-...")

$ aws ec2 attach-internet-gateway --vpc-id vpc-084c32e238a4823 --internet-gateway-id igw-039ad1c7fe6cc895
$
```

**Everyone
should be in a
job they love.**

Check out nextwork.org for
more projects

