

Build a Virtual Private Cloud



chinmayegowda11@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.

chinmaye-VPC

IPv4 CIDR block Info

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

IPv4 CIDR

10.0.0.0/24

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 that lets you create an isolated network in AWS, where you can control IP ranges, subnets, route tables, and security settings. It's useful for securing workloads, enabling internet access, and managing private cloud.

How I used Amazon VPC in this project

I used Amazon VPC in today's project to create a secure, isolated network for my AWS resources. I set up subnets, attached an internet gateway for public access, and ensured proper networking for my instances to communicate effectively.

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

One thing I didn't expect in a VPC project is the level of detail needed in subnetting and route table configuration to ensure optimal security and connectivity.

This project took me...

The VPC project took around 50 minutes to complete, focusing on subnetting, route table configuration, and security group setup. While it can vary depending on complexity, this time frame allowed for a streamlined, basic setup.

Virtual Private Clouds (VPCs)

VPCs are Virtual Private Clouds, isolated network environments within a public cloud that allow users to define & control their own networking setup, including subnets, route tables, & security groups, ensuring secure & scalable cloud infrastructure.

There was already a default VPC in my account ever since my AWS account was created. This is because AWS provides a default VPC to enable easy deployment of resources with basic networking, including subnets, route tables, & internet access.

To set up my VPC, I had to define an IPv4 CIDR block, which is a range of IP addresses allocated for my network. It helps segment & manage IP assignments within the VPC, ensuring organized communication b/w resources while avoiding address conflicts.

The screenshot shows the 'Create VPC' configuration page in the AWS Management Console. The page has a header 'Create VPC Info' and a sub-header: 'A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.' Below this is a section titled 'VPC settings' with a sub-section 'Resources to create Info'. It contains two radio buttons: 'VPC only' (selected) and 'VPC and more'. A 'Name tag - optional' field contains the value 'chinmaye-VPC'. Under 'IPv4 CIDR block Info', there is a radio button for 'IPv4 CIDR manual input' (selected) and another for 'IPAM-allocated IPv4 CIDR block'. The input field shows '10.0.0.0/24'. A note states: 'CIDR block size must be between /16 and /28.' Below this is an 'IPv6 CIDR block Info' section with a radio button for 'No IPv6 CIDR block' (selected) and three other options: 'IPAM-allocated IPv6 CIDR block', 'Amazon-provided IPv6 CIDR block', and 'IPv6 CIDR owned by me'. At the bottom is a 'Tenancy Info' section with four radio button options: 'Shared', 'AWS Lambda', 'Host', and 'Dedicated'. The 'Dedicated' option is selected.

Subnets

Subnets are smaller, segmented networks within VPC that help organize & control traffic flow. There are already subnets existing in my acc, one for every Availability Zone in the default VPC, allowing resources to be distributed for high availability

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

The difference b/w public & private subnets are their accessibility from the internet. For a subnet to be considered public, it has to be associated with a route table that directs traffic to an Internet Gateway, allowing external access to resources

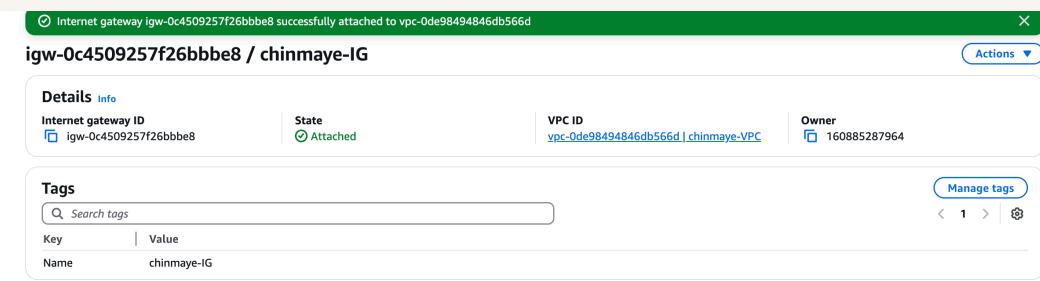
Edit subnet settings [Info](#)

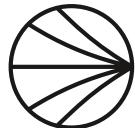
Subnet	Name
Subnet ID subnet-07f50f590e19df0b2	Name Public 1
Auto-assign IP settings Info Enable AWS to automatically assign a public IPv4 or IPv6 address to a new primary network interface for an instance in this subnet.	
<input checked="" type="checkbox"/> Enable auto-assign public IPv4 address Info	
<input type="checkbox"/> Enable auto-assign customer-owned IPv4 address Info Option disabled because no customer owned pools found.	
Resource-based name (RBN) settings Info Specify the hostname type for EC2 instances in this subnet and optional RBN DNS query settings.	
<input type="checkbox"/> Enable resource name DNS A record on launch Info	
<input type="checkbox"/> Enable resource name DNS AAAA record on launch Info	
Hostname type Info	
<input type="radio"/> Resource name	

Internet gateways

Internet gateways are AWS components that allow traffic between your VPC & the internet. They enable instances in a public subnet to communicate externally. Each VPC can have only one internet gateway, which must be attached to route outbound traffic

Attaching an internet gateway to a VPC means enabling internet access for resources in public subnets. If I missed this step, instances wouldn't be able to connect to or receive traffic from the internet, making it impossible to host public-facing.





NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

