

Placement Empowerment Program

Cloud Computing and DevOps Centre

Set Up a Private Network in the Cloud : Create a Virtual Private Cloud (VPC) with subnets for your instances. Configure routing for internal communication between subnets.

Name: Vishali V

Department:
CSE

Introduction

The goal of this Proof of Concept (PoC) was to set up a Private Network in the Cloud by creating a Virtual Private Cloud (VPC) in AWS, configuring subnets, and ensuring internal communication between instances within the VPC. This setup focused on isolating cloud resources in a private network, providing a secure environment for communication, and making sure that only internal traffic is allowed, without exposing resources to the public internet.

In this PoC, we created a private subnet where EC2 instances

could

communicate with each other without direct exposure to external networks.

Overview

In this PoC, we:

1. Created a VPC in AWS, which serves as the isolated private network.
2. Created a private subnet inside the VPC where EC2 instances can reside, ensuring no direct access from the public internet.
3. Set up routing to allow communication between the instances within the same VPC and subnet.
4. Launched EC2 instances in the private subnet and verified their ability to communicate internally using their private IP addresses.

The setup is designed to simulate a secure cloud environment where resources can interact securely without being exposed to external traffic.

Objective

The primary objectives of this PoC were:

1. **Establish a Private Network:** Set up a private VPC and subnets for cloud resources to reside in, ensuring they are isolated from the public internet.
2. **Internal Communication:** Ensure that EC2 instances within the private subnet can communicate with each other using their private IPs.
3. **Security:** Maintain internal communication only within the VPC, preventing direct exposure of instances to the public internet.
4. **Simplify Management:** Organize cloud resources into subnets for easier management and scaling, with clear routing between them.

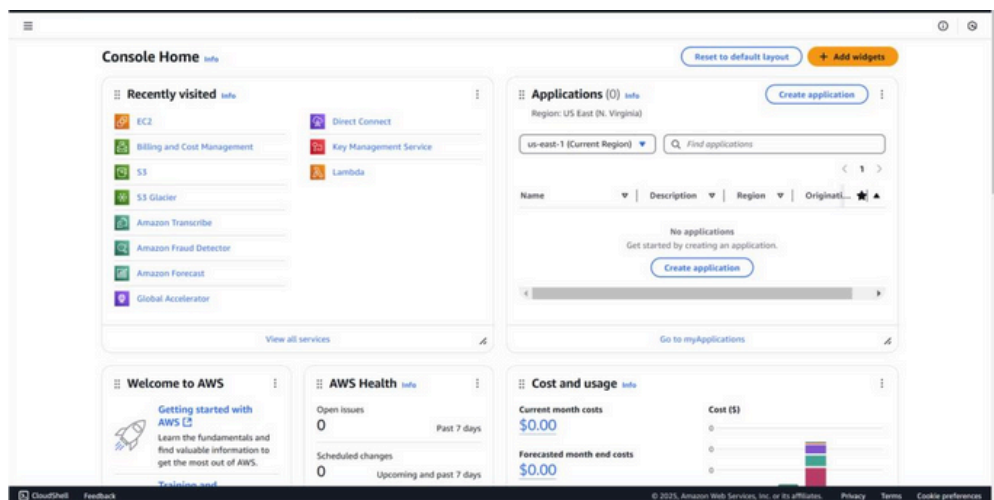
Importance

1. **Security:** By placing EC2 instances in a private subnet and ensuring that no public IP is assigned, the resources are isolated from external traffic. This is crucial for keeping sensitive data and services protected.
2. **Cost Efficiency:** Using internal communication and private subnets can help reduce costs related to public internet access and data transfer.
3. **Flexibility:** This setup provides a foundation for building more complex cloud infrastructures, such as multi-tier applications where only backend servers (databases, app servers) are private, while frontend servers may be public.

Step-by-Step Overview

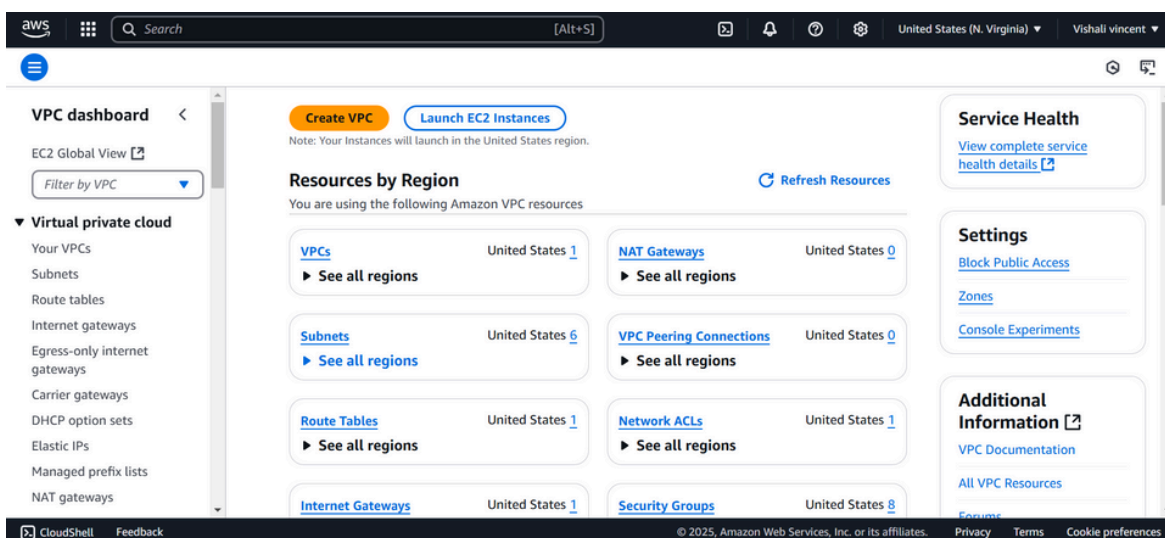
Step 1:

1. Go to [AWS Management Console](#).
2. Enter your username and password to log in.



Step 2:

In the VPC Dashboard, click the Create VPC button.



Step 3:

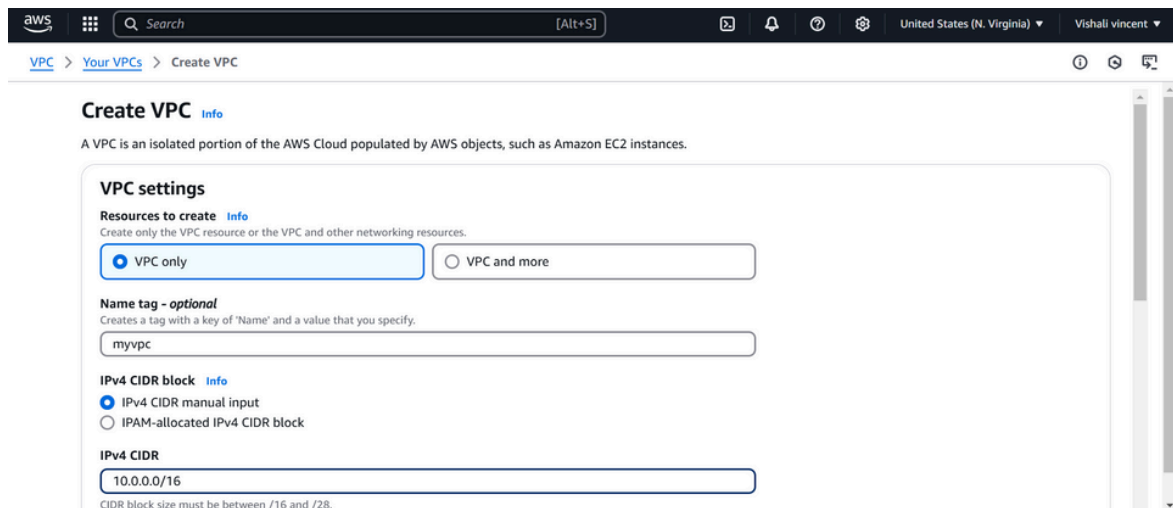
In the VPC creation wizard, select VPC only.

Name tag: Enter MyVPC .

IPv4 CIDR block: Enter 10.0.0.0/16 (this defines the IP range for your VPC).

Tenancy: Leave it as Default.

Click Create VPC.

The screenshot shows the AWS Management Console's 'Create VPC' page. The breadcrumb navigation at the top reads 'VPC > Your VPCs > Create VPC'. The main heading is 'Create VPC' with an 'Info' link. Below this is a descriptive sentence: 'A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.' The 'VPC settings' section contains three main parts: 1. 'Resources to create' with two radio buttons: 'VPC only' (selected) and 'VPC and more'. 2. 'Name tag - optional' with a text input field containing 'myvpc'. 3. 'IPv4 CIDR block' with two radio buttons: 'IPv4 CIDR manual input' (selected) and 'IPAM-allocated IPv4 CIDR block'. Below this is a text input field for the CIDR block containing '10.0.0.0/16'. A small note at the bottom states 'CIDR block size must be between /16 and /28'.

Step 4:

In the VPC Dashboard, click on Subnets in the left-hand menu.

Click the Create subnet button.

VPC: Select MyVPC (the one you just created).

Subnet name: Enter Private-Subnet.

Availability Zone: Pick any (e.g., us-east-1a or any zone from your region).

IPv4 CIDR block: Enter 10.0.1.0/24 (this is a smaller range within the VPC's IP range).

Click Create subnet.

The screenshot shows the AWS VPC dashboard. A green notification bar at the top states: "You have successfully created 1 subnet: subnet-076c596b1cbf01323". Below this, the "Subnets (1) Info" section displays a table with one subnet:

Name	Subnet ID	State	VPC
myprivatevpc	subnet-076c596b1cbf01323	Available	vpc-029a3880b63d3d88b my...

The left sidebar shows the "Virtual private cloud" section with options like "Your VPCs", "Subnets", "Route tables", etc.

The screenshot shows the "Create VPC" wizard. Under "VPC settings", the "Resources to create" section has "VPC only" selected. The "Name tag - optional" field contains "myvpc". Under "IPv4 CIDR block", "IPv4 CIDR manual input" is selected, and the "IPv4 CIDR" field contains "10.0.0/16". A note below states: "CIDR block size must be between /16 and /28."

The screenshot shows the details page for subnet "subnet-076c596b1cbf01323 / myprivatevpc". The "Details" section includes the following information:

- Subnet ID:** subnet-076c596b1cbf01323
- IPv4 CIDR:** 10.0.1.0/24
- Availability Zone:** us-east-1a
- Route table:** -
- Auto-assign IPv6 address:** No
- IPv4 CIDR reservations:** -

Other details include Subnet ARN, Available IPv4 addresses (251), Availability Zone ID (use1-az2), Network ACL, and various configuration options like "Block Public Access" (Off) and "Auto-assign public IPv4 address" (No).

Step 5:

In the VPC Dashboard, click on Route Tables in the left-hand menu. Click Create route table.

Name tag: Enter InternalRouteTable.

VPC: Select MyVPC (the one you created earlier).

Click Create route table.

The screenshot shows the 'Create route table' page in the AWS Management Console. The breadcrumb navigation is 'VPC > Route tables > Create route table'. The page title is 'Create route table' with an 'Info' link. A description states: 'A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.'

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.
internalroutetable

VPC
The VPC to use for this route table.
vpc-029a3880b63d3d88b (myvpc)

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key
Name

Value - optional
internalroutetable

Buttons: 'Add new tag', 'Remove'.

The screenshot shows the 'Route table details' page in the AWS Management Console. The breadcrumb navigation is 'VPC > Route tables > rtb-0a24b13f96b74a7d0'. A green success message at the top states: 'Route table rtb-0a24b13f96b74a7d0 | internalroutetable was created successfully.'

Details

Route table ID
rtb-0a24b13f96b74a7d0

VPC
vpc-029a3880b63d3d88b | myvpc

Main
No

Owner ID
970547350598

Explicit subnet associations
-

Edge associations
-

Routes (1)

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

Step 6:

Select the InternalRouteTable you just created.

Go to the Subnet Associations tab (it's near the bottom).

Click Edit subnet associations.

Select Private-Subnet (the subnet you created earlier).

Click Save associations.

The screenshot shows the AWS Management Console interface for editing subnet associations. The breadcrumb trail at the top indicates the path: VPC > Route tables > rtb-0a24b13f96b74a7d0 > Edit subnet associations. The page title is 'Edit subnet associations' with a subtitle 'Change which subnets are associated with this route table.' Below this, there are two main sections: 'Available subnets (1/1)' and 'Selected subnets'. The 'Available subnets' section contains a table with one row for 'myprivatevpc' (subnet-076c596b1cbf01323) with IPv4 CIDR 10.0.1.0/24. The 'Selected subnets' section shows the same subnet and name. At the bottom right, there are 'Cancel' and 'Save associations' buttons. The footer includes 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates.

aws [Search] [Alt+S] United States (N. Virginia) Vishali vincent

VPC > Route tables > rtb-0a24b13f96b74a7d0 > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/1)

Filter subnet associations

<input checked="" type="checkbox"/>	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/>	myprivatevpc	subnet-076c596b1cbf01323	10.0.1.0/24	-	Main (rtb-0778480acb700c046)

Selected subnets

subnet-076c596b1cbf01323 / myprivatevpc

Cancel Save associations

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 7:

To launch a new EC2 instance in your private subnet, go to the EC2 Dashboard, click Launch Instance, and fill in the details: Name it "Private-Instance", choose an Amazon Linux 2 AMI (or another free-tier eligible image), select the t2.micro instance type, and either choose an existing key pair or create a new one for SSH access. Under Network settings, select your MyVPC and Private-Subnet, and make sure Auto-assign Public IP is disabled to keep it private. Leave all other settings as default, then click Launch Instance.

aws Search [Alt+S] United States (N. Virginia) Vishali vincent

EC2 > Instances > Launch an instance

Network settings Info

VPC - required Info
vpc-09f8b583f34e30082 (default) 172.31.0.0/16

Subnet Info
No preference Create new subnet

Auto-assign public IP Info
Disable

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
☒ Create security group ☐ Select existing security group

Security group name - required
launch-wizard-8

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters are: a-z, 0-9, hyphen, underscore, equals, at, and colon.

Summary

Number of instances Info
1

opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel Launch instance Preview code

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Search [Alt+S] United States (N. Virginia) Vishali vincent

EC2 > Instances > Launch an instance

Network settings Info

VPC - required Info
vpc-029a3880b63d3d88b (myvpc) 10.0.0.0/16

Subnet Info
subnet-076c596b1cbf01323 myprivatevpc VPC: vpc-029a3880b63d3d88b Owner: 970547350598 Availability Zone: us-east-1a Zone type: Availability Zone IP addresses available: 251 CIDR: 10.0.1.0/24 Create new subnet

Auto-assign public IP Info
Disable

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
☒ Create security group ☐ Select existing security group

Security group name - required
launch-wizard-8

Summary

Number of instances Info
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...read more
ami-05b10e08d247fb927

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)

Cancel Launch instance Preview code

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 8: Verify Internal Communication

1. Find the private IP of your instance:

Go to the EC2 Dashboard.

Select your instance in Private-Subnet.

Note the Private IPv4 address (e.g., 10.0.1.x).

2. Ping the Private IP:

If you have only one instance, you can skip this. If you have multiple instances in the private subnet, SSH into one instance and try pinging the private IP of the other instance.

Outcome

By completing this PoC of setting up a Private Network in AWS, you will:

1. Deploy a VPC with a private subnet to isolate cloud resources securely from the public internet.
2. Launch EC2 instances within the private subnet and ensure internal communication between them using private IPs.
3. Configure routing tables to enable efficient communication within the VPC while maintaining the isolation of private resources.
4. Implement security groups to allow only internal traffic between instances while restricting external access.
5. Gain practical experience in designing secure cloud architectures and foundational AWS services like VPC, EC2, and private networking.

