



# Creating a Private Subnet

G Gloria

**Subnet 1 of 1**

**Subnet name**  
Create a tag with a key of 'Name' and a value that you specify.  
  
The name can be up to 256 characters long.

**Availability Zone** [Info](#)  
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

**IPv4 VPC CIDR block** [Info](#)  
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

**IPv4 subnet CIDR block**  
 256 IPs  
[Edit](#)

**Tags - optional**

Key	Value - optional
<input type="text" value="Name"/> <a href="#">X</a>	<input type="text" value="NextWork Private Subnet"/> <a href="#">X</a> <a href="#">Remove</a>

[Add new tag](#)  
You can add 49 more tags.  
[Remove](#)

[Add new subnet](#)



# 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

I used Amazon VPC in today's project to create a secure and isolated network, set up private and public subnets, configured route tables for traffic control, and implemented custom network ACLs to enhance security and manage traffic flow.

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

One thing I didn't expect in this project is how critical security is at every level. Configuring subnets, route tables, and network ACLs made me realize the importance of carefully managing traffic to ensure the environment remains secure.

## This project took me...

This project took me 20 minutes to complete. With prior setups from earlier projects, the process was smooth and beginner-friendly, highlighting the importance of security configurations at every step.

# Private vs Public Subnets

The difference between public and private subnets is that a public subnet has a route to the internet through an internet gateway, allowing external access, while a private subnet does not, keeping its resources isolated and accessible only within.

Having private subnets are useful because they enhance security by isolating sensitive resources, like databases or application servers, from direct internet access. This reduces exposure to external threats while allowing controlled communication.

My private and public subnets cannot have the same CIDR block, as each subnet in a VPC must have a unique range of IP addresses to ensure proper routing and avoid conflicts within the network.

Subnet 1 of 1

Subnet name  
Create a tag with a key of 'Name' and a value that you specify.  
  
The name can be up to 256 characters long.

Availability Zone Info  
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 VPC CIDR block Info  
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block  
 256 IPs

Tags - optional

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="NextWork Private Subnet"/>

Add new tag  
You can add 49 more tags.

# A dedicated route table

By default, my private subnet is associated with the main route table of the VPC, which the public subnet is already associated with. This setup could expose the private subnet to public internet routes, making it necessary to create a new one.

I had to set up a new route table because my private subnet needs routing rules specific to internal communication, like directing traffic to a NAT gateway, while avoiding public internet access routes to maintain security and isolation

My private subnet's dedicated route table only has one inbound and one outbound rule that allows local traffic within the VPC, ensuring secure communication between internal resources while blocking direct access to and from the internet.

The screenshot shows the AWS Route Table configuration page for 'rtb-0d7910e1d42fc5895' in the 'NextWork Private' VPC. The 'Details' tab is selected, showing the following information:

Route table ID rtb-0d7910e1d42fc5895	Main No	Explicit subnet associations subnet-081c60923999096ea / NextWork Private Subnet	Edge associations -
VPC vpc-0346b590d00e00f0b   NextWork VPC	Owner ID 366551344481		

Below the details, the 'Routes' tab is selected, displaying one route entry:

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

# A new network ACL

By default, my private subnet is associated with the default network ACL provided by AWS, which allows all inbound and outbound traffic unless specific deny rules are added.

I set up a dedicated network ACL for my private subnet because it allows me to define custom rules for controlling inbound and outbound traffic, enhancing security by restricting access and isolating the subnet from unwanted traffic.

My new network ACL has two simple rules - an inbound rule that denies all traffic ('0.0.0.0/0') across all protocols and ports to block unauthorized access, and an outbound rule that also denies all traffic to prevent any data from leaving the subnet

Inbound rules (1)						
<input type="text"/> Filter inbound rules						
Rule number	Type	Protocol	Port range	Source	Allow/Deny	
*	All traffic	All	All	0.0.0.0/0	<input checked="" type="radio"/> Deny	



NextWork.org

# Everyone should be in a job they love.

Check out nextwork.org for  
more projects

