



Creating a Private Subnet



RAJARSHI VERMA

VPC

VPC ID
Create subnets in this VPC.

Associated VPC CIDRs

IPv4 CIDRs

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

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

 You can add 49 more tags.



Introducing Today's Project!

What is Amazon VPC?

Amazon VPC (Virtual Private Cloud) allows you to create a private, isolated section of the AWS cloud where you can launch AWS resources in a virtual network. It's useful for controlling network settings, enhancing security, and managing traffic.

How I used Amazon VPC in this project

In today's project I created a private subnet in case of keeping resources private like in case of database or sensitive data. Along with that I created private route table, private NACL.

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

It is quiet easy to make a private subnet and by default everything is pointing to open access to internet so it becomes necessary to check every aspect while setting up private subnet because you don't want anyone to sneak inside your machine.

This project took me...

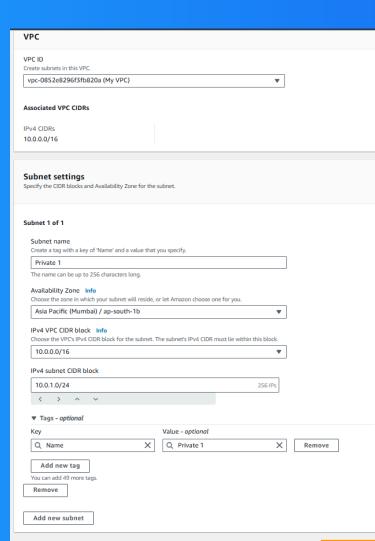
It took me around 60 minutes to set everything from scratch that including creating VPC, public and private subnets, internet gateway, public and private routetable and public and private network acces control list.

Private vs Public Subnets

The difference between public and private subnets is that public subnets have direct access to the internet via an internet gateway, while private subnets do not. Private subnets are used for resources that should not be directly accessible.

Having private subsets are useful because there are always resources which do not require the direct access to internet and it is not best practice to place them in public subnet like database or any sensitive information.

My private and public subnets cannot have the same CIDR block. Each subnet in a VPC must have a unique CIDR block to avoid IP address conflicts and ensure proper routing of traffic within the network. For public-1 10.0.0.0/24 and for private 10.0.1.0



A dedicated route table

By default, my private subnet is associated with the main route table of the VPC. This route table does not have a route to an internet gateway, ensuring that the subnet remains private and isolated from direct internet access.

I had to set up a new route table because the default route table is associated with the public subnet and has a route to the internet gateway. To keep the private subnet isolated, it needs a separate route table without internet access.

My private subnet's dedicated route table only has one inbound and one outbound rule that allows internal VPC traffic. This ensures that the subnet remains isolated from the internet while still enabling communication within the VPC.

Route tables (3) Info						
<input type="checkbox"/>	Name	Route table ID	Explicit subnet assoc...	Edge associations	Main	VPC
<input type="checkbox"/>	My Public RouteTable	rtb-0377165513ec74936	subnet-0e7af847f642f9d...	-	Yes	vpc-0852e8296f3fb820a My V...
<input type="checkbox"/>	-	rtb-0ea8ba70ad753075d	-	-	Yes	vpc-0dde4a12dc34212b
<input type="checkbox"/>	My Private RouteTable	rtb-09fd949c2ed82516	subnet-0cb405dae9741a...	-	No	vpc-0852e8296f3fb820a My V...

A new network ACL

By default, my private subnet is associated with the default network ACL of the VPC. This default ACL allows all inbound and outbound traffic, which is why it's important to create a custom ACL to restrict traffic and enhance security.

I set up a dedicated network ACL for my private subnet because the default ACL allows all traffic, which poses security risks. A custom ACL restricts inbound and outbound traffic, enhancing the security of my private subnet.

My new network ACL has two simple rules - one inbound and one outbound rule. Both rules deny all traffic by default. Custom network ACLs start with denying all inbound and outbound traffic. We'll customize them later to allow specific traffic sources

The screenshot shows the AWS Network ACLs management interface. At the top, there is a search bar labeled "Find resources by attribute or tag". Below the search bar is a table titled "Network ACLs (1/4) Info". The table lists four network ACLs:

Name	Network ACL ID	Associated with	Default	VPC ID	Inbound rules count	Outbound rules count
-	arn-0180ebbededcf65f4	-	Yes	vpc-0852e8296f3fb820a / My_VPC	2 Inbound rules	2 Outbound rules
My Public ACL	arn-0ab5b42d287d4b6b5	subnet-0e7af847f642f9df5 / My_Public_Subnet	No	vpc-0852e8296f3fb820a / My_VPC	2 Inbound rules	2 Outbound rules
-	arn-0ea173371bce1bc8	3 Subnets	Yes	vpc-0dde4a12dc34212b	2 Inbound rules	2 Outbound rules
My Private ACL	arn-07129811680d45065	subnet-0cb403dae9741a69f / My_Private_Subnet	No	vpc-0852e8296f3fb820a / My_VPC	1 Inbound rule	1 Outbound rule

Below the table, a modal window is open for "acl-07129811680d45065 / My Private ACL". The modal has tabs for "Details", "Inbound rules" (which is selected), "Outbound rules", "Subnet associations", and "Tags". The "Inbound rules" tab shows a single rule:

Rule number	Type	Protocol	Port range	Source	Allow/Deny
*	All traffic	All	All	0.0.0.0/0	Deny



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

