

# Creating a Private Subnet

Erik Gonzalez

subnet-086b0142574263411 / nimbus-private-subnet Actions ▾

Details	
Subnet ID	<a href="#">subnet-086b0142574263411</a>
IPv4 CIDR	<a href="#">10.0.1.0/24</a>
Availability Zone	<a href="#">use1-az2 (us-east-1b)</a>
Network ACL	-
Auto-assign customer-owned IPv4 address	No
IPv6 CIDR reservations	-
Resource name DNS AAAA record	Disabled
Subnet ARN	<a href="#">arn:aws:ec2:us-east-1:312831649097:subnet/subnet-086b0142574263411</a>
Available IPv4 addresses	251
Network border group	<a href="#">us-east-1</a>
Default subnet	No
Customer-owned IPv4 pool	-
IPv6-only	No
DNS64	Disabled
State	<span>Available</span>
IPv6 CIDR	-
VPC	<a href="#">vpc-0629ca7556832552b   nimbus-vpc</a>
Auto-assign public IPv4 address	No
Outpost ID	-
Hostname type	IP name
Owner	<a href="#">312831649097</a>
Block Public Access	<input checked="" type="radio"/> Off
IPv6 CIDR association ID	-
Route table	<a href="#">rtb-0720941bdb7c35185   Nimbus-Route-Table</a>
Auto-assign IPv6 address	No
IPv4 CIDR reservations	-
Resource name DNS A record	Disabled

# Introducing Today's Project!

## How I used Amazon VPC in this project

In today's project, I used Amazon VPC to create subnets, routing tables, inbound/outbound rules etc..

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

One thing I didn't expect in this project is how a VPC is already created when setting up an AWS account.

## This project took me...

This project took me about one hour to complete.

# Private vs Public Subnets

The difference between public and private subnets is that public subnets are accessible by and can access the internet, while private subnets can be completely isolated from the internet.

Having private subnets is useful because keeping resources away from the internet is extremely important for security and confidential resources/data.

My private and public subnets cannot have the same IPv4 CIDR Block i.e. the same range of IP addresses. The CIDR block for every subnet must be unique so as to not cause any overlap.

subnet-086b0142574263411 / nimbus-private-subnet Actions ▾

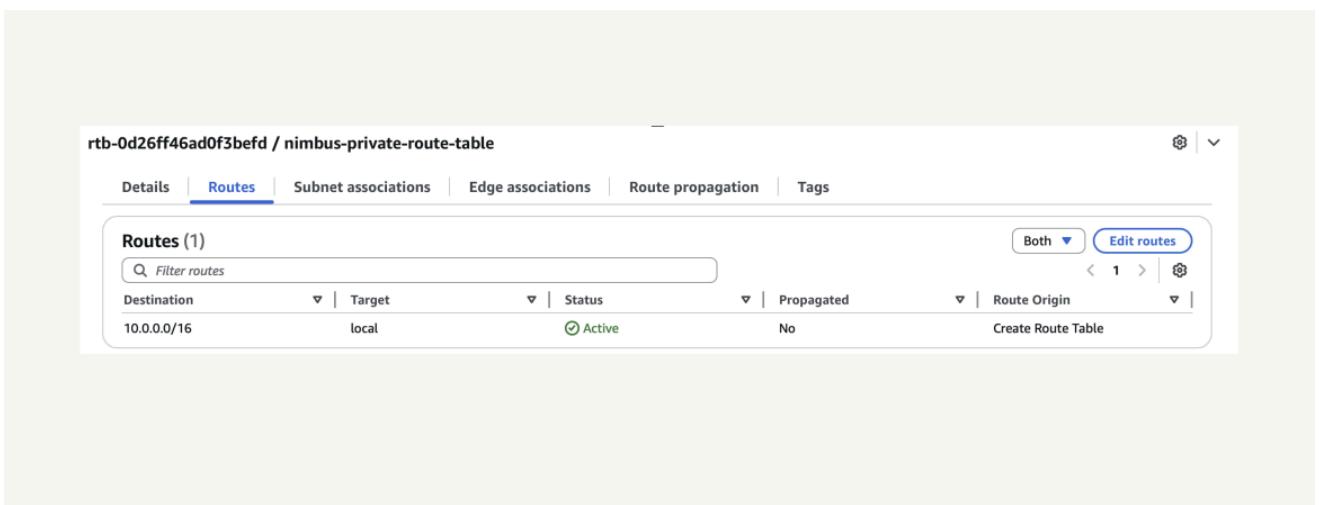
Details	
<b>Subnet ID</b> <a href="#">subnet-086b0142574263411</a>	<b>Subnet ARN</b> <a href="#">arn:aws:ec2:us-east-1:312831649097:subnet/subnet-086b0142574263411</a>
<b>IPv4 CIDR</b> <a href="#">10.0.1.0/24</a>	<b>Available IPv4 addresses</b> <a href="#">251</a>
<b>Availability Zone</b> <a href="#">use1-az2 (us-east-1b)</a>	<b>Network border group</b> <a href="#">us-east-1</a>
<b>Network ACL</b> -	<b>Default subnet</b> No
<b>Auto-assign customer-owned IPv4 address</b> No	<b>Customer-owned IPv4 pool</b> -
<b>IPv6 CIDR reservations</b> -	<b>IPv6-only</b> No
<b>Resource name DNS AAAA record</b> Disabled	<b>DNS64</b> Disabled
	<b>State</b> <a href="#">Available</a>
	<b>IPv6 CIDR</b> -
	<b>VPC</b> <a href="#">vpc-0629ca7556832552b   nimbus-vpc</a>
	<b>Auto-assign public IPv4 address</b> No
	<b>Outpost ID</b> -
	<b>Hostname type</b> IP name
	<b>Owner</b> <a href="#">312831649097</a>
	<b>Block Public Access</b> <a href="#">Off</a>
	<b>IPv6 CIDR association ID</b> -
	<b>Route table</b> <a href="#">rtb-0720941bdb7c35185   Nimbus-Route-Table</a>
	<b>Auto-assign IPv6 address</b> No
	<b>IPv4 CIDR reservations</b> -
	<b>Resource name DNS A record</b> Disabled

# A Dedicated Route Table

By default, my private subnet is associated with the default route table i.e. a route table that has a route to the internet gateway.

I had to set up a new route table because my subnet can't have a route to an internet gateway.

My private subnet's dedicated route table only has one inbound and one outbound rule that allows internal communication i.e. with another destination of another resource within my VPC.



The screenshot shows the AWS Route Table configuration page for a specific route table. The top navigation bar includes tabs for Details, Routes (which is selected), Subnet associations, Edge associations, Route propagation, and Tags. Below the tabs, there is a section titled "Routes (1)" with a table containing one row of data. The table columns are Destination, Target, Status, Propagated, and Route Origin. The single row shows a Destination of "10.0.0.0/16", a Target of "local", a Status of "Active", and a Propagated value of "No". There is also a "Create Route Table" button. At the bottom right of the table area, there are buttons for "Both" and "Edit routes".

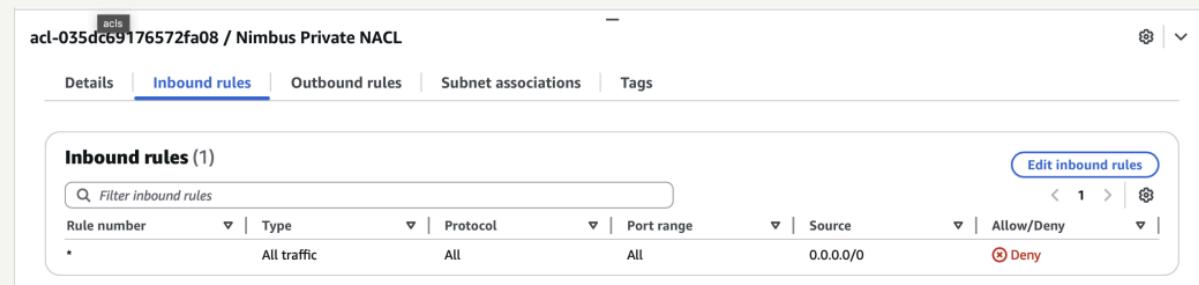
Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	Create Route Table

# A New Network ACL

By default, my private subnet is associated with the default network ACL that's set up for every VPC created in my AWS account.

I set up a dedicated network ACL for my private subnet because a network ACL becomes crucial in the event of security breaches—where traffic that has compromised my public subnet could access my private subnet if my network ACL rules allow all inbound and outbound traffic.

My new network ACL has two simple rules – deny all inbound traffic and deny all outbound traffic.



The screenshot shows the AWS Network ACL management interface. The top navigation bar includes 'acls' and the specific ACL name 'acl-035dc69176572fa08 / Nimbus Private NACL'. Below the navigation are tabs for 'Details', 'Inbound rules' (which is selected and highlighted in blue), 'Outbound rules', 'Subnet associations', and 'Tags'. The 'Inbound rules' section displays one rule: 'Inbound rules (1)'. A table lists the rule details: Rule number is '\*', Type is 'All traffic', Protocol is 'All', Port range is 'All', Source is '0.0.0.0/0', and Allow/Deny is 'Deny'. To the right of the table are buttons for 'Edit inbound rules' and navigation controls ('< 1 >').