



# Build a Virtual Private Cloud



Gabriel Taveira Mazer

[VPC](#) > [Your VPCs](#) > Create VPC

## 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.

GabrielMazer VPC

**IPv4 CIDR block** [Info](#)

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

**IPv4 CIDR**

10.0.0.0/16

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 ▼



**Gabriel Mazer**

# Introducing Today's Project!

## What is Amazon VPC?

Amazon VPC (Virtual Private Cloud) is a private network that allows you to isolate and manage your resources in AWS, offering control over security, traffic flow, and network settings, making it essential for cloud infrastructure.

## How I used Amazon VPC in this project

In today's project, I used Amazon VPC to create a custom virtual network, set up subnets, and attached an internet gateway, enabling controlled access between my resources and the internet.

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

I didn't expect how straightforward and visual the process of configuring a VPC would be, especially with the guided steps and AWS console layout.

## This project took me...

This project took me about 2 hours to complete, including reading the instructions, creating the VPC, subnets, and configuring the internet gateway.



# Virtual Private Clouds (VPCs)

VPCs are isolated virtual networks in AWS that allow you to control and manage resources, such as EC2 instances, with custom security, IP addressing, and networking configurations.

There was already a default VPC in my account ever since my AWS account was created. This is because AWS provides a default VPC for quick resource deployment without needing custom networking setup.

To set up my VPC, I had to define an IPv4 CIDR, which means specifying a range of IP addresses that can be assigned to resources within the VPC, ensuring no overlap with other networks.

VPC > Your VPCs > Create VPC

## 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.

GabrielMazer VPC

**IPv4 CIDR block** [Info](#)  
☒ IPv4 CIDR manual input  
☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR  
10.0.0.0/16  
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

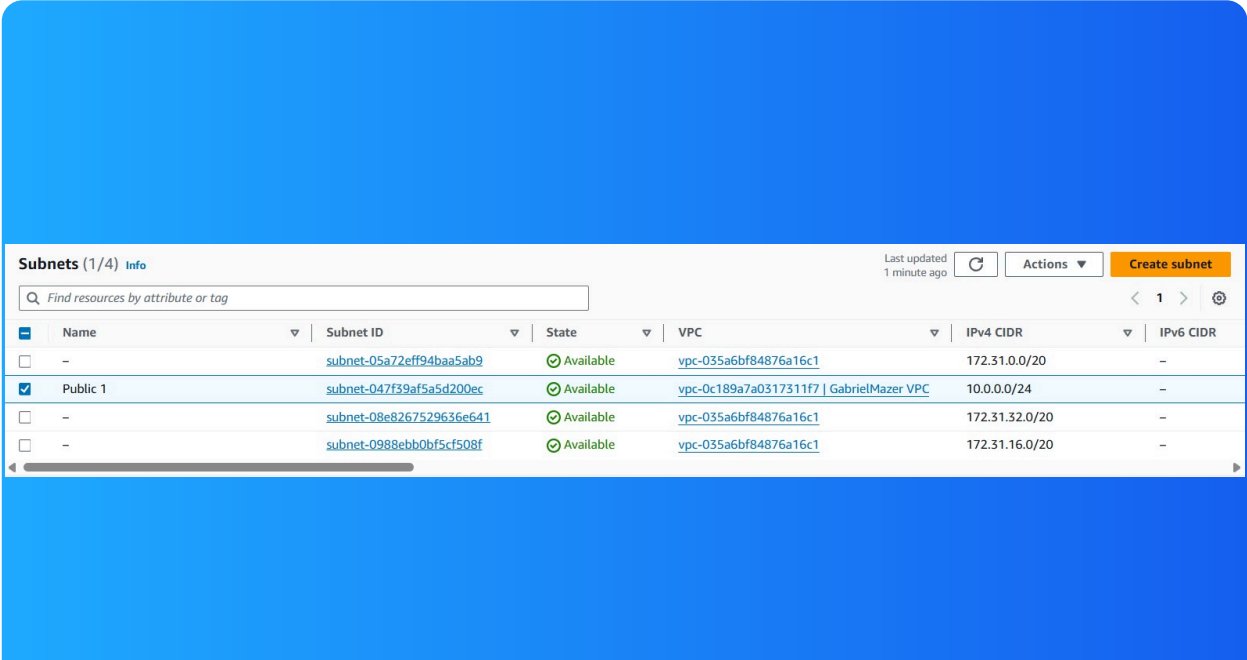


# Subnets

Subnets are subdivisions of a VPC that allow you to segment your network. Each subnet hosts resources that can communicate within the subnet and with other subnets, while controlling traffic through network access control.

There are already subnets existing in my account, one for every Availability Zone in the selected AWS region, allowing me to launch resources without needing to manually create subnets from scratch.

I named my subnet Public 1, but that doesn't automatically make my subnet a public subnet. For a subnet to be considered public, it has to be associated with an Internet Gateway and have a route to the internet.



The screenshot shows the AWS Management Console interface for the 'Subnets' page. At the top, there's a header with 'Subnets (1/4)' and an 'Info' link. Below this is a search bar with the placeholder text 'Find resources by attribute or tag'. To the right of the search bar, there's a 'Last updated 1 minute ago' timestamp, a refresh icon, an 'Actions' dropdown menu, and a 'Create subnet' button. The main content area is a table with columns: Name, Subnet ID, State, VPC, IPv4 CIDR, and IPv6 CIDR. The table contains four rows. The second row, 'Public 1', is selected with a checkbox. The table is set to show 1 page of results.

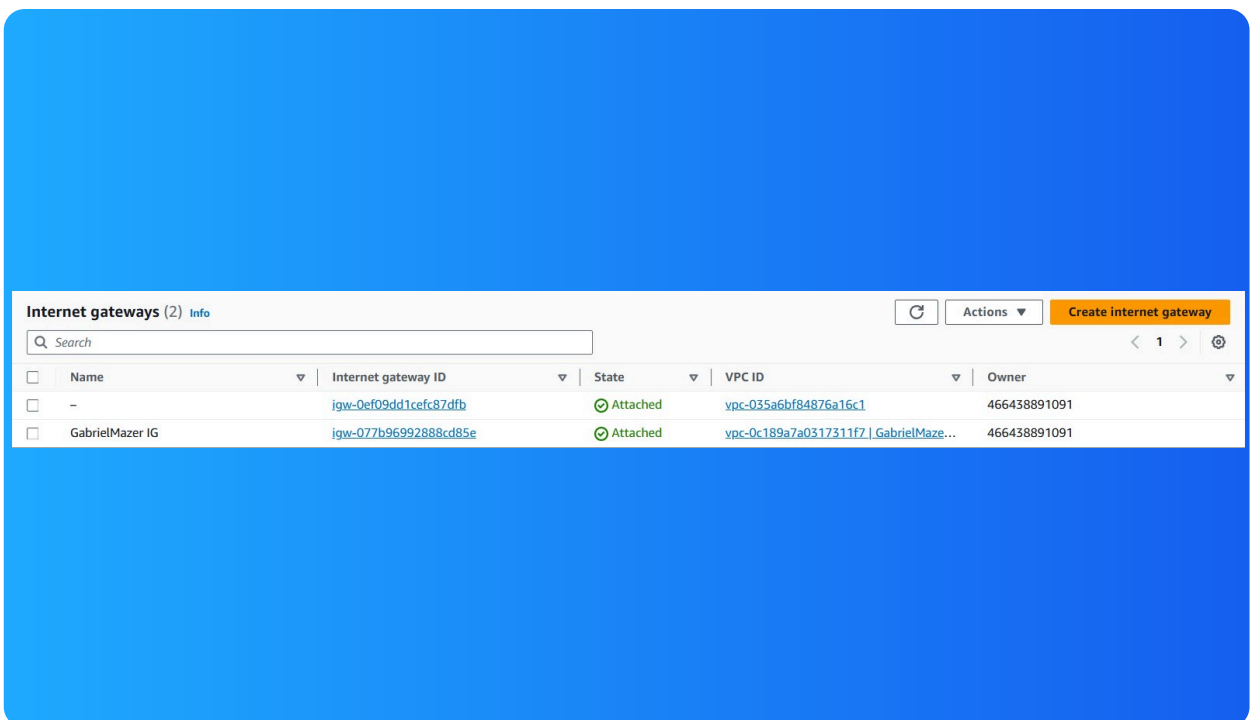
Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR
-	subnet-05a72eff94baa5ab9	Available	vpc-035a6bf84876a16c1	172.31.0.0/20	-
Public 1	subnet-047f39af5a5d200ec	Available	vpc-0c189a7a0317311f7   GabrielMazer VPC	10.0.0.0/24	-
-	subnet-08e8267529636e641	Available	vpc-035a6bf84876a16c1	172.31.32.0/20	-
-	subnet-0988ebb0bf5cf508f	Available	vpc-035a6bf84876a16c1	172.31.16.0/20	-



# Internet gateways

Internet gateways are horizontally scaled, redundant, and highly available gateways that allow communication between resources within a VPC and the internet.

Attaching an internet gateway to a VPC means the resources within the VPC can send and receive data to and from the internet, enabling public access when properly configured.



Internet gateways (2) <a href="#">Info</a>						<a href="#">Refresh</a>	<a href="#">Actions</a>	<a href="#">Create internet gateway</a>
<input type="text" value="Search"/>						<a href="#">1</a>		
<input type="checkbox"/>	Name	Internet gateway ID	State	VPC ID	Owner			
<input type="checkbox"/>	-	<a href="#">igw-0ef09dd1cefc87dfb</a>	Attached	<a href="#">vpc-035a6bf84876a16c1</a>	466438891091			
<input type="checkbox"/>	GabrielMazer IG	<a href="#">igw-077b96992888cd85e</a>	Attached	<a href="#">vpc-0c189a7a0317311f7   GabrielMaze...</a>	466438891091			



# VPC Traffic Flow and Security



Gabriel Taveira Mazer

Security group (sg-0b8bdda5dab64472f | GabrielMazer Security Group) was created successfully

Details

VPC > Security Groups > sg-0b8bdda5dab64472f - GabrielMazer Security Group

sg-0b8bdda5dab64472f - GabrielMazer Security Group

Actions

Details

Security group name GabrielMazer Security Group	Security group ID sg-0b8bdda5dab64472f	Description A Security Group for the GabrielMazer VPC.	VPC ID vpc-0c189a7a0317311f7
Owner 466438891091	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Inbound rules Outbound rules Tags

Inbound rules (1)

Search

Manage tags Edit inbound rules

	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sgr-0a72208eb5b7d2224	IPv4	HTTP	TCP	80	0.0.0.0/0	-



# Route tables

Route tables are a set of rules that direct network traffic. They guide data on where to go within or outside a VPC, like a GPS for resources in a subnet.

Route tables are needed to make a subnet public because they must include a route directing traffic to an internet gateway. This allows communication with external networks.

Updated routes for rtb-0752217034b58a042 / NextWork route table successfully

VPC > Route tables > rtb-0752217034b58a042

rtb-0752217034b58a042 / NextWork route table

Actions

Details Info

Route table ID  
rtb-0752217034b58a042

VPC  
vpc-0c189a7a0317311f7 | GabrielMazer VPC

Main  
Yes

Owner ID  
466438891091

Explicit subnet associations  
-

Edge associations  
-

Routes Subnet associations Edge associations Route propagation Tags

Routes (2)

Filter routes

Both Edit routes

< 1 > ⌕

Destination	Target	Status	Propagated
0.0.0.0/0	igw-077b96992888cd85e	Active	No
10.0.0.0/16	local	Active	No



Gabriel Mazer

# Route destination and target

Routes are defined by their destination (the IP range traffic wants to reach) and target (the path or resource traffic will use to get there).

The route in my route table that directed internet-bound traffic had a destination of 0.0.0.0/0 (all IPs) and a target of the internet gateway.

Updated routes for rtb-0752217034b58a042 / NextWork route table successfully  
Details

VPC > Route tables > rtb-0752217034b58a042

rtb-0752217034b58a042 / NextWork route table

Details info

Route table ID  
rtb-0752217034b58a042

VPC  
vpc-0c189a7a0317311f7 | GabrielMazer VPC

Main  
Yes

Owner ID  
466438891091

Explicit subnet associations  
-

Edge associations  
-

Routes

Subnet associations

Edge associations

Route propagation

Tags

Routes (2)

Filter routes

Both Edit routes

< 1 > ⌕

Destination	Target	Status	Propagated
0.0.0.0/0	igw-077b96992888cd85e	Active	No
10.0.0.0/16	local	Active	No





# Security groups

Security groups are virtual firewalls that control incoming and outgoing traffic for resources in a VPC, based on defined rules.

## Inbound vs Outbound rules

Inbound rules are settings that allow specific traffic to enter a resource. I configured an inbound rule that allows HTTP traffic (port 80) from anywhere (IPv4).

Outbound rules are settings that allow specific traffic to leave a resource. By default, my security group's outbound rule allows all traffic to leave the instance.

Security group (sg-0b8bda5dab64472f | GabrielMazer Security Group) was created successfully

Details

VPC > Security Groups > sg-0b8bda5dab64472f - GabrielMazer Security Group

sg-0b8bda5dab64472f - GabrielMazer Security Group

Details

Security group name

GabrielMazer Security Group

Security group ID

sg-0b8bda5dab64472f

Description

A Security Group for the GabrielMazer VPC.

VPC ID

vpc-0c189a7a0317311f7

Owner

466438891091

Inbound rules count

1 Permission entry

Outbound rules count

1 Permission entry

Inbound rules

Outbound rules

Tags

Inbound rules (1)

Manage tags

Edit inbound rules

Search

	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sg-0a72208eb5b7d2224	IPv4	HTTP	TCP	80	0.0.0.0/0	-



**Gabriel Mazer**

# Network ACLs

Network ACLs are like community watchmen that secure traffic at the subnet level in a VPC, controlling inbound/outbound traffic for all resources in the subnet.

## Security groups vs. network ACLs

The difference between a security group and a network ACL is that security groups secure resources at the resource level, while network ACLs secure traffic at the subnet level.



# Default vs Custom Network ACLs

## Similar to security groups, network ACLs use inbound and outbound rules

By default, a network ACL's inbound and outbound rules will allow all traffic in both directions for any protocol.

In contrast, a custom ACL's inbound and outbound rules are automatically set to deny all traffic until specific allow rules are added.

The screenshot displays the AWS Management Console interface for Network ACLs. At the top, a green notification bar states: "You have successfully updated subnet associations for acl-0da7b3106cbe776b9 / GabrielMazer Network ACL." Below this, the "Network ACLs (1/3)" section shows a table of ACLs. The table has columns for Name, Network ACL ID, Associated with, Default, VPC ID, Inbound rules count, Outbound rules count, and Owner. The ACL "acl-0da7b3106cbe776b9" is highlighted, showing it is associated with "subnet-047f59af5a5d200ec / Public 1" and has 2 inbound and 2 outbound rules. Below the table, the "Inbound rules (2)" section is expanded, showing a table of rules. The table has columns for Rule number, Type, Protocol, Port range, Source, and Allow/Deny. Two rules are listed: Rule 100, which allows all traffic from 0.0.0.0/0, and a default rule that denies all traffic from 0.0.0.0/0.

Name	Network ACL ID	Associated with	Default	VPC ID	Inbound rules count	Outbound rules count	Owner
-	acl-0950080aa97f293c	-	Yes	vpc-0c189a7a0317311f7 / GabrielM...	2 Inbound rules	2 Outbound rules	466438891091
-	acl-0128d7fcbfeef9b07	3 Subnets	Yes	vpc-035a6bf84876a16c1	2 Inbound rules	2 Outbound rules	466438891091
GabrielMazer Netwo...	acl-0da7b3106cbe776b9	subnet-047f59af5a5d200ec / Public 1	No	vpc-0c189a7a0317311f7 / GabrielM...	2 Inbound rules	2 Outbound rules	466438891091

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