

Table of contents

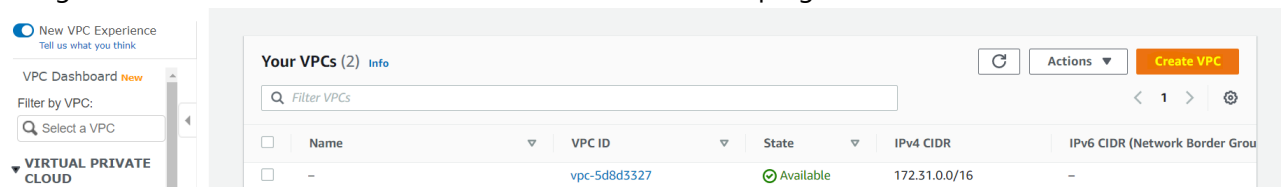
- [Table of contents](#)
- [Defining your VPC :](#)
 - [Creating VPC](#)
 - [Creating subnet under our VPC](#)
 - [Route tables](#)
 - [Test run of our new setup](#)
 - [Internet gateway](#)
 - [Test run of our new setup](#)
 - [NACL](#)

Defining your VPC :

- Last time we tried understanding the default VPC and all the components inside it .
- Agenda for today will be trying to replicate the exact same settings in our own VPC

Creating VPC

- In order to start with our process , we need to create a VPC first .
- A VPC needs to have a private CIDR , decide it beforehand before initiating the create process
- We will try our hand with "10.0.0.0/16"
- Navigate to VPC dashboard , and click on "Create VPC " on top right corner



- After clicking, enter the name , CIDR block and tags if required . Tenancy here is similar to which we saw earlier with EC2 . Dedicated tenancy ensure that the instances launched in the VPC are deployed on the dedicated hardware . Let us keep it at its default value which is "Default".

- Click on create VPC to create the VPC .

Creates a tag with a key of 'Name' and a value that you specify.

MyFirstVPC

IPv4 CIDR block [Info](#)

10.0.0.0/16

IPv6 CIDR block [Info](#)

☒ No IPv6 CIDR block

☐ Amazon-provided IPv6 CIDR block

☐ IPv6 CIDR owned by me

Tenancy [Info](#)

Default

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 | Value - optional | |
|--------|------------------|--------|
| Q Name | Q MyFirstVPC | Remove |

[Add new tag](#)

You can add 49 more tags.

Cancel [Create VPC](#)

- We should now be able to see our new VPC under the "Your VPCs" section .
- Going forward we'll be using the same VPC for our operations.

Creating subnet under our VPC

- Since we have a VPC setup, we can create the subsets of the VPC i.e. subnets .
- Subnets let us controle the CIDR , availability zone and auto assign publiip ip settings .
- Let us create 2 subnets . Remember these should be a subset of the VPC , also these shoudnt be colliding with each other. For ex
 - Subnet A : 10.0.0.0/28
 - Subnet B : 10.0.0.16/28
- Let us go ahead and create these subnets . Before we do that , as a best practice select our own VPC from the filter in upper left corner . This is to ensure that all our operations are being under appropriate VPC

New VPC Experience
Tell us what you think

[Create subnet](#) [Actions](#)

VPC Dashboard [New](#)

Filter by VPC:

Q vpc-090e1...

vpc-090e17630b861c7ee
MyFirstVPC
Owner: 384395217903

VIRTUAL PRIVATE CLOUD

Your VPCs [New](#)

[Subnets](#)

Filter by tags and attributes or search by keyword

You do not have any Subnets in this region

Click the Create Subnet button to create your first Subnet

[Create subnet](#)

- Let us click on create subnet to get started with the subnet creation .
- We will have to enter the name , the VPC id,availability zone and the CIDR which we want to have .
 - Let us name our subnet as subnet-A
 - Select our newly created VPC
 - Select the availability zone as "us-east-1a"
 - Enter the CIDR that we have decided previously i.e "10.0.0.0/28"

Subnets > Create subnet

Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag ⓘ

VPC* ⓘ

Availability Zone ⓘ

| VPC CIDRs | CIDR | Status | Status Reason |
|-----------|-------------|------------|---------------|
| | 10.0.0.0/16 | associated | |

IPv4 CIDR block* ⓘ

* Required

Cancel Create

- Repeat the same process for subnet-B as well .
 - Let us name this subnet as subnet-B
 - Select the appropriate VPC
 - For this subnet , we will select availability zone as "us-east-1b"
 - Enter the second CIDR i.e "10.0.0.16/28"

Subnets > Create subnet

Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag ⓘ

VPC* ⓘ

Availability Zone ⓘ

| VPC CIDRs | CIDR | Status | Status Reason |
|-----------|-------------|------------|---------------|
| | 10.0.0.0/16 | associated | |

IPv4 CIDR block* ⓘ

* Required

Cancel Create

- We should now be able to see the newly created subnets under the subnet tab in our VPC

New VPC Experience
Tell us what you think

VPC Dashboard **New**

Filter by VPC:

vpc-090e17630b861c7ee
MyFirstVPC
Owner: 384395217903

VIRTUAL PRIVATE CLOUD

Your VPCs **New**

Subnets

Route Tables

Internet Gateways **New**

Egress Only Internet Gateways **New**

Carrier Gateways **New**

DHCP Options Sets **New**

Elastic IPs **New**

Managed Prefix Lists **New**

Create subnet Actions

Filter by tags and attributes or search by keyword

| Name | Subnet ID | State | VPC | IPv4 CIDR | Available IPv4 | IPv6 CIDR | Availability Zone | Availab |
|----------|--------------------------|-----------|---------------------------|--------------|----------------|-----------|-------------------|---------|
| subnet-B | subnet-08ef3122762dbabca | available | vpc-090e17630b861c7ee ... | 10.0.0.16/28 | 11 | - | us-east-1b | use1-az |
| subnet-A | subnet-0f606fbc5d0bfe50d | available | vpc-090e17630b861c7ee ... | 10.0.0.0/28 | 11 | - | us-east-1a | use1-az |

Subnet: subnet-0f606fbc5d0bfe50d

Description Flow Logs Route Table Network ACL Tags Sharing

| Property | Value | Property | Value |
|---|------------------------------------|---------------------------------|---------------------|
| Subnet ID | subnet-0f606fbc5d0bfe50d | State | available |
| VPC | vpc-090e17630b861c7ee MyFirstVPC | IPv4 CIDR | 10.0.0.0/28 |
| Available IPv4 Addresses | 11 | IPv6 CIDR | - |
| Availability Zone | us-east-1a (use1-az6) | Network Border Group | us-east-1 |
| Route Table | rtb-0aa6aa07706487ef8 | Network ACL | acl-0f79cf93420c385 |
| Default subnet | No | Auto-assign public IPv4 address | No |
| Auto-assign customer-owned IPv4 address | No | Customer-owned IPv4 pool | - |
| Auto-assign IPv6 address | No | Outpost ID | - |
| Owner | 384395217903 | | |

- Notice that the setting "auto assign public ip" as by default set to "No"
- Since we are yet in initiation phase , we want our instances to have public ip . Even though while launching the instance , this setting can be changed . Changing it here makes much more easier to use .
- In order to change it , select the subnet and click on actions
- We'll see an option called "Modify auto-assign public ip settings ". Post clicking we should see the screen with options as below

Subnets > Modify auto-assign IP settings

Modify auto-assign IP settings

Enable the auto-assign IP address setting to automatically request a public IPv4 or IPv6 address for an instance launched in this subnet. You can override the auto-assign IP settings for an instance at launch time.

Subnet ID subnet-08ef3122762dbabca

Auto-assign IPv4 ☒ Enable auto-assign public IPv4 address ⓘ

Auto-assign Co-IP ☐ Enable auto-assign customer-owned IPv4 address ⓘ

* Required

Cancel Save

- As seen above , let us click on "Enable auto-assign public ipv4 address "
- The other option stated "Enable auto-assign customer-owned IPv4 address" is meant if we own a set of public ip addresses and want to use it . Since we do not own any , let us keep it blank .
- Click on save and proceed with the same approach for other subnet

Route tables

- Notice that there in one route table already created here . This is what we refer to as main route table
- When we create a new VPC , main route table is always created with it

New VPC Experience
Tell us what you think

Create route table Actions

Filter by tags and attributes or search by keyword

| Name | Route Table ID | Explicit subnet association | Edge associations | Main | VPC ID | Owner |
|------|-----------------------|-----------------------------|-------------------|------|---------------------------|--------------|
| | rtb-0aa6aa07706487ef8 | - | - | Yes | vpc-090e17630b861c7ee ... | 384395217903 |

Route Table: rtb-0aa6aa07706487ef8

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Edit routes

View All routes

| Destination | Target | Status | Propagated |
|-------------|--------|--------|------------|
| 10.0.0.0/16 | local | active | No |

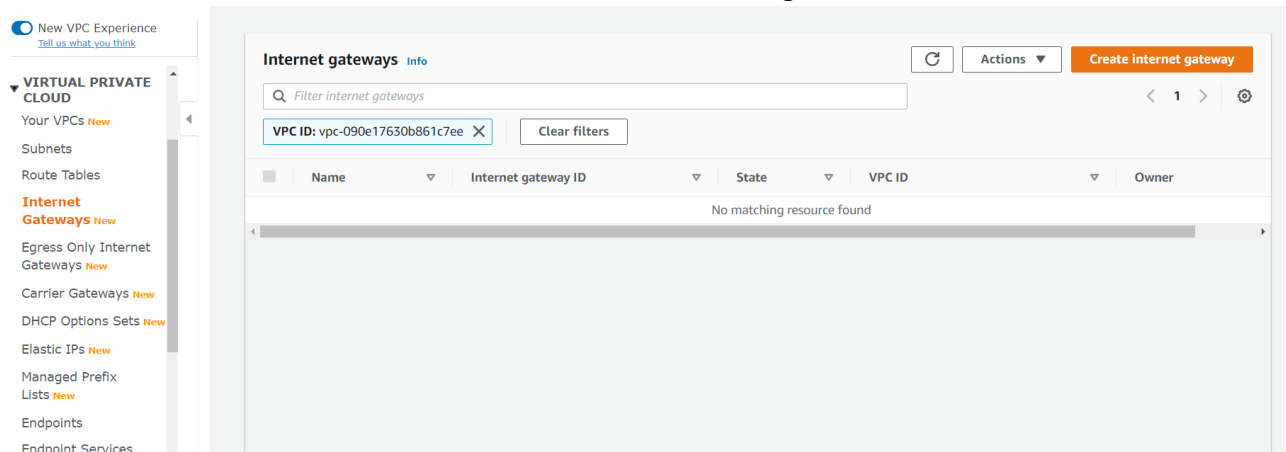
- Observe that under "routes" there is already a route added . We can see the route of our VPC is already added . That is a default route that gets added whenever any route table is created .
- Since we are planning to have all the public instances , we will use the same route table .

Test run of our new setup

- Let us launch a new instance in our VPC in any of the subnets.
- Test if you can connect to your instance once it is launched .
- Since we are yet to attach the internet gateway , the connection wont be established .

Internet gateway

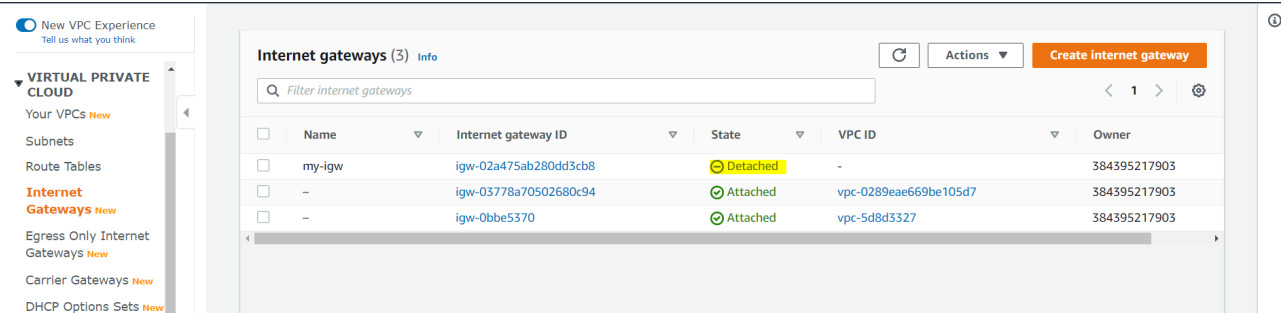
- We have seen that the internet gateway is like a highway for our postal codes . i.e. allows our subnets to connect to the internet .
- In order to have that feature available , we'll need to create an igw .



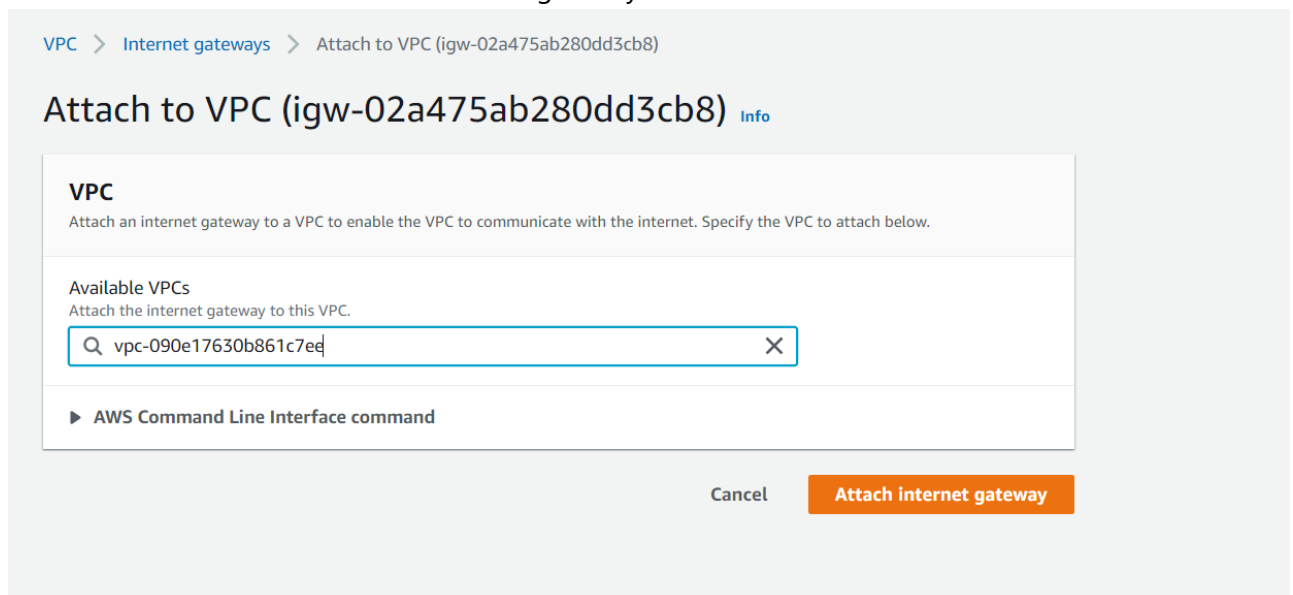
- Note that an igw is not created by default with a VPC , it needs to be created explicitly .
- Let us now create a igw by clicking on the upper right option of "Create internet gateway "
- We only have to provide the name for the igw and click on create

- Since we have the VPC filter set on top left corner , we wont be able to see the igw as it is yet to be attached to our VPC .

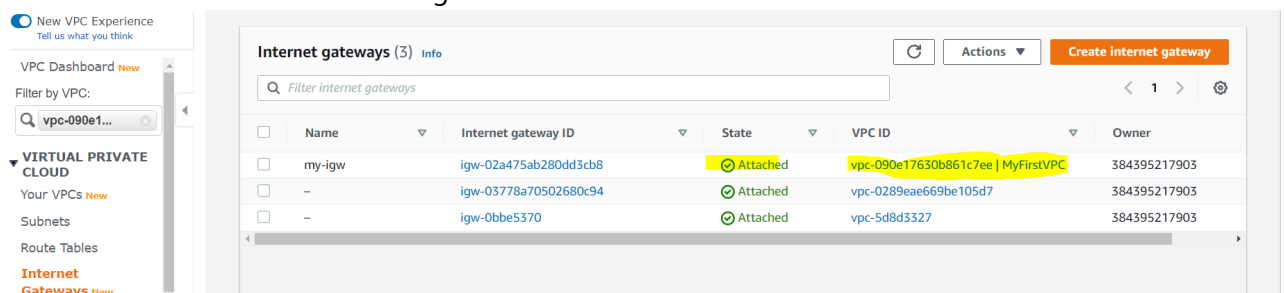
- Remove the filter temporarily and we should be able to see the igw
- Notice our newly created igw is in "detached " state .



- Let us select the igw and click on actions .
- Select "Attach to VPC "
- select our VPC and click on attach internet gateway .



- We should now be able to see the igw is in "Attached " state



- Since the igw is attached ,it is safe for us to say that the highway has been created .
- But there will not be any use of this highway unless there are roads connecting to it . which can allow postal codes to connect to that highway
- Similarly subnets need a route to this igw. Same could be added in route table
- Let us go to the route table and select the one which has been created under our VPC
- Click on routes and select edit routes

- Under destination add "0.0.0.0/0" which indicates the internet
- For target select "internet gateway " option and that will prompt available igw . Select the one which we have created

[Route Tables](#) > Edit routes

Edit routes

| Destination | Target | Status | Propagated |
|-------------|-----------------------|--------|------------|
| 10.0.0.0/16 | local | active | No |
| 0.0.0.0/0 | igw-02a475ab280dd3cb8 | | No |

Add route

* Required

Cancel Save routes

Test run of our new setup

- Since now we have an igw which could be used by our subnets , test the connectivity to the instance
- We should now be able to connect to the instance. Log in and check if it has internet connectivity by using below command

```
ping www.google.com
```

NACL

- Similar to route table , NACL is created by default with the VPC .

New VPC Experience
Tell us what you think

Create network ACL Actions

Filter by tags and attributes or search by keyword

| Name | Network ACL ID | Associated with | Default | VPC | Owner |
|------|-----------------------|-----------------|---------|------------------------------------|--------------|
| | acl-0f79cff93f420c385 | 2 Subnets | Yes | vpc-090e17630b861c7ee MyFirstVPC | 384395217903 |

Network ACL: acl-0f79cff93f420c385

Details Inbound Rules Outbound Rules Subnet associations Tags

Network ACL ID: acl-0f79cff93f420c385
Associated with: 2 Subnets
Owner: 384395217903

Default: Yes
VPC: vpc-090e17630b861c7ee | MyFirstVPC

- Notice that since it is default , both the subnets we created in the VPC are associated with it

| Rule # | 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 |

- Let us also not that inbound rules , we can see a rule with # 100 which allows all traffic from anywhere
- Hence we can conclude , the default NACL by default allows all access
- However , if you create a new NACL , it will not allow any traffic .
- Let us try and create a new NACL and use it .
- Click on create NACL

[Network ACLs](#) > Create network ACL

Create network ACL

A network ACL is an optional layer of security that acts as a firewall for controlling traffic in and out of a subnet.

Name tag

VPC*

* Required

[Cancel](#) [Create](#)

- Lets name it NACL-A , select the VPC and create it

| Rule # | Type | Protocol | Port Range | Source | Allow / Deny |
|--------|-------------|----------|------------|-----------|--------------|
| * | ALL Traffic | ALL | ALL | 0.0.0.0/0 | DENY |

- As stated earlier , we can observe from the inbound rules that it is not allowing access from anywhere
- We can also see that there are no subnets associated with it .
- Let us go ahead and attach subnet-A to it

- Select the NACL and go to actions
- Select edit subnet association

[Network ACLs](#) > Edit subnet associations

Edit subnet associations



- Select subnet-A and click on edit .
- This basically translates that any of the resources launched in the subnet will be having controlled traffic from this NACL
- Note that one subnet at a time can only be associated with one NACL
- Let us now edit the inbound rules to test the access .
- Things to test
 - Start with rule #100 which allows all traffic from anywhere
 - Then create additional rule #99 which denies any traffic from one of your friend's ip
 - Try and create a rule #101 which allows that ip
 - Experiment with different set of combinations to test how the priority works along with allow/deny actions
- **Points to consider**
 - Post practice make sure to delete the components inside the VPC , i.e. igw ,NACL etc before deleting the VPC. Without that VPC will not be deleted
 - Keep track of the resources created under a VPC
 - NACL can be kept open to all ,however should be used to block known and malicious traffic
 - Use multiples of 100 while adding rules in NACL. High priority or critical rules can be added as #99 etc