**Amazon VPC**

Amazon Virtual Private Cloud (Amazon VPC) provides a secure and isolated section within the AWS Cloud. In this guide, we will create a VPC with the IP range 10.0.0.0/16, consisting of two subnets - one for **web servers (public)** and another for **database servers (private)**. The web server will be accessible from the internet, while the database server will remain isolated.

IP Range: 10.0.0.0/16.

32 – 16 = 16.

2 power 16 is 65536.

So, in our virtual private cloud we can have only 65536 machines only.

By default, subnets are private.

**Practical**

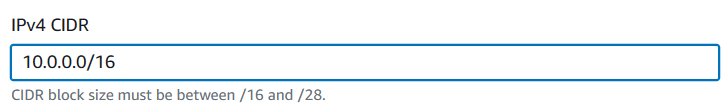
Select Mumbai Region.

**Step 1:** Create VPC

* Region: Mumbai.
* Services > Network and content delivery > VPC.

We have some default VPC, Default subnet, Default Route tables.

* Name: MyVPC.
* CIDR Block: 10.0.0.0/16.
* Create VPC.



(Note: Our VPC can have maximum 2 power 16 machines)

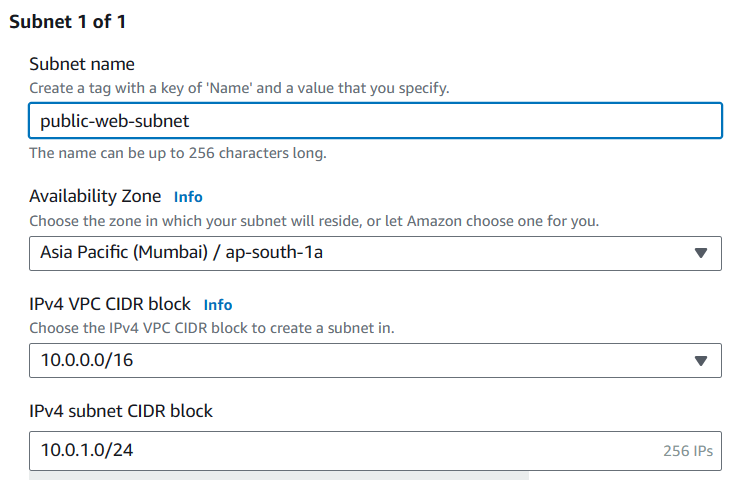
Create VPC

**Step 2: Create Subnet for Web Servers**

* In MyVPC > Subnets > Create subnet.

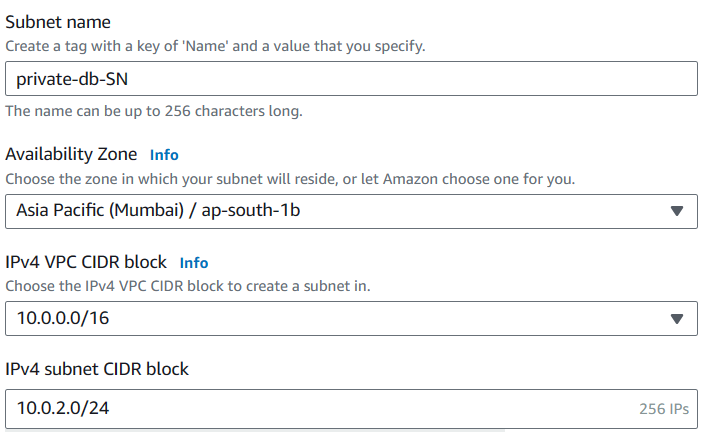


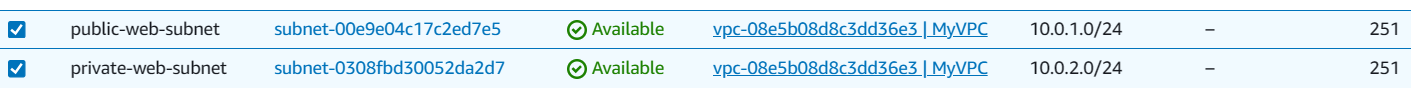
* Name: public-web-SN.
* VPC: MyVPC.
* Availability Zone: ap-southeast-1a (choose any zone).
* CIDR Block: 10.0.1.0/24. (32 -24 = 8, which is 2 power 8 is 256 machiness)
* Name Tag: 10.0.1.0/24 - ap-southeast-1a.
* Create.



**Step 3:** Create Subnet for Database Servers:

* In MyVPC -> Subnets -> Create subnet.
* Name: private-db-SN.
* VPC: MyVPC.
* Availability Zone: ap-southeast-1b (choose a different zone).
* CIDR Block: 10.0.2.0/24.
* Name Tag: 10.0.2.0/24 - ap-southeast-1b.
* Create.





**Observe:** Available IPV4 column it is showing as 251

But we should get 2 power 8(256)

That means 5 ip address are missing.

**Note:** In every subnet, 5 IP address are reserved.

You can search in google "Reserved IP address in AWS"

Select VPC and subnet sizing. We can see the list of IPs which are reserved.

10.0.0.0

10.0.0.1

10.0.0.2

10.0.0.3

10.0.0.255

**Note:** Every subnet will be private by default.

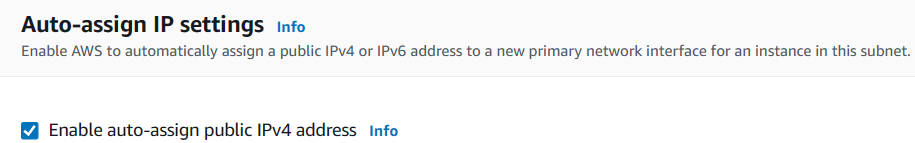
We want to make one subnet as public.

To make subnet public its two-step process.

**Step 1:** **Configuring public subnet**

Select the subnet(10.0.1.0/24) > Actions > Edit subnet settings

> Enable Auto Assigning public IPV4 Address – Save

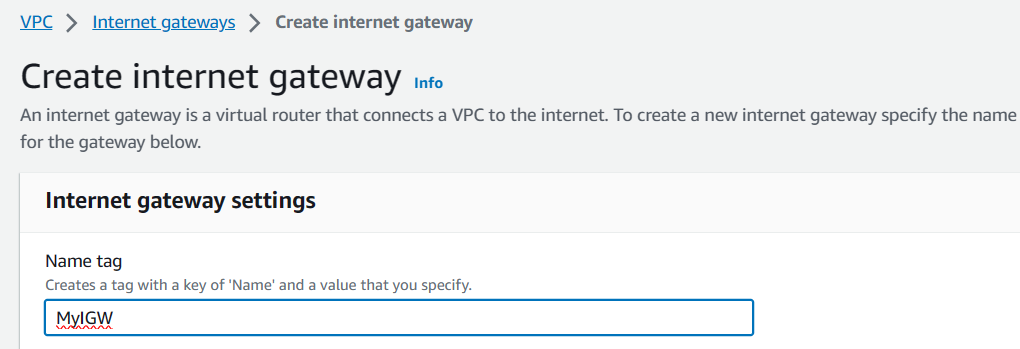


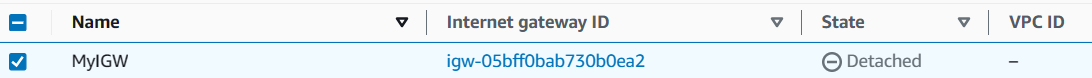
(From now, public IP will be assigned to the machines in this Subnet)

**Step 2:** **Connect to Internet Gateway via Route Table**.

We have one default internet gateway. Do not disturb this.

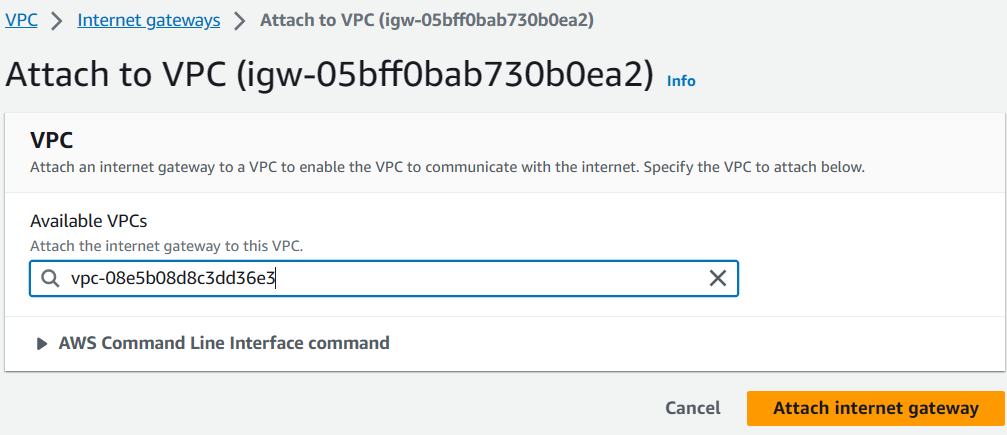
Create internet gateway.





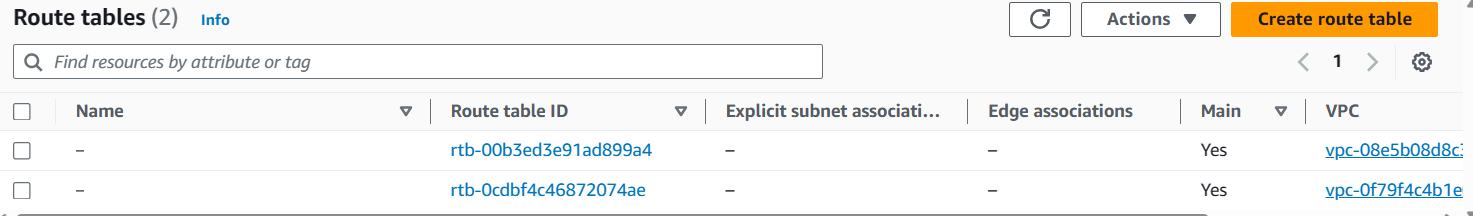
**Observation:** Status is detached.

Attach internet gateway to VPC.



We cannot attach Internet Gateway to subnet directly. For that we create new Route table.

Select Route tables



**Observation:** We have two route tables.

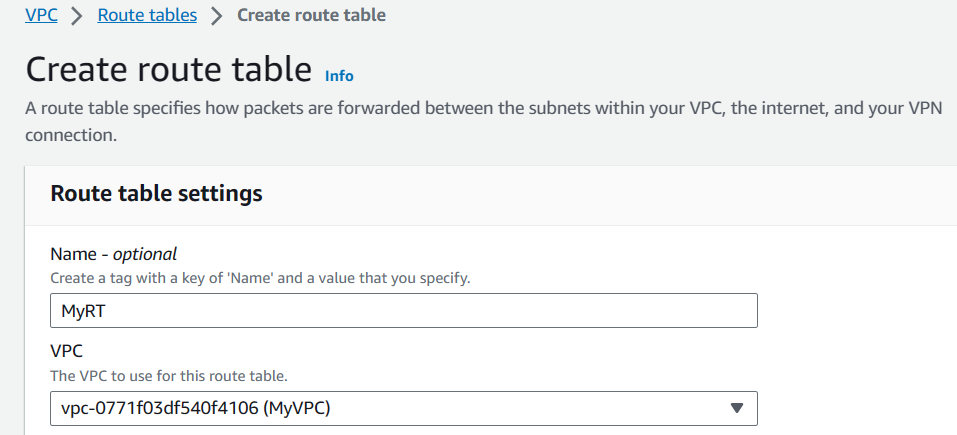
One route table attached to default VPC.

Another route table attached to our VPC (MyVPC)

This means, one default route table is created automatically, when we create our own VPC.

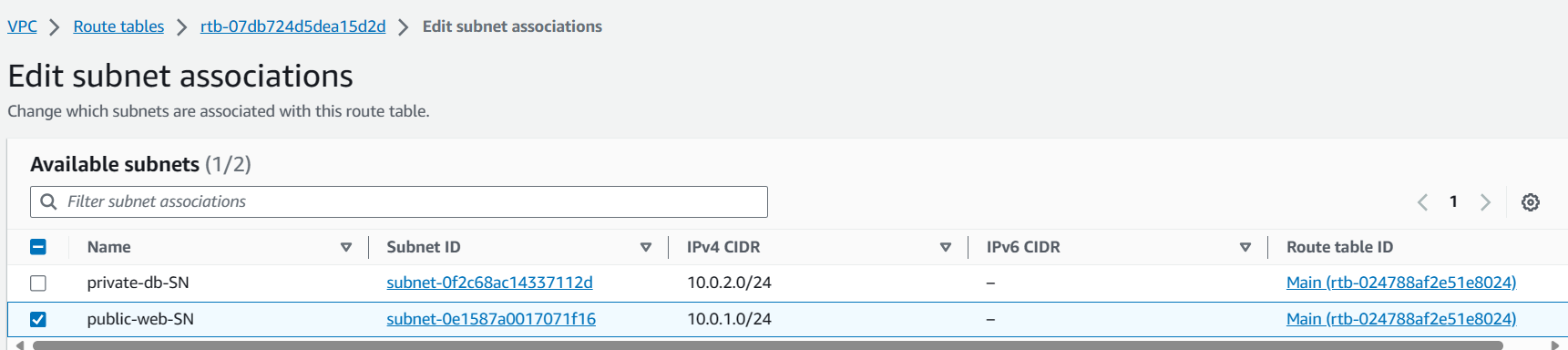
**Creating new route table**

Create Route table > Name Tag: InternetRT > VPC: MyVPC > Create



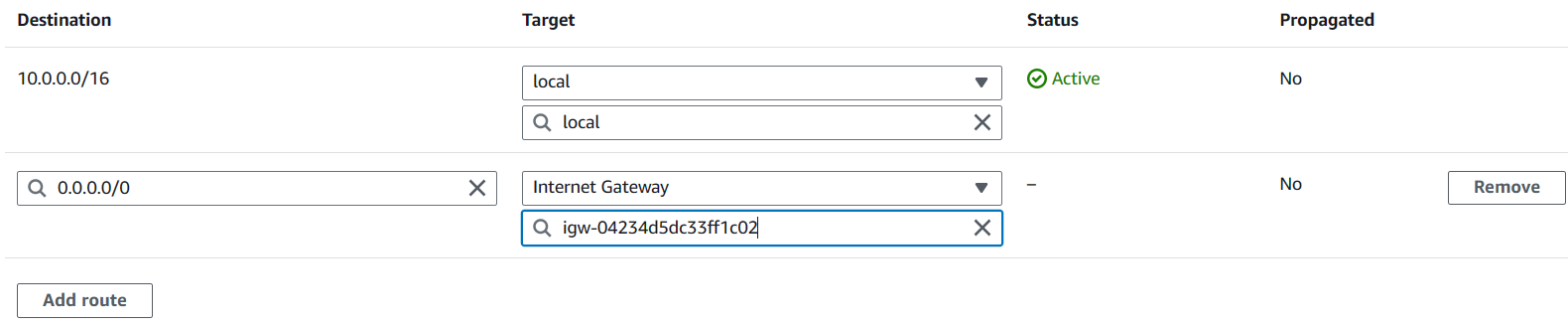
Now, we need to connect Route table to Subnet.

Select the route table (InternetRT) > Actions > Edit Subnet Associations > Select the subnet (10.0.1.0/24) > Save

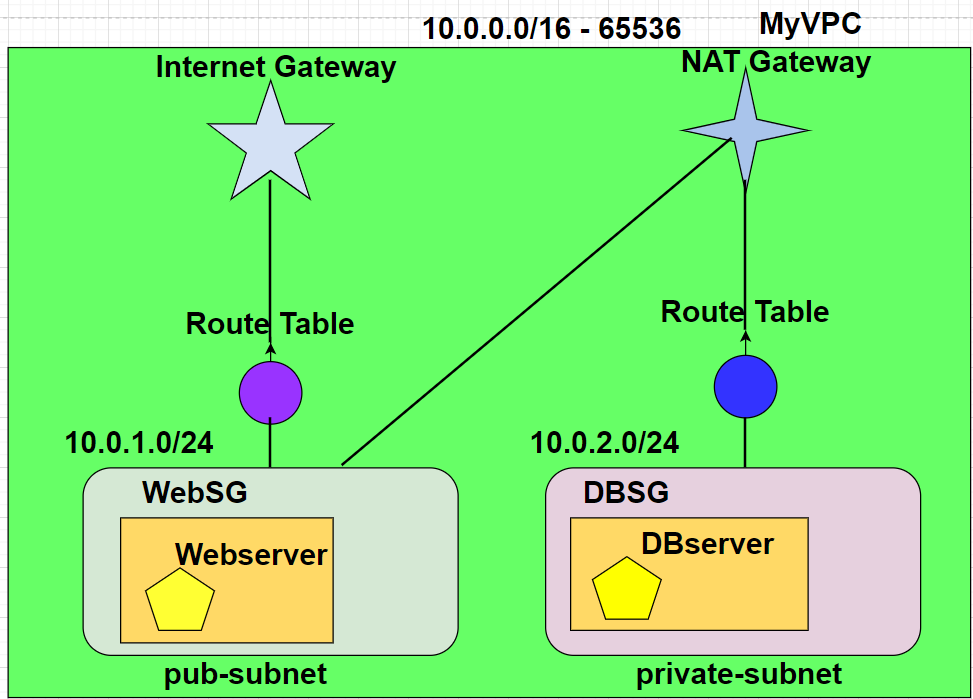


Now, another end to route table, we need to connect to Internet Gateway.

Select the route table (InternetRT) > Action >Edit routes > Add route > Target: Internet Gateway, select MyIGW. Destination: 0.0.0.0/0(Mandatory) > Save routes > Close



Now, our Subnet is public Subnet.



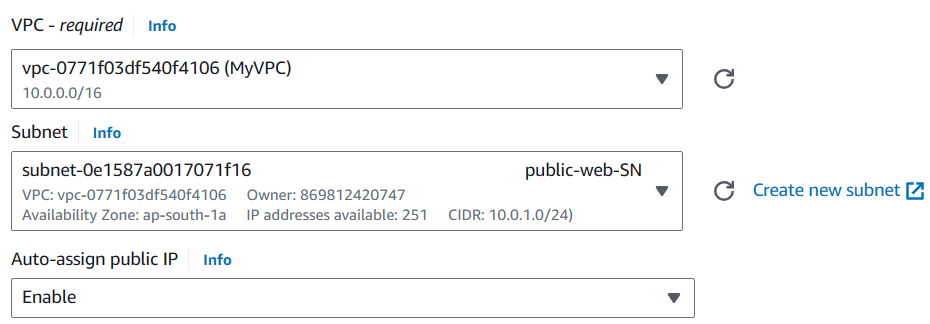
1. **Launch webserver** in public subnet.

Services > Ec2 > Launch instance

Name: WebServer

Instance: Amazon Linux, t2.micro, create keypair

Edit Network settings. Select MyVPC, public subnet Subnet(10.0.1.0/24), Security group name: webSG, add security group rule



**Type** **Source**

SSH Anywhere

HTTP Anywhere

**Additional Details -- User Data**

#!/bin/bash

sudo su

yum update -y

yum install httpd -y

cd /var/www/html

echo "Hello World" > index.html

srvice httpd start

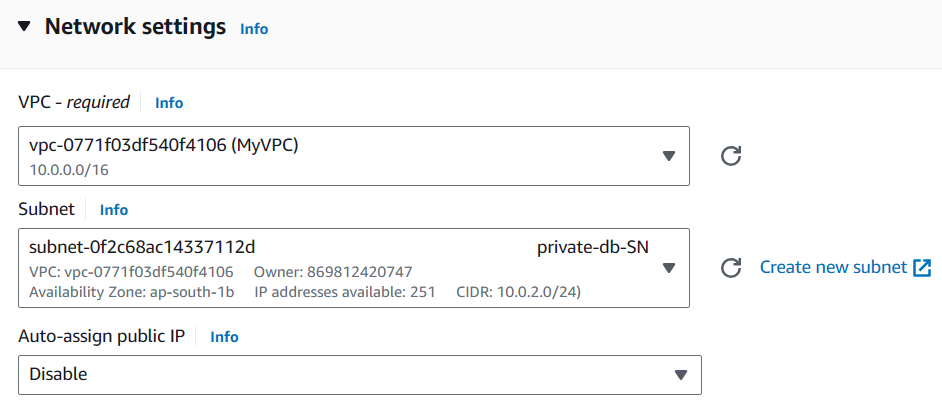
chkconfig httpd on

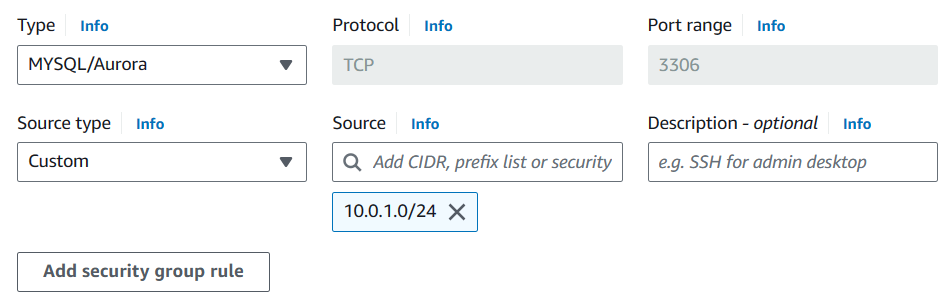
1. **Launch Database Server** in Private Subnet.

Name: DbServer

Instance: Amazon Linux, t2.micro, create keypair

Edit Network settings. Select MyVPC, private subnet Subnet(10.0.2.0/24), Security group name: DbSG, add security group rule



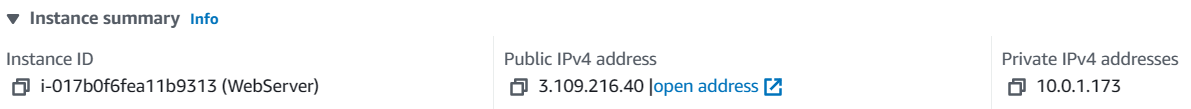


Change Type from SSH to MYSQL/Aurora

**Type**  **Source**

MYSQL/Aurora Custom 10.0.1.0/24

(MySQL Port is open to entire subnet) then Launch instance.

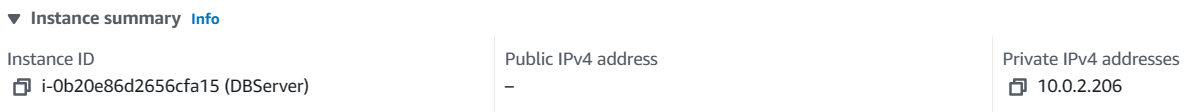


Observe the private IP address of the webserver.

The series will be 10.0.1.x

Now, check the Dbserver.

It should not have public IP and Private IP series should be 10.0.2.x



Now, let’s check, are we able to access webserver?

Copy the public IP in the browser

**Testing Scenarios:**

* Access web server using public IP. Yes, it is working.
* Detach Route table(InternetRT) from public-web-SN: Web server becomes inaccessible (refresh browser).
* Re-attach InternetRT: Web server becomes accessible again.
* Access web server from PuTTY using SSH (possible).
* Detach InternetRT again: SSH connection lost.
* Re-attach InternetRT: SSH connection restored.
* Access database server from EC2 console (possible).
* Access database server from PuTTY (not possible due to private subnet and no internet).

**Deletion Process**

* Terminate EC2 instances.
* Delete security groups.
* Delete MyVPC (automatically deletes subnets, routes, and gateways).