CLOUD CONCEPTS

(Virtual Private Cloud- VPC)

- VPC is virtual private cloud. Amazon provides a default VPC is every region. **Never delete default VPC** because VPC decides which type of traffic is allowed and blocked.

- It provisions you isolate the section of AWS logically where you can launch AWS resources in a virtual network that you define.
  - ✓ Complete control over your virtual networking environment
  - ✓ Selection of your IP address range
  - ✓ Creation of subnets
  - ✓ Configuration of route table and networking gateways

Additionally, you can create a hardware Virtual Network (VPN) connection between your corporate datacentre and your VPC and leverage the AWS cloud as an extension of your corporate datacentre.

Benefits:

- **Easy to use**

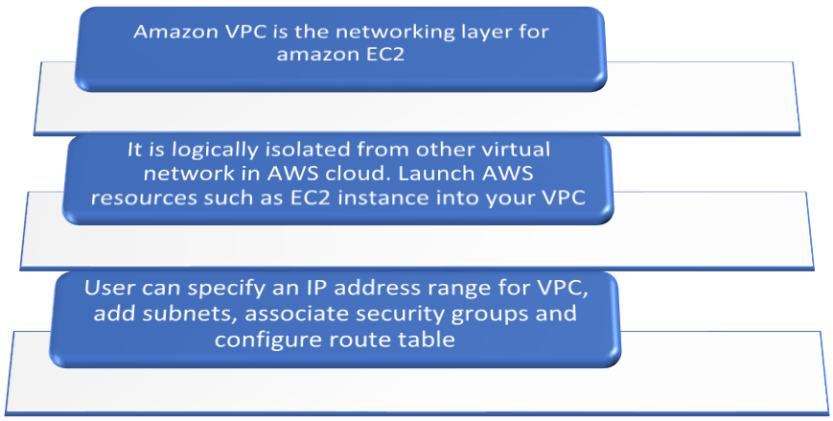
Ease of creating a VPC in very simple steps by selection network set-ups as per requirement. Define subnets, IP range, route tables, and security groups will be automatically created.

- **Pricing for Amazon VPC**

There's no additional storage care for using Amazon VPC. Pay the standard rates for the instance and other Amazon EC2 features that you use.

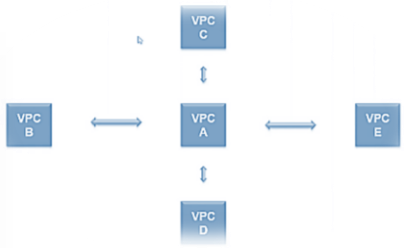
Features:

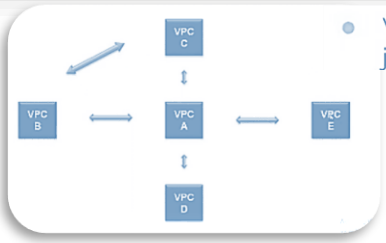
- Create an Amazon VPC on AWS scalable infrastructure and specify its private IP address range from any range you choose.
- Expand VPC by adding secondary IP ranges.
- Divide VPC private IP address range into one or more public or private subnets to facilitate running applications and services in VPC.
- Control inbound and outbound access to and from individual subnets using network access control lists.
- Store data in Amazon S3 and set permissions such that the data can only be accessed from within Amazon VPC.
- Attach one or more Amazon Elastic IP address to any instance in VPC so it can be reached directly from the internet.
- Connect VPC with other VPCs and access resources in the other VPCs via private IP address using VPC peering.
- Enable EC2 instance in the EC2-classic platform to communicate wit instance in a VPC using private IP address.
- Privately connect to other AWS services without using an internet gateway, NAT or firewall proxy through a VPC endpoint.
- Bridge VPC and onsite IT Infrastructure with an encrypted VPN connection.
- Privately connect to customized services or SaaS solution powered by AWS private link.
- User can use VPC flow logs to log information about network traffic going in and out of network interfaces in VPC.



- Say you have your database in your VPC, and you want to access it through the internet. For this an Internet Gateway is required (public internet access). NAT is used for private communication.
- Private address is mandatory.
- Communication of one VPC with other VPC though private IP address is **VPC Peering. If one VPC is in one account and other VPC is in another account even then the communication is possible.**

- Allow you to connect one VPC with another via a direct network route using private IP address.
- instance behave as if they were on the same private network
- You can peer VPC's with other AWS accounts as well as with other VPCs in the same account.
- Peering is in a star configuration: i.e.; 1 central VPC peer with 4 others. Not Transitive Peering!!!





- VPC Transitive Peering (Combine image no need for graphics just add arrow between VPC B and VPC C in second slide)

Default VPC vs Custom VPC

- Default VPC is user friendly, allowing you to immediately deploy instance.
- All subnet in default VPC have a route out the internet.
- Each EC2 instance has both a public and private IP address.

Key Concept | VPC and Subnet Sizing:

- VPC and Subnet Sizing for IPv4

While creating a VPC, must specify an IPv4 CIDR block for the VPC. The allowed block size is between a/16 netmask (65,536 IP address) and /28 netmask (16 IP address).

- Adding IPv4 CIDR Block to a VPC

Associate secondary IPv4 CIDR block with VPC. When CIDR block associate with VPC , a route is automatically added to VPC route tables to enable routing within the VPC.

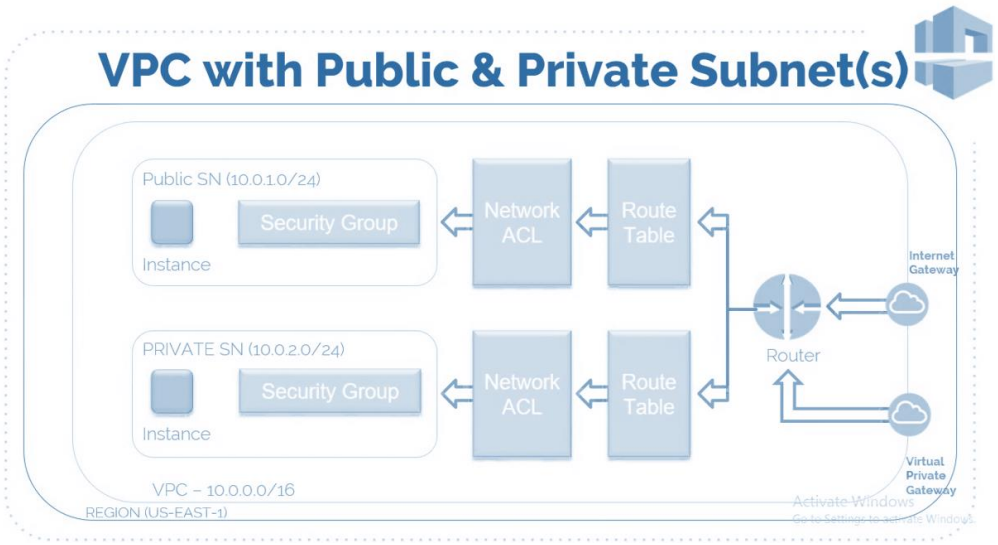
IMPORTANT POINTS:

- Preferred by the enterprise.
- Scale automatically up to 10Gbps.
- No need to patch
- Not associated with security groups
- Automatically assigned a public in IP address
- Remember to update your route tables.
- No need to disable source/destination checks
- More secure than a NAT instance

VPC Restrictions:

- 5 Elastic IP Address
- 5 Internet Gateway
- 5 VPCs per Region (Can be increased on request)
- 50 VPN connection per region
- 50 Customer Gateway per region
- 200 route tables per region
- 100 Security Group per VPC
- 50 Rules per Security group

- Think of a VPC as a logical datacenter in AWS.
- Consists of IGWs (or Virtual Private Gateway), Route tables, network access control lists, subnets, and security groups
- 1 subnet = 1 Availability Zone
- Security groups are stateful; Network access control lists are stateless.
- No transitive peering



- **Public subnet** = website for public access (traffic to/from internet in associated SG)
- **Private subnet** = data base for private access (traffic to/from internet is blocked in associated SG)
- Both subnets are in Availability Zone in a region. (One AZ = One Subnet)

STEPS:

10.0.0.0 – 10.255.255.255 (10/8 prefix)
172.16.0.0 – 172.31.255.255 (172.16/12 prefix)
192.168.0.0 – 192.168.255.255 (192.168/16 prefix)

1. Select a region selected and launch the VPC using specific IP range.
2. Define two gateways.
    a. Internet gate way for public access
    b. Virtual private gate way for private communication (*for DB*)
3. Setting up route tables. Any request received in sent to the route table. Route table will manage and decide the request to be forwarded to public or private subnet.

While creating the VPC AWS recommended specify CIDR block (of/16 or smaller) from the private IPv4 address ranges.

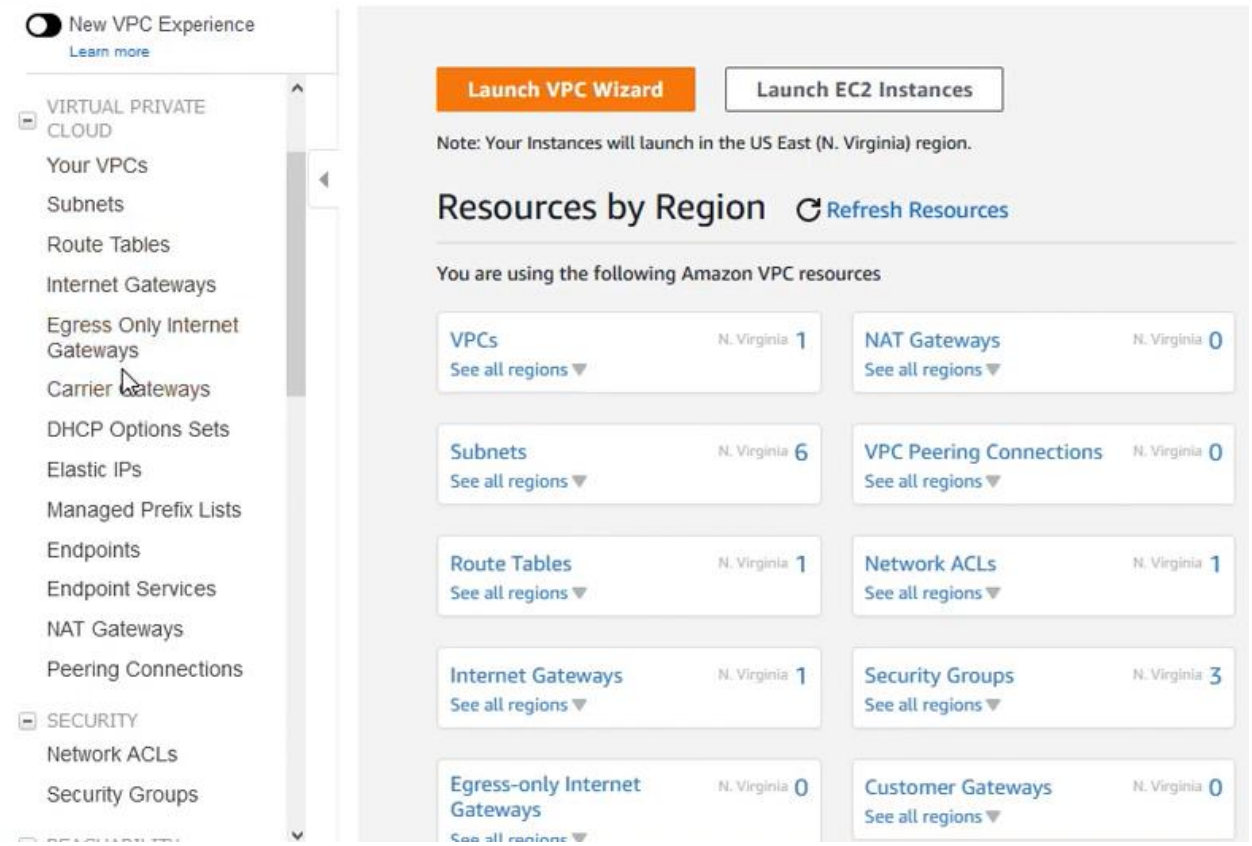
LAB (Custom VPC)

- Launch instance into a subnet of your choosing
- Assign custom IP address ranges in each subnet
- Configure route tables between subnets
- Create internet gateway and attach it to our VPC
- Much better security control over your AWS resources
- Instance Security groups
- Subnet network access control lists (ACLS)

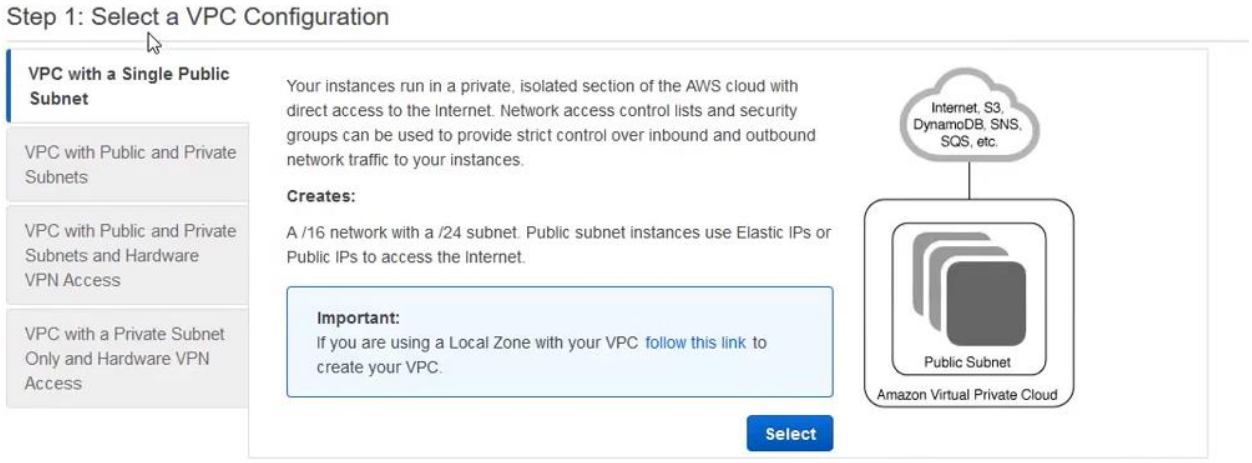
1. First select any specific region. (Say Ohio or North Virginia)

The number of AZ varies in different regions. No of AZs = No of subnets (more can be created)

2. Search VPC. (Explore the options)

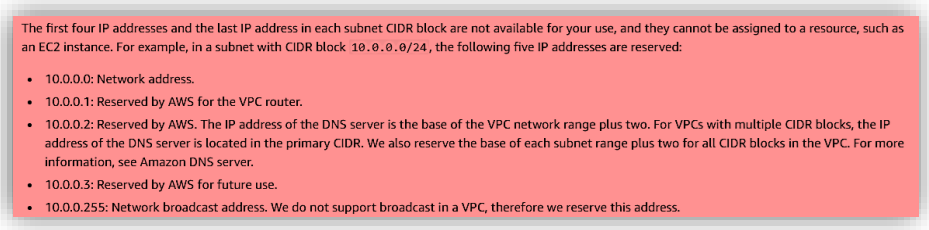


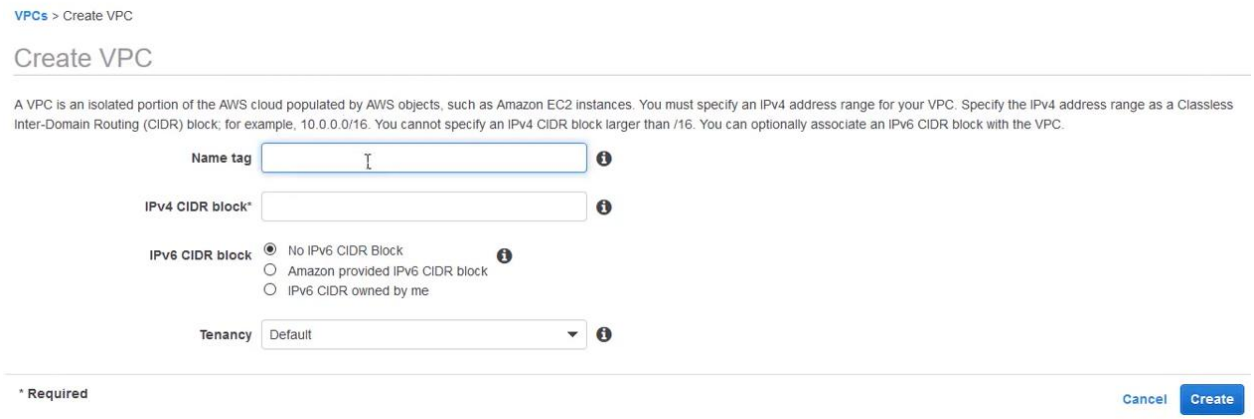
3. [SKIP THIS STEP TO STEP 4] Click on Launch VPC Configurations. You will see different options. You can select any of the options depending upon the requirement. (*Say optn1: anything in public subnet is accessible or optn2: say Database is in private subnet and your website is in public subnet*

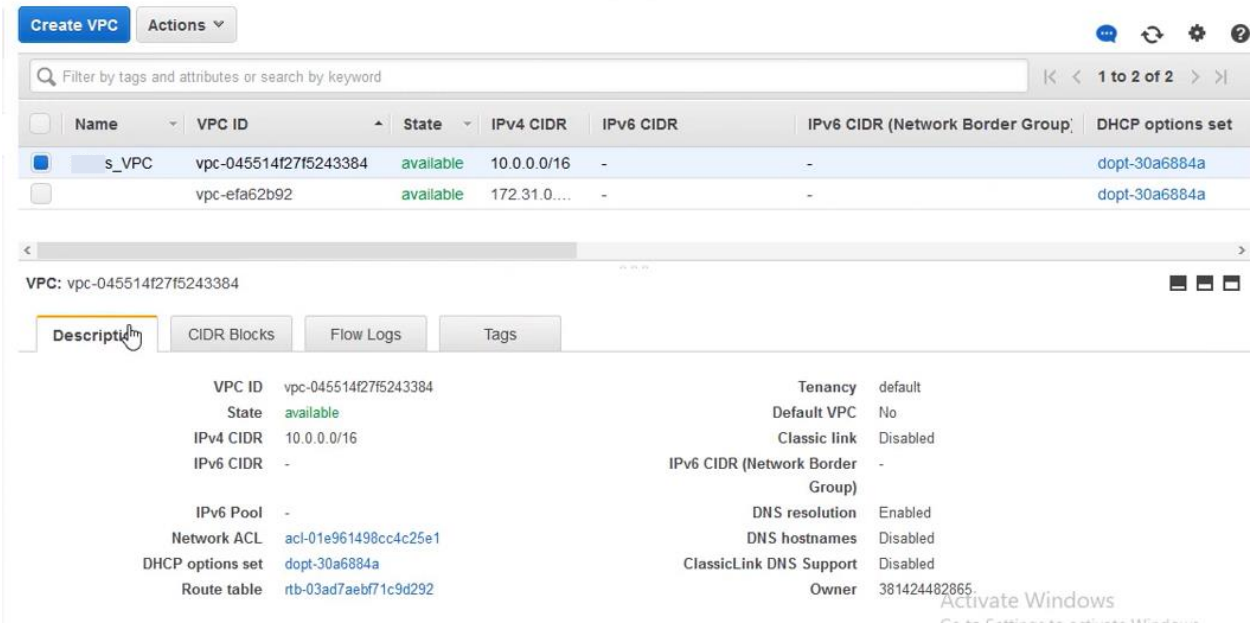


4. Click on “YOUR VPC” and check your running VPC --- Default VPC will be running which is connected to main route table. Never delete it.

5. Click on Create VPC and set the name of the VPC and define the IPv4 address from the given ranges. Leave the other options default and create the VPC.
    a. Default tenancy means shared hard drive is being used for VPC and dedicated tenancy means VPC will launch on underlying hypervisor.
    b. Used IP address range can be 10.0.0.0/16
        - When you create a VPC, you must specify an IPv4 CIDR block for the VPC. The allowed block size is between a /16 netmask (65,536 IP addresses) and /28 netmask (16 IP addresses)
        - The IP addresses for your subnets are represented using Classless Inter-Domain Routing (CIDR) notation.
        - The CIDR block of a subnet can be the same as the CIDR block for the VPC (to create a single subnet in the VPC), or a subset of the CIDR block for the VPC (to create multiple subnets in the VPC). The allowed block size is between a /28 netmask and /16 netmask. If you create more than one subnet in a VPC, the CIDR blocks of the subnets cannot overlap. For example, if you create a VPC with CIDR block 10.0.0.0/24, it supports 256 IP addresses. You can break this CIDR block into two subnets, each supporting 128 IP addresses. One subnet uses CIDR block 10.0.0.0/25 (for addresses 10.0.0.0 - 10.0.0.127) and the other uses CIDR block 10.0.0.128/25 (for addresses 10.0.0.128 - 10.0.0.255).
    c. In VPC 251 IP addresses available (5 are reserved). This is different from the subnetting we studied in which out of 256, available addresses are 254 [2 reserved].







6. After creating the VPC now create subnet click subnet. You will find the available subnets associated with the default VPC. The number of subnets varies if your region is changed. Now click on create subnet.

7. Set the name of the subnet custom. Select the VPC you created from the drop-down menu. Select the availability zone and make a new CIDR block for the subnet.
    a. Assuming 10.0.0.0/16 as VPC CIDR BLOCK [say 256 subnets]

- MAX 200 subnets per AZ are allowed in AWS VPC.
- MAX 5 VPCs are allowed per region.
- Subnets belong to one AZ and VPC can span more than one AZ.

    i. First subnet 10.0.1.0/24 [251 IPs]
    ii. Second subnet 10.0.2.0/24 [251 IPs]
    iii. Third subnet 10.0.3.0/24 [251 IPs]
8. Create three subnets in three different availability zones of a region. It must be done explicitly else Amazon will set up all the subnets in ONE AZ. (We are looking for high availability setup)

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 No preference

VPC CIDRs

CIDR	Status	Status Reason

IPv4 CIDR block*

* Required

Cancel Create

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 10.0.1.0 - a

VPC* vpc-045514f27f5243384

Availability Zone us-east-1a

VPC CIDRs

CIDR	Status	Status Reason
10.0.0.0/16	associated	

IPv4 CIDR block* 10.0.1.0/24

* Required

Cancel Create

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 10.0.2.0 - b

VPC* vpc-045514f27f5243384

Availability Zone us-east-1b

VPC CIDRs

CIDR	Status	Status Reason
10.0.0.0/16	associated	

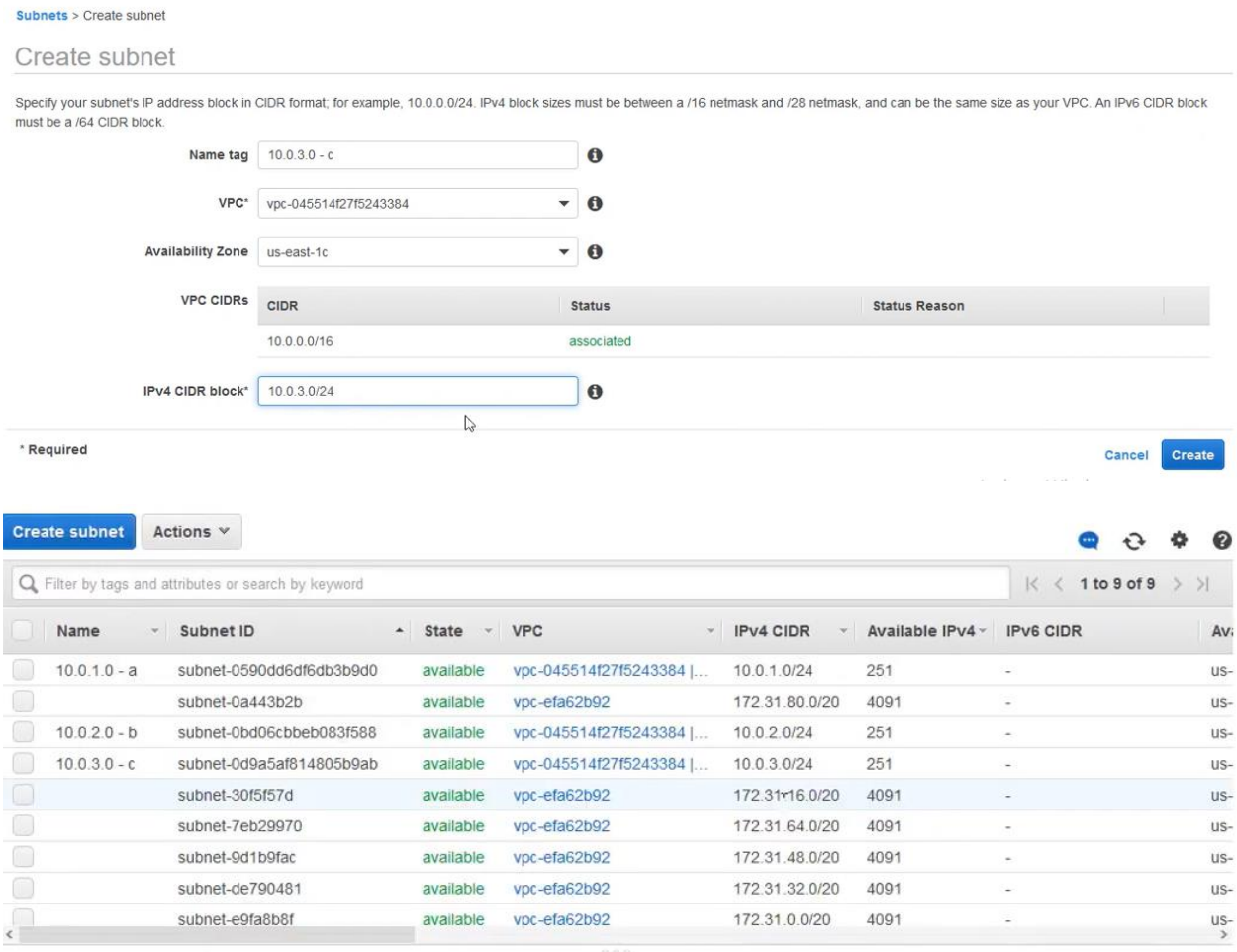
IPv4 CIDR block* 10.0.2.0/24

* Required

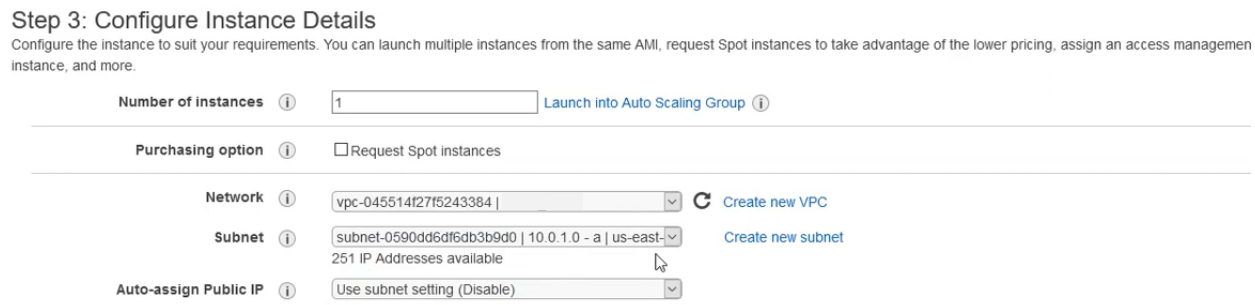
Cancel Create

E & OE

Handouts: Drakhshan Bokhat

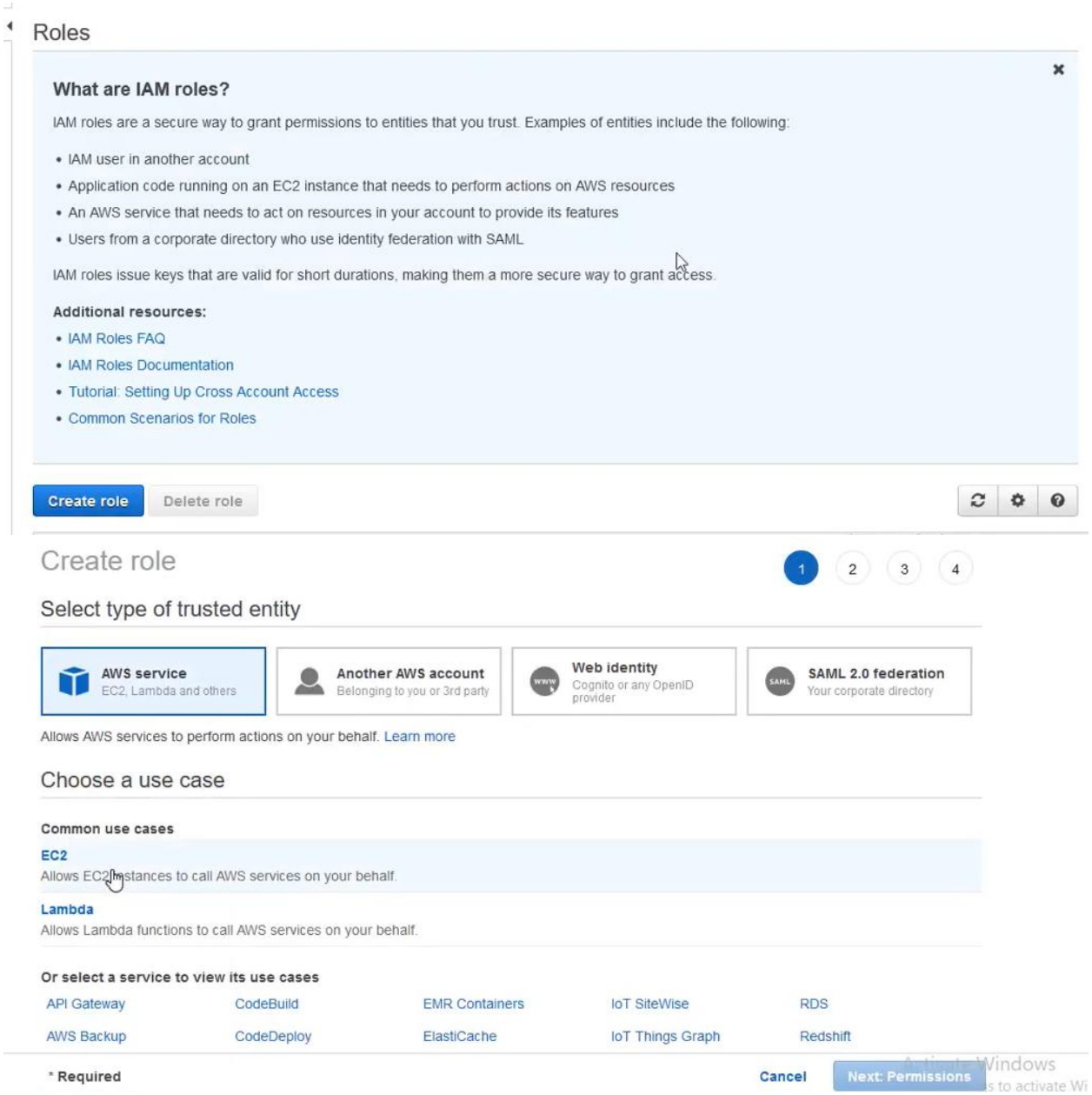


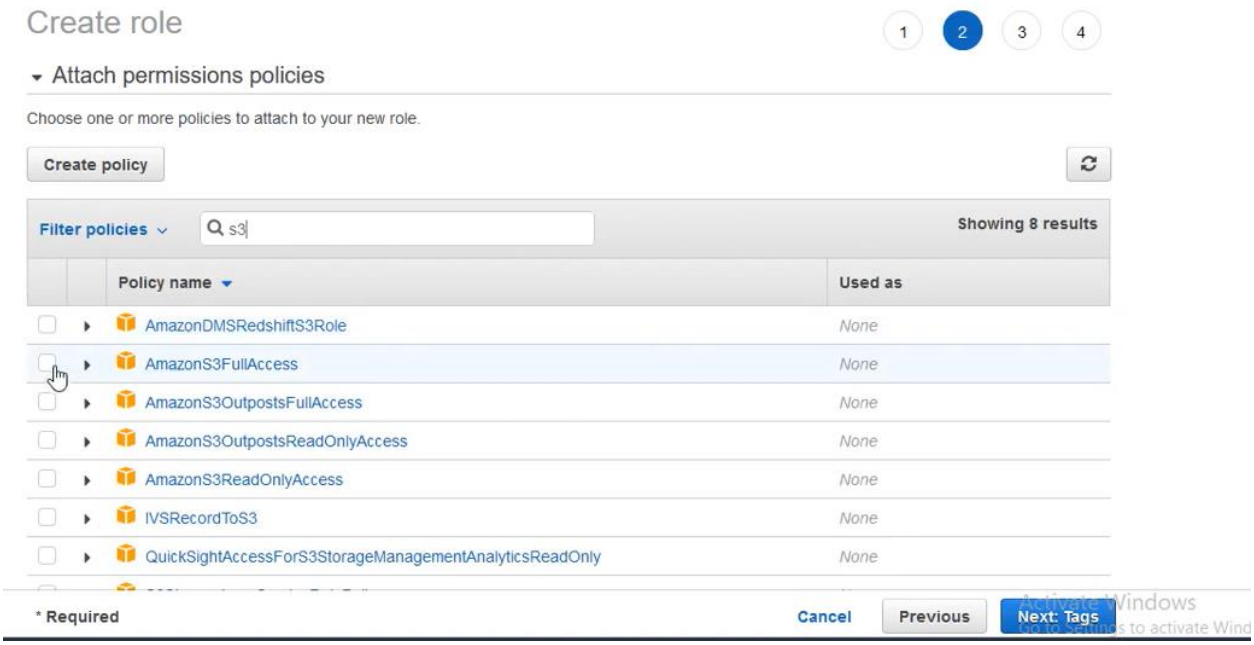
9. **All these subnets are private (we want our data to be private)**. If any server is launched in an ec2 instances these servers cannot be accessed through public internet. It can only be accessed through load balancer.
10. Now launch a server using EC2 service in your VPC (custom VPC) and specify the subnet from the subnets you created.

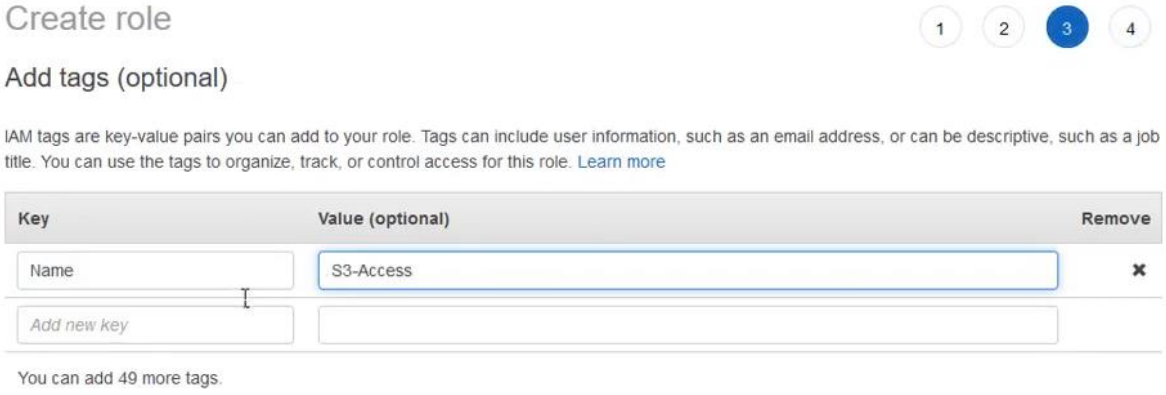


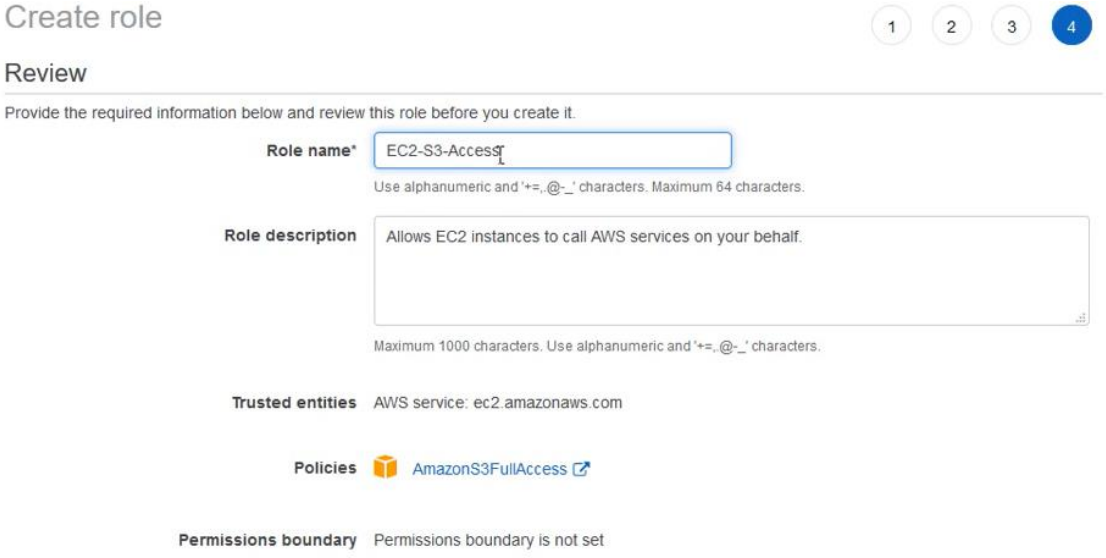
11. Attach IAM role S3 if you want your server (EC2) to communicate with S3. IAM role is defined whenever one service needs to communicate with another service.

12. To create the role. Click on create new role. Select EC2 as EC2 wants to communicate with S3 so attach the permission of S3. Set the names and tags and create the role.

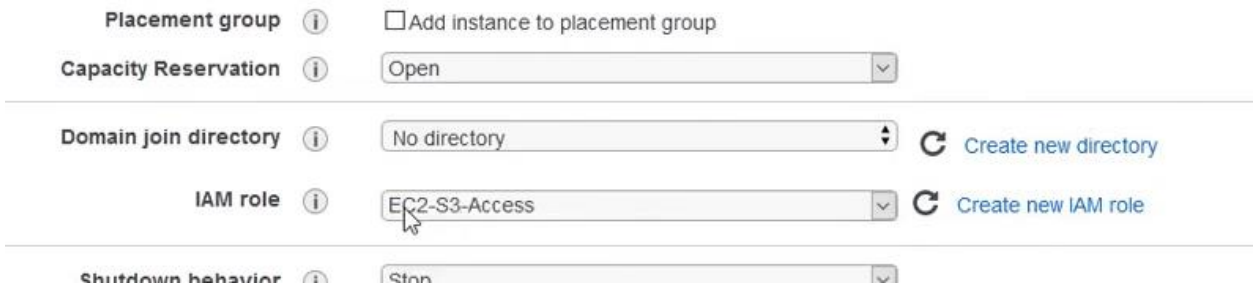


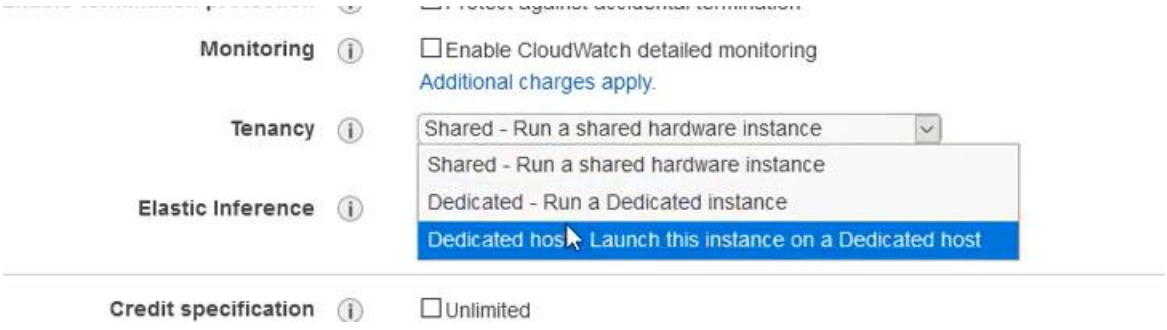

Create role
1 2 3 4
Attach permissions policies
Choose one or more policies to attach to your new role.
Create policy
Filter policies
s3
Showing 8 results
Policy name
Used as
AmazonDMSRedshiftS3Role
AmazonS3FullAccess
AmazonS3OutpostsFullAccess
AmazonS3OutpostsReadOnlyAccess
AmazonS3ReadOnlyAccess
IVSRecordToS3
QuickSightAccessForS3StorageManagementAnalyticsReadOnly
None
* Required
Cancel
Previous
Next: Tags

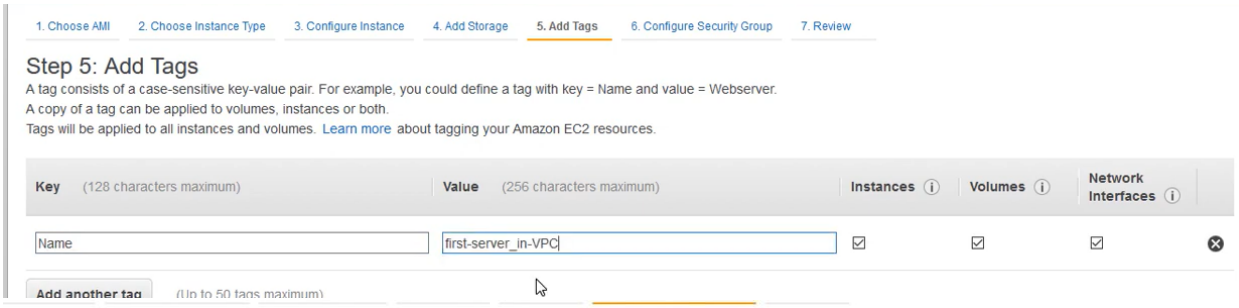

Create role
1 2 3 4
Add tags (optional)
IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. Learn more
Key
Value (optional)
Remove
Name
S3-Access
Add new key
You can add 49 more tags.

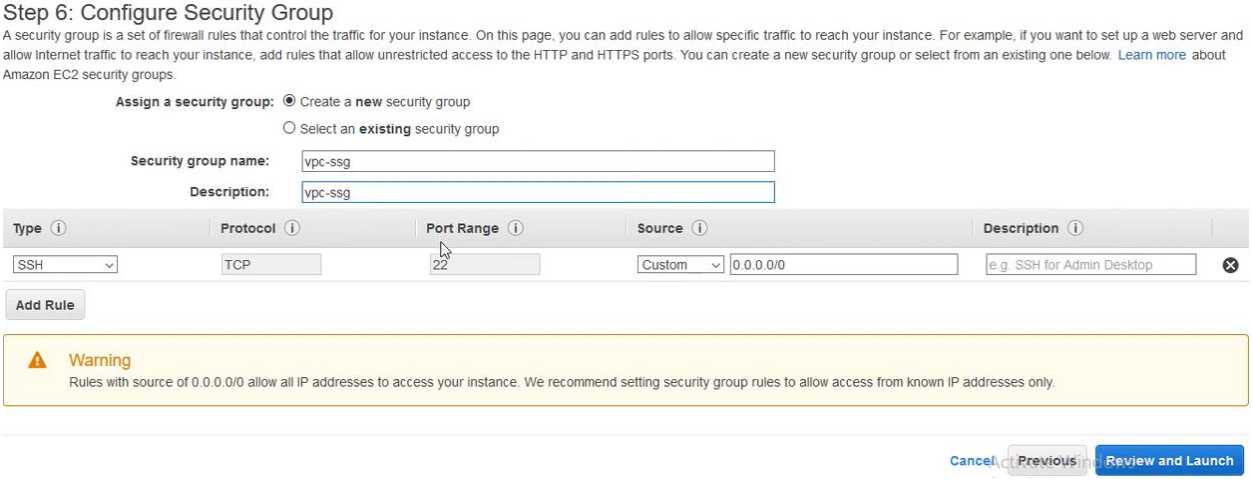

Create role
1 2 3 4
Review
Provide the required information below and review this role before you create it.
Role name*
EC2-S3-Access
Use alphanumeric and '+=,.@-_' characters. Maximum 64 characters.
Role description
Allows EC2 instances to call AWS services on your behalf.
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.
Trusted entities
AWS service: ec2.amazonaws.com
Policies
AmazonS3FullAccess
Permissions boundary
Permissions boundary is not set

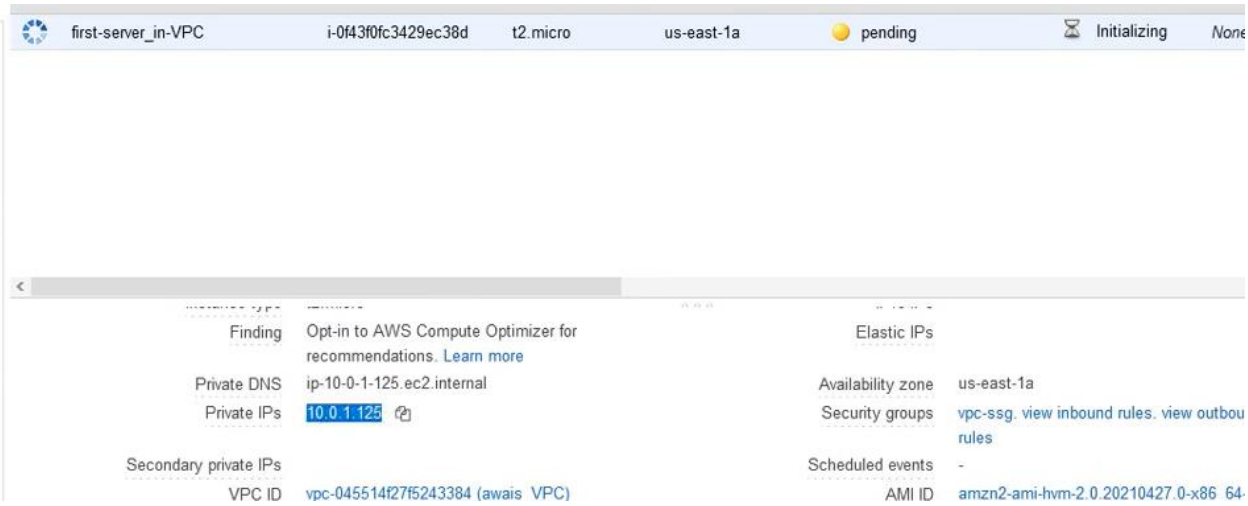
13. Now attach the role. Keep the other values as default (tenancy = shared) and attach the security group. Please note that this SG resides in your custom VPC not in the region. Review and Launch.
14. Launch three instances in different AZs and install the Apache server (different servers index file for the difference) on them.





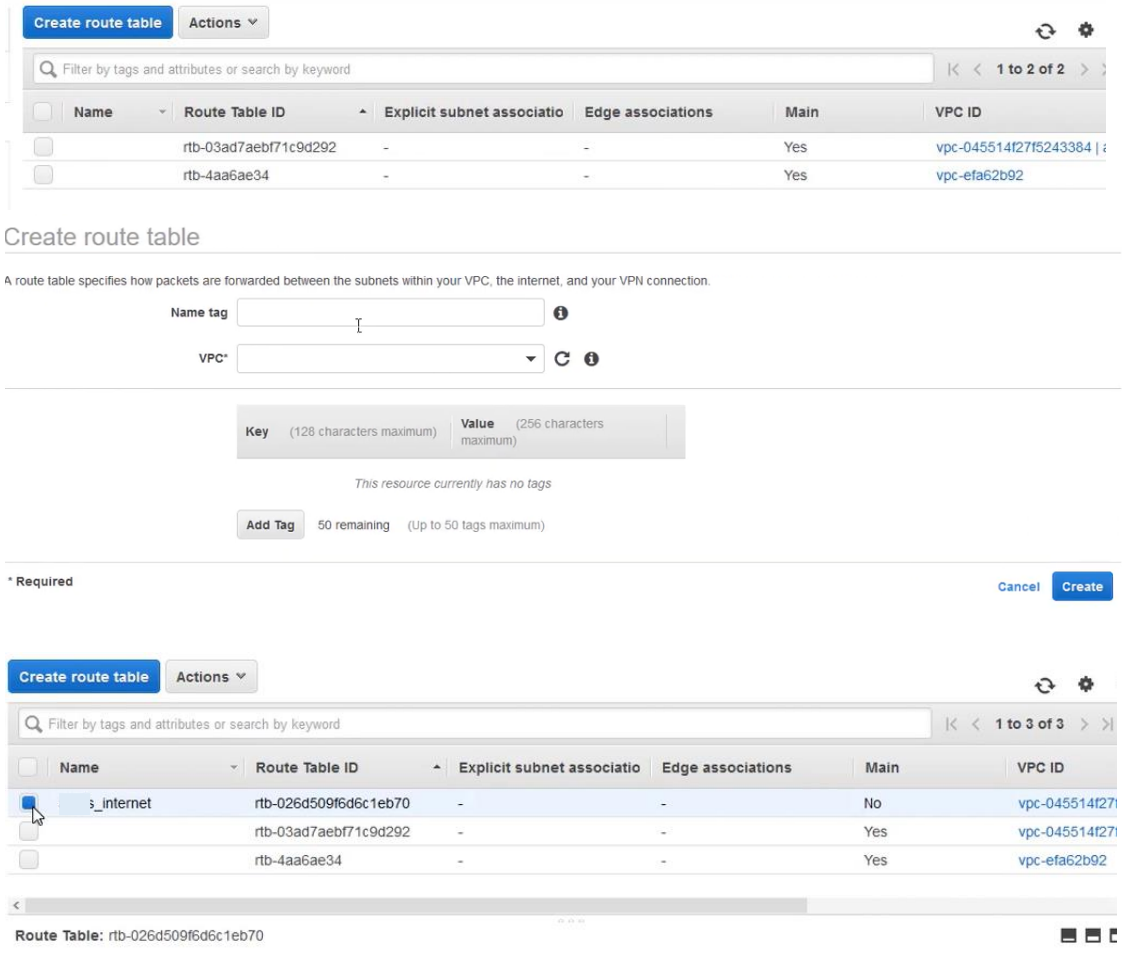




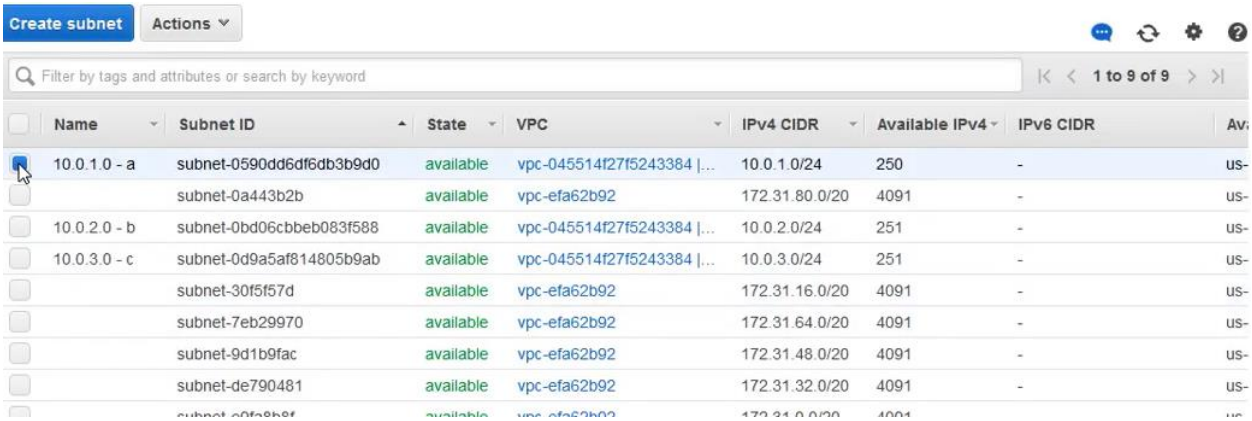


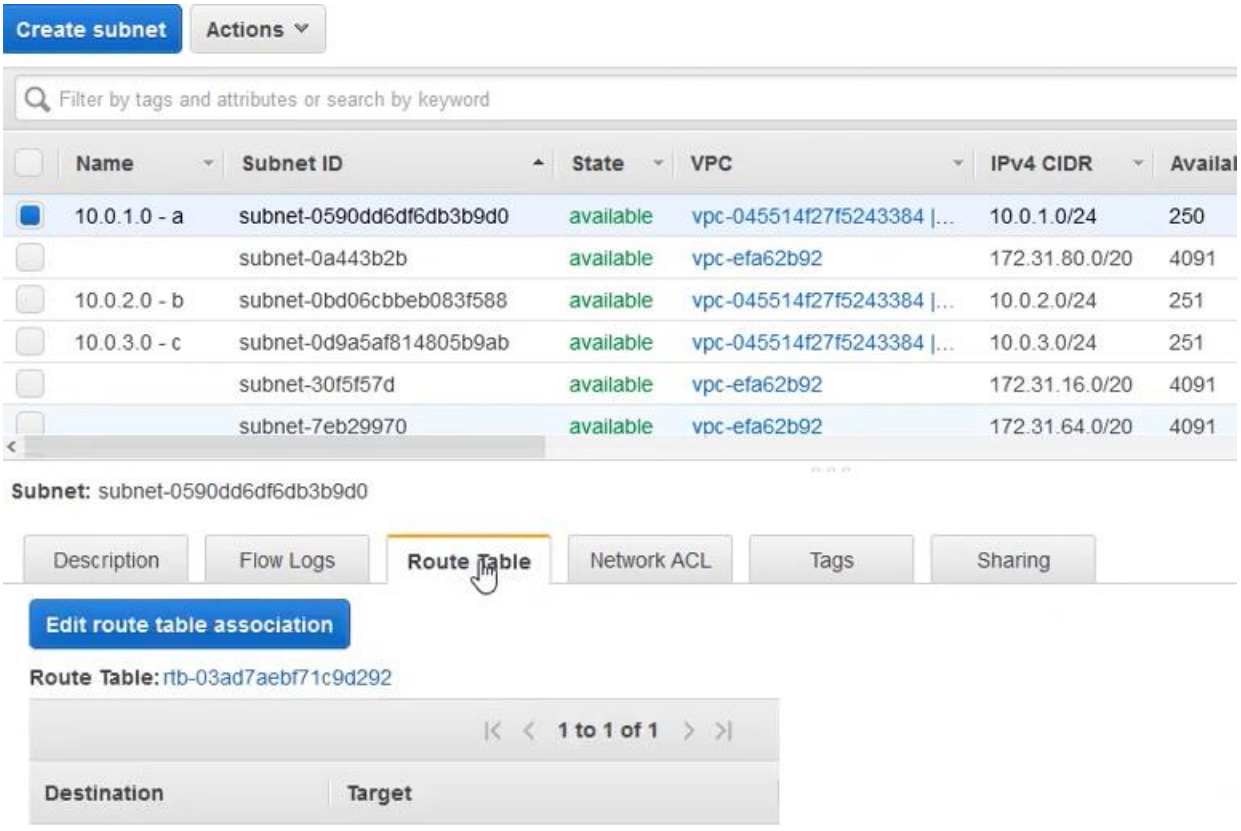
15. After creating subnets and launching the server in the subnet now create the route table. In route tables you will see two route tables. One is default and other is with your custom VPC. Both the route tables are private. Now click on create table click give name and select your custom VPC and tag if you want. Your route table is created.

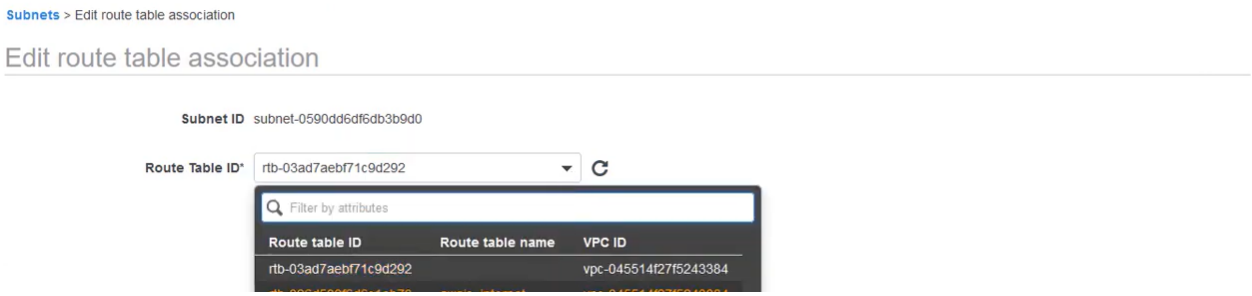
    • Each subnet can only be associated with one route table.

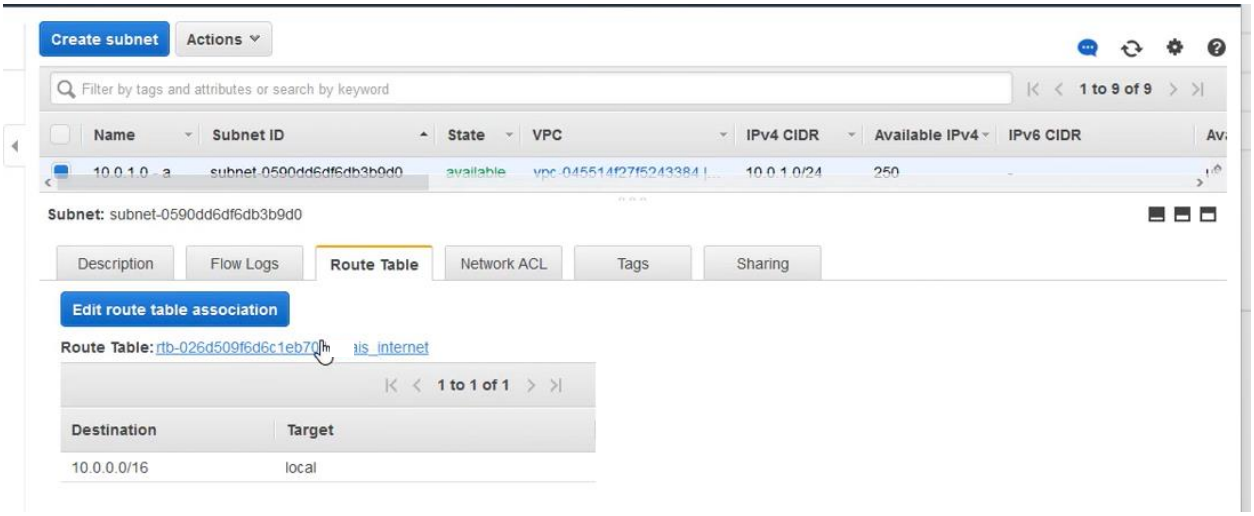


16. After creating the route table go to the subnet and select a subnet to associate the created route table with that subnet. Your subnet is linked with the default route table.

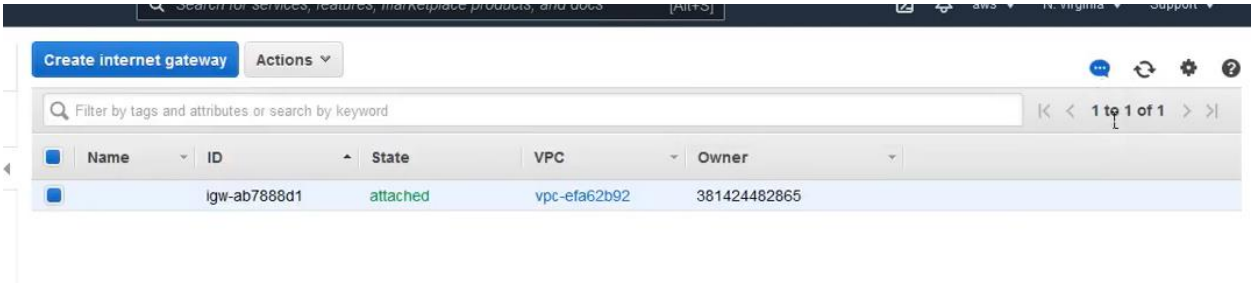




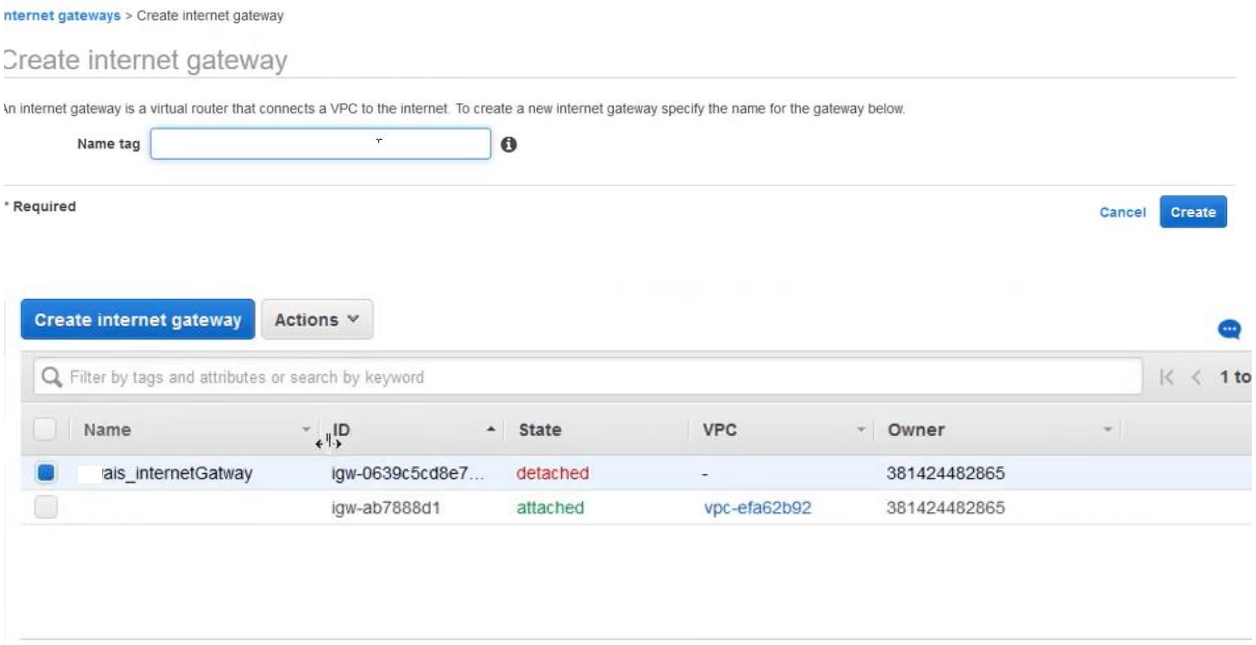


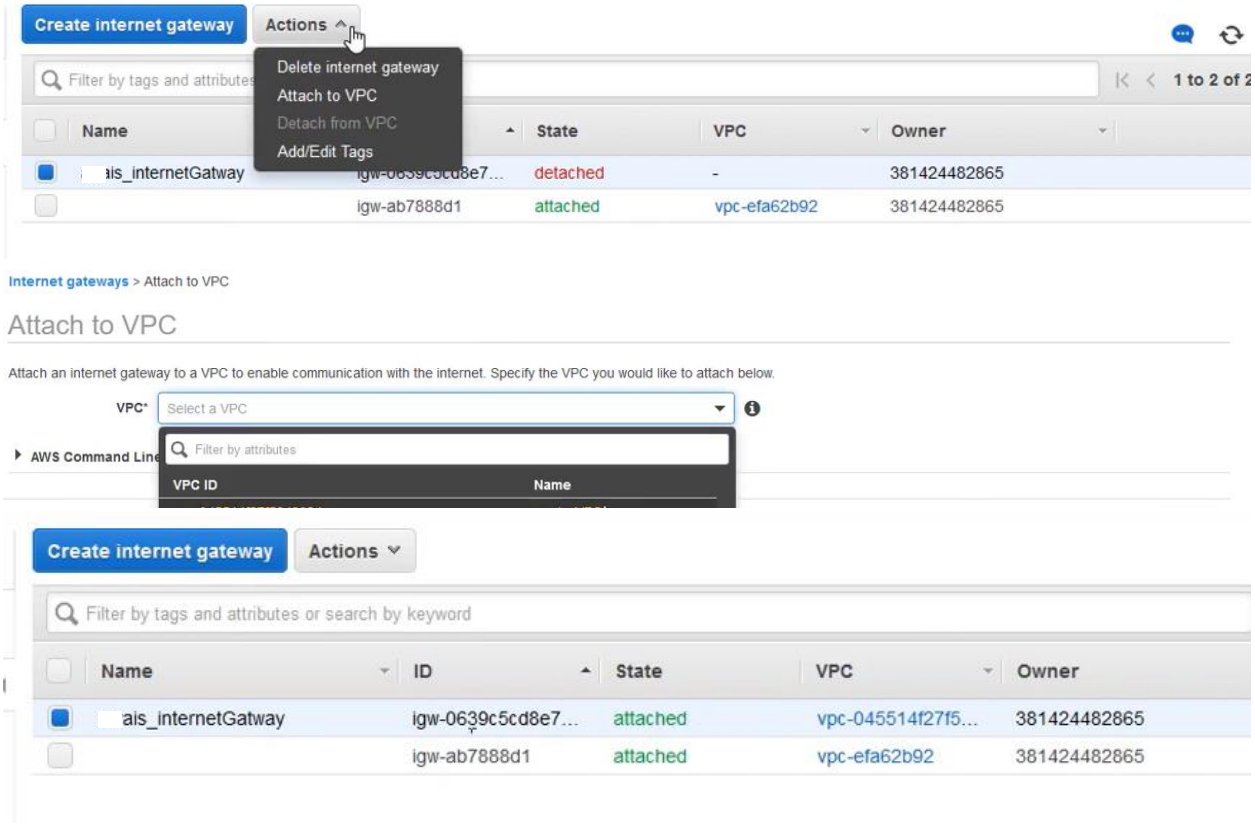


17. Now create internet gateways. The default gateway associated with default VPC is there. You must create your own gateway.

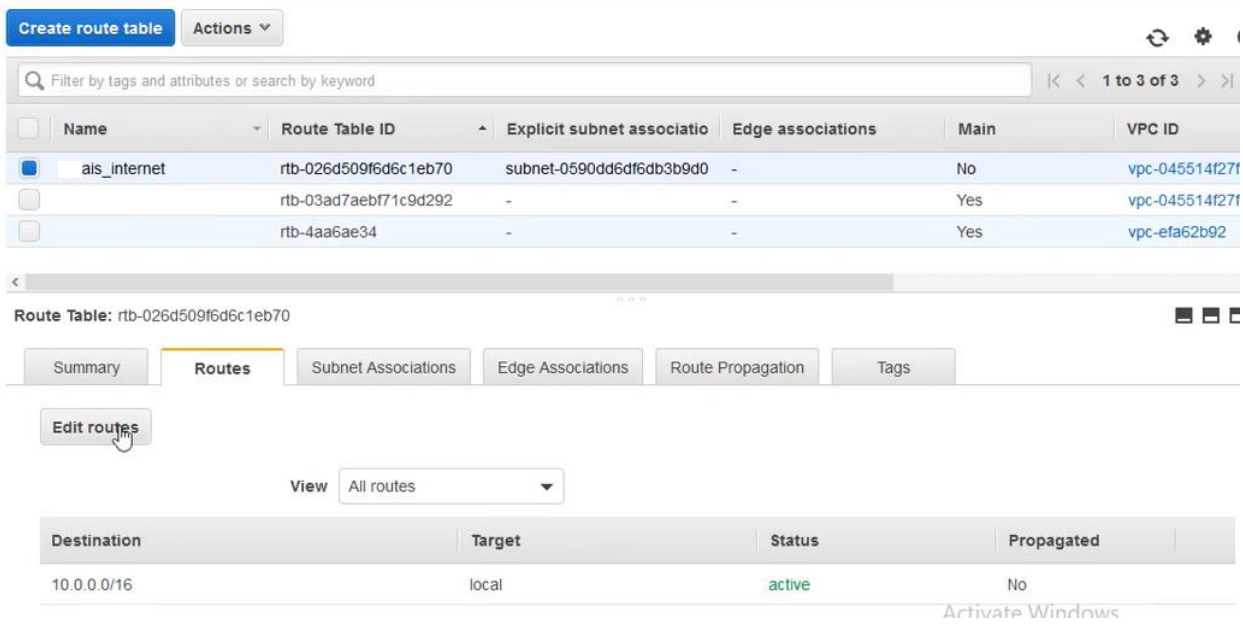


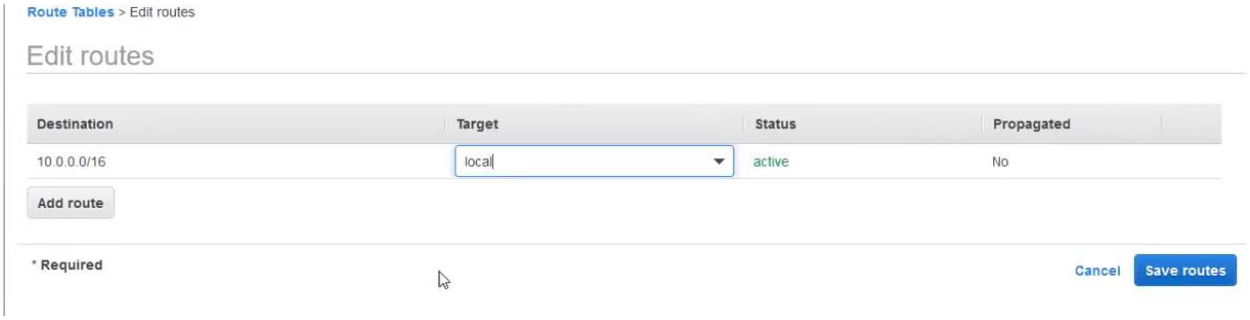
18. Create the gateway and attach it with your custom VPC. This gateway is used for internet access for the public subnets.

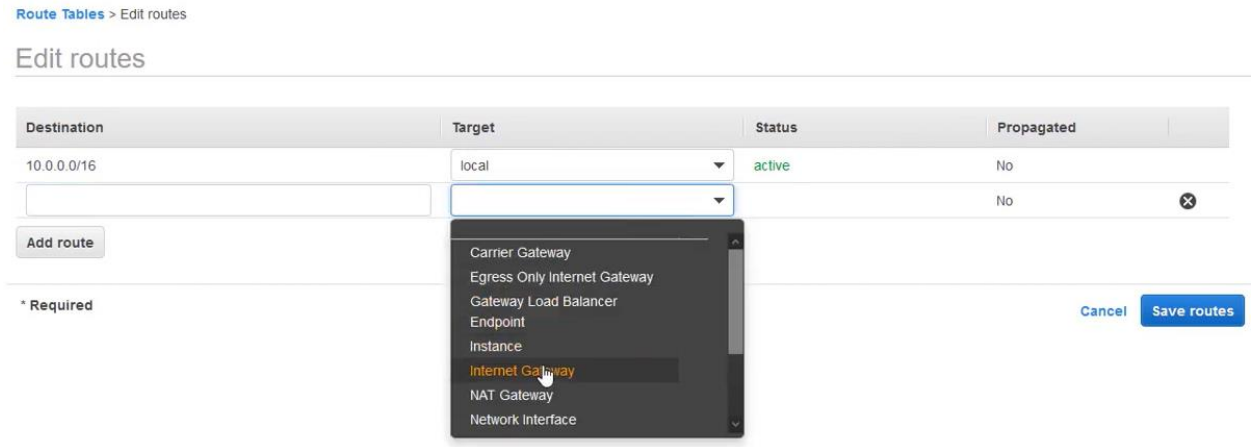


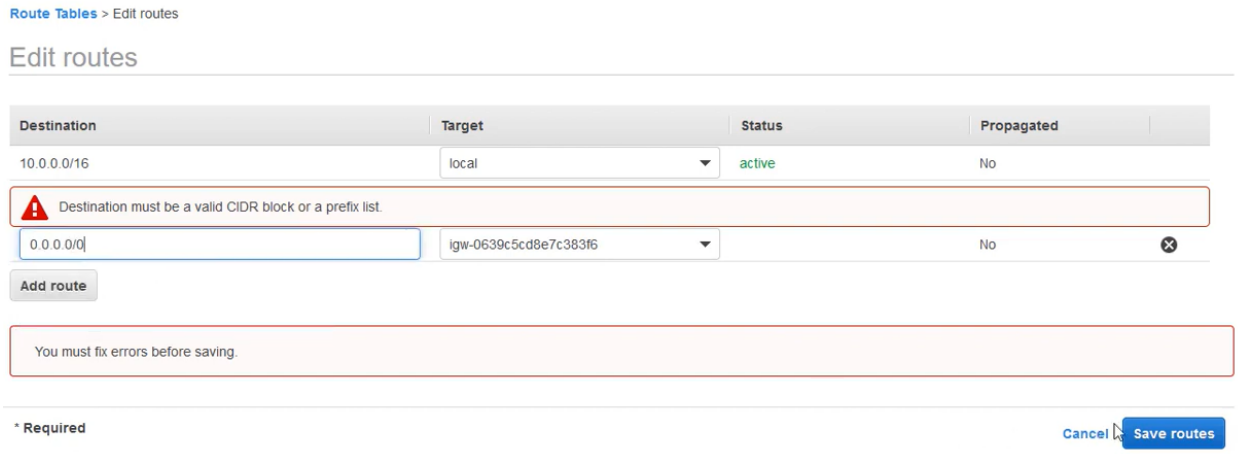


19. Now enable the gateway and set the route in your route table. Go to route table select your custom route table and add the route. Attach your IGW and allow all the traffic from the internet.

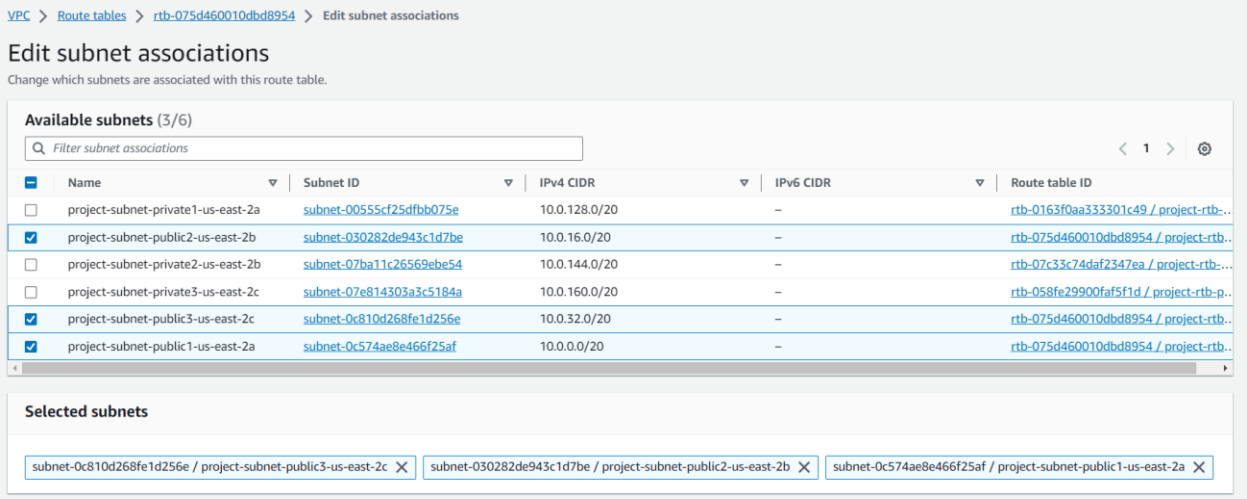




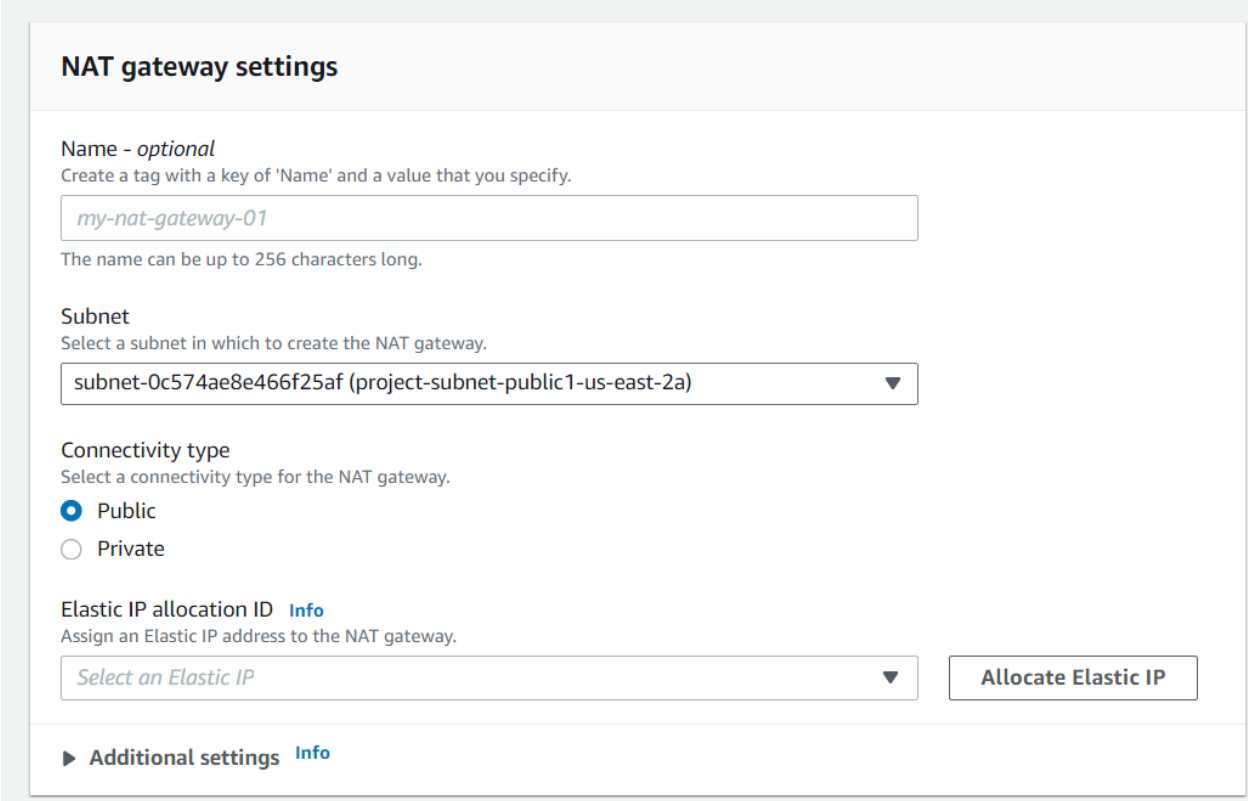




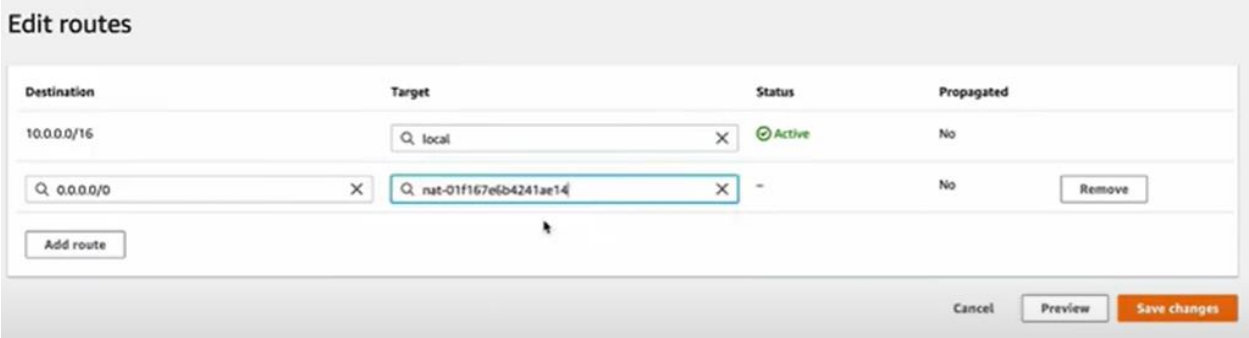
20. (for public subnets) Click on route table and then go to associate subnets. Edit and associate the public subnets with the route table.

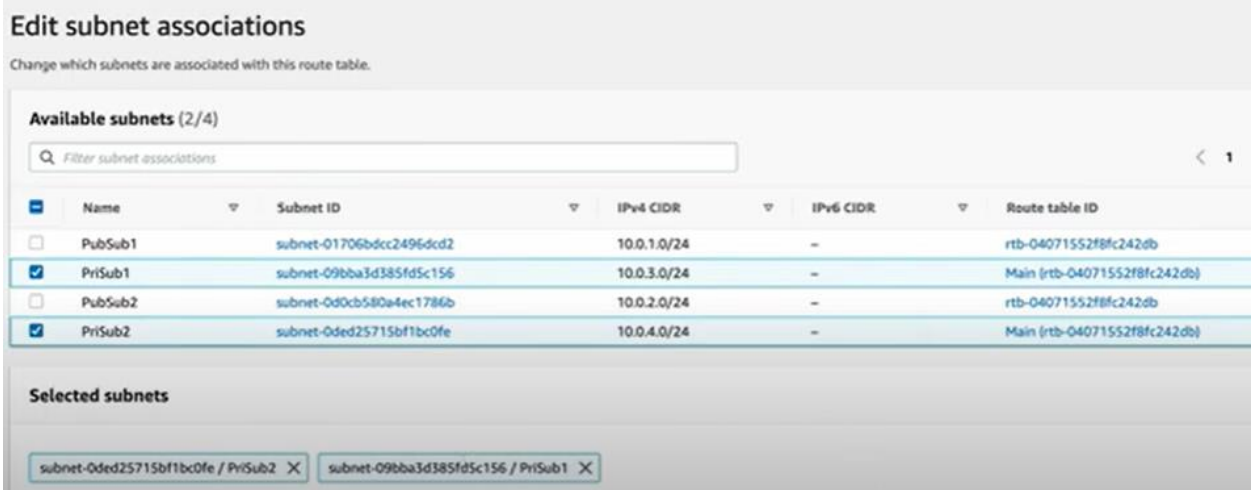


21. Now create the NAT for the internet access in virtual machines in the private subnet, in public subnet and set the connectivity as public. Set the name and assign the elastic IP.

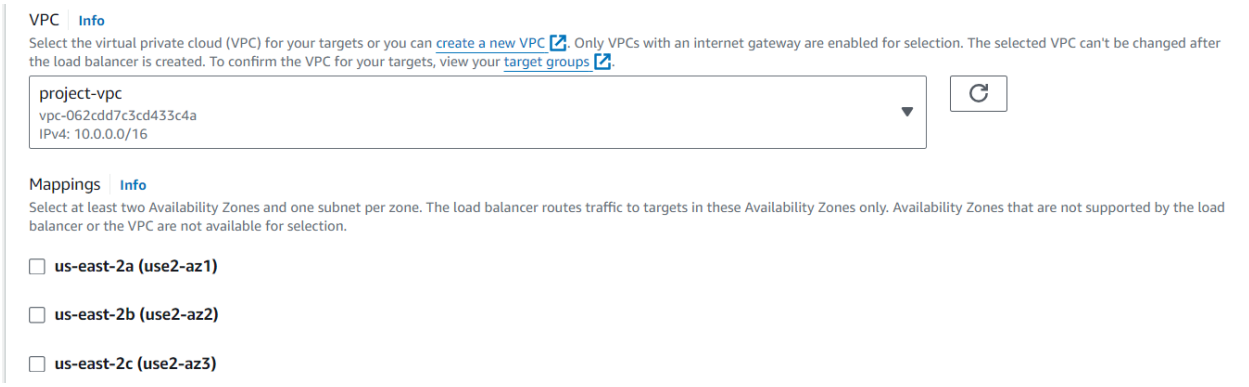


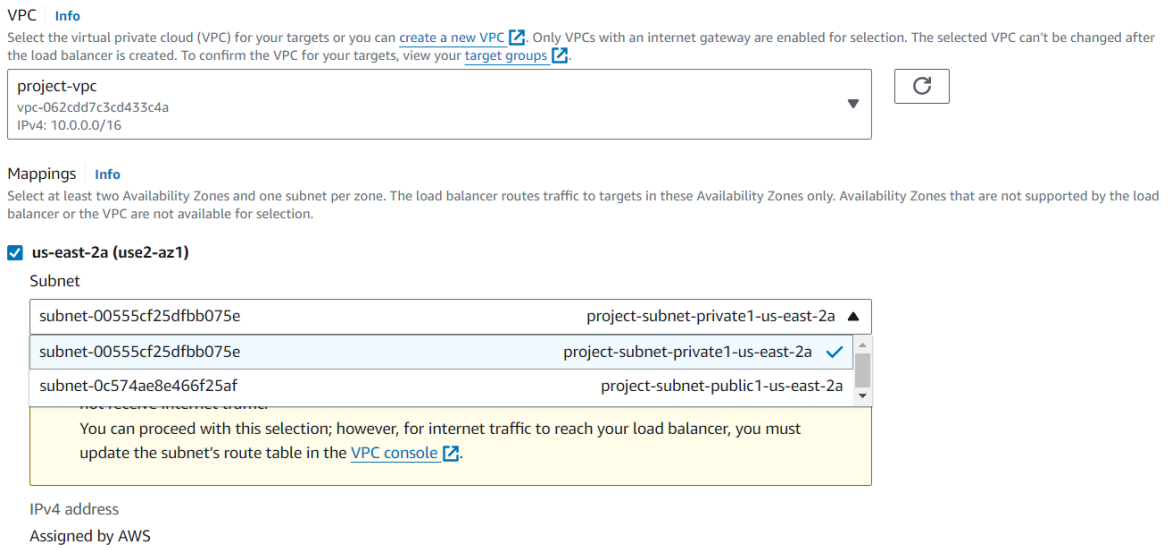
22. Now create a separate route table for NAT under the same VPC. Add the route and associate the private subnets to this route table.





23. Now create the ELB. Please note that ELB must be in your custom VPC. Add your instances work under your load balancer. The difference between the ELB in VPC and creating load balancer in EC2 is that in VPC we must add subnets. Since there is one subnet per AZ you must add the three AZ features.

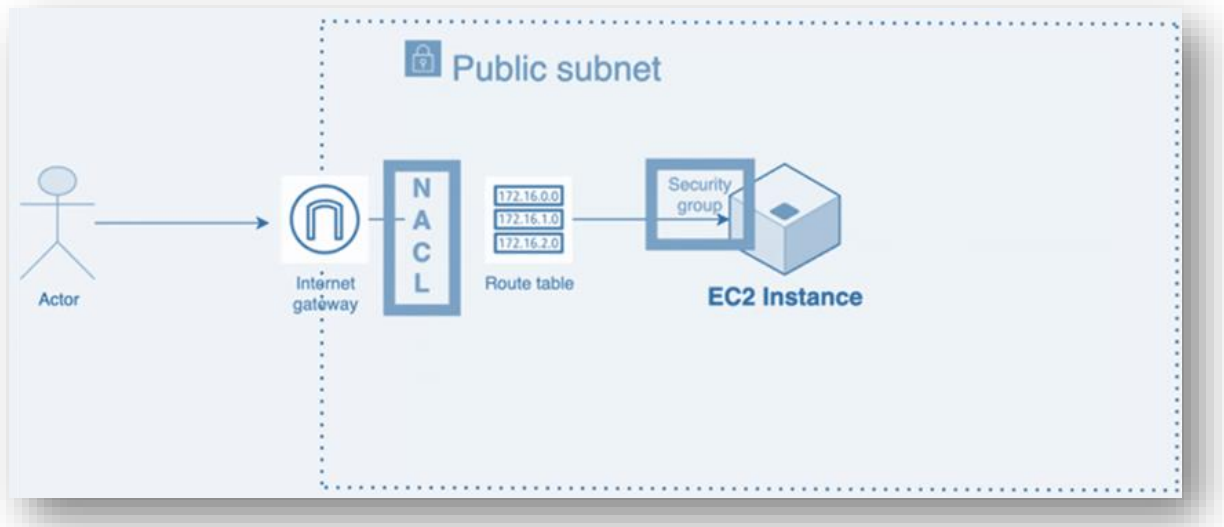




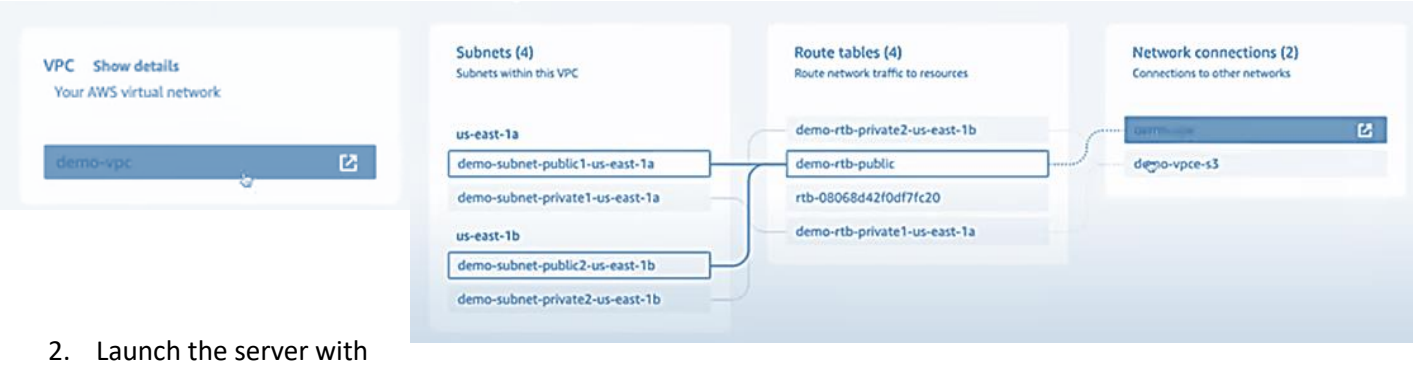
24. Instances (private) in VPC are not accessible through public IP so attach the elastic IPs with your instances. You must have elastic IPs created for that purpose. Now go to load balancer and access the load balancer through public DNS. Stop the instances to see the ELB in work in your VPC.
25. If you have your own DNS, you can use route 53 setup your DNS and redirect the traffic to load balancer.
    - **AWS Route 53** translates URL names, such as www.wordpress.com, into their corresponding numeric IP addresses.

LAB TASK: WORKING OF Network Access Control List

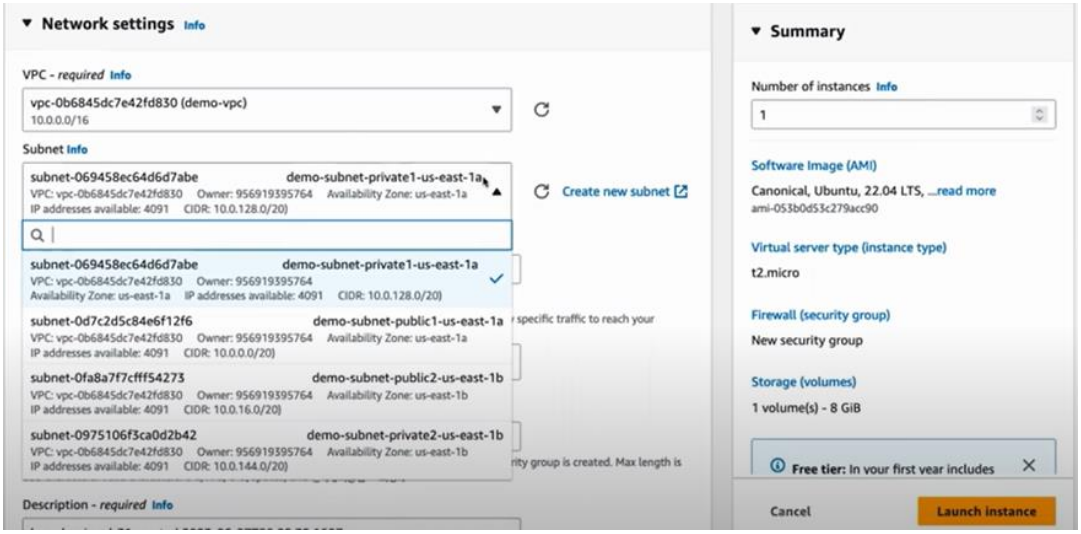
One of the tools in the AWS security toolkit for enabling defense-in-depth, is the Network Access Control List (NACL). A NACL is a security layer for your VPC, that acts as a firewall for controlling traffic in and out of one or more subnets



1. Create a VPC with more options.



2. Launch the server with
(assign) public IP in public subnet of your custom VPC (one subnet = one AZ will be enough). (Recommended or default selection by Amazon will be private subnet to keep your organizational data private).



3. Install python application or Apache server though SSH.
    a. Update the packages: Sudo apt update.
    b. Check if python is installed: Python3
    c. Install simple http server on python and access through port 8080: python3 -m http.server 8080
4. Try to access the application installed /running on your server though IP address and port number (on Browser of your system). [Format: htttp://IP:port no]
5. Not accessed? only SSH is allowed in SG attached to the instance? [**Yes**/No]
6. Now check NACL and check the inbound rules? All the traffic is allowed? [**Yes**/No]
7. NACL works at subnet level so its first level of security. Internet gateway will let the traffic enter the public subnet.
8. Change the SG rules and allow the traffic.
9. Refresh the page output will be shown.
10. Now restrict the traffic at NACL keeping SG rules allowing that type of traffic.
11. Try different rule numbers for different rules in NACL and check its pattern of execution.
12. Block specific IP addresses.