BLACKBUCKS INTERNSHIP REPORT

Building an Architecture for Application and Database Deployment

SUBMITTED BY

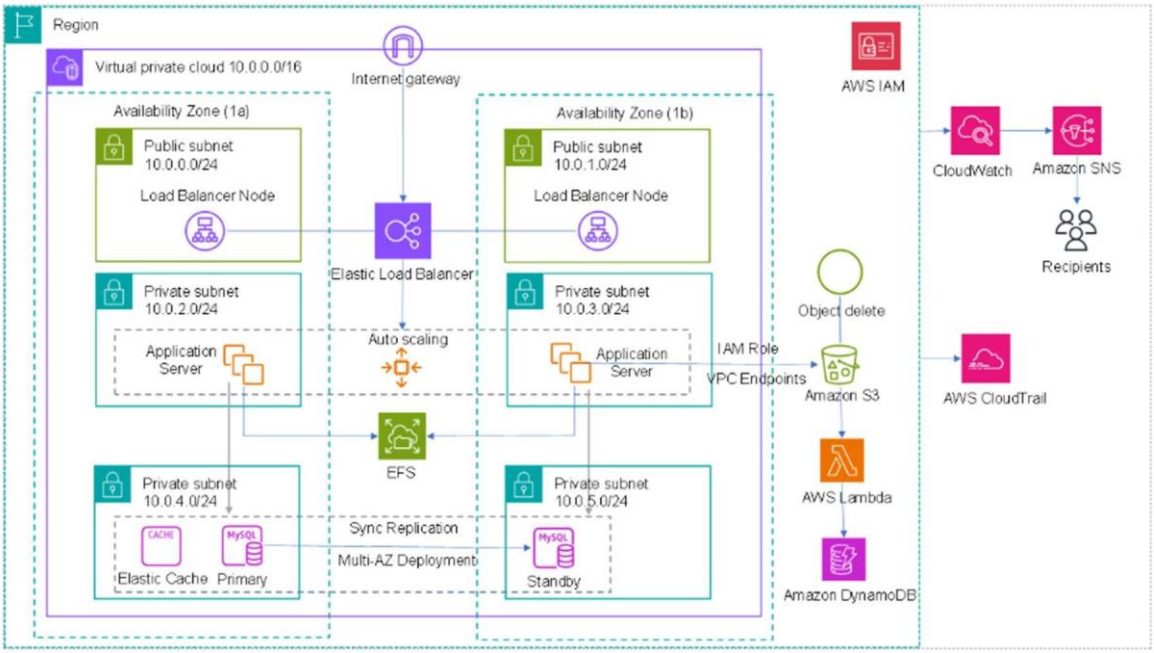
Siri Chandana Varre(20B91A12I3)

UNDER THE GUIDANCE OF MS. Anupama Voruganti

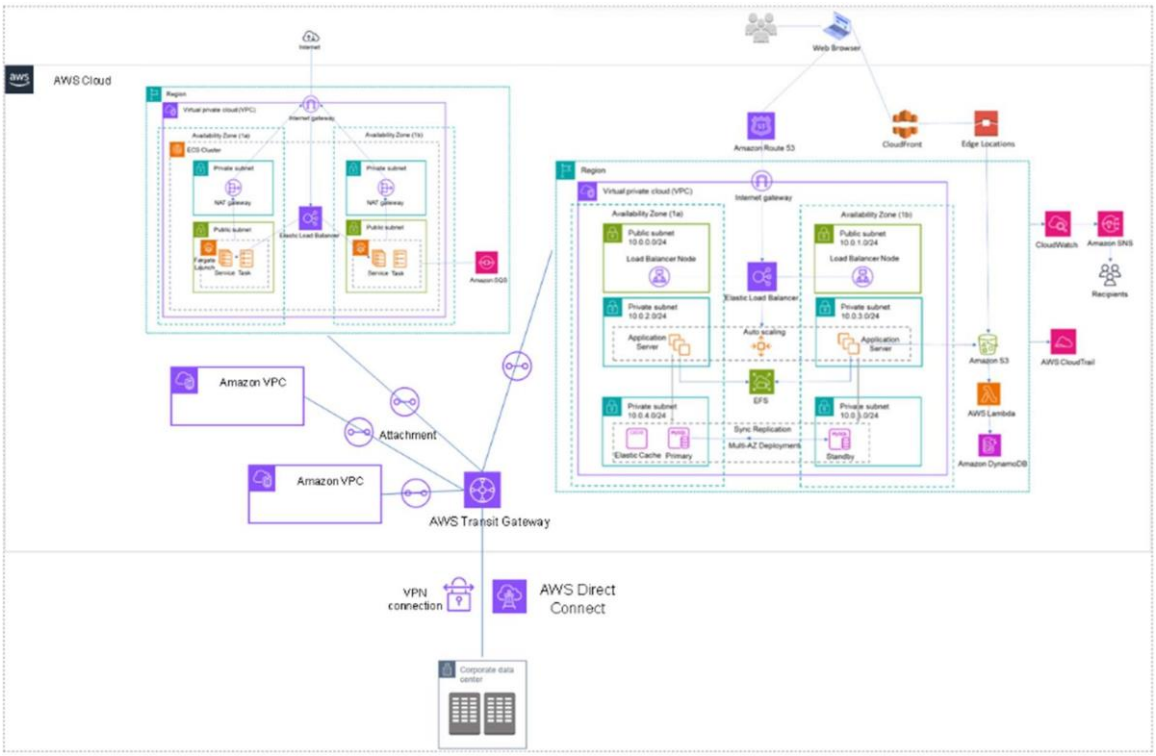
Blackbuck Engineers Pvt. Ltd

Road No 36, Jubilee Hills, Hyderabad

AWS Services acquired during this internship program (Initial Architecture)



AWS Services acquired during this internship program (Complete architecture)



AWS Services acquired during this internship program

- **EC2(Elastic Compute Cloud)-**Virtual servers in the cloud that can be easily scaled up or
  down to meet computing needs.
- **VPC(Virtual Private Cloud)-** Isolated virtual network environment that allows users to provision AWS resources in a secure and controlled manner.
- **EFS(Elastic File System)-** Scalable and fully managed file storage system designed for use with AWS EC2 instances.
- **RDS(Relational Database Service)-** Managed service that provides easy setup, operation and scaling of relational databases, supporting various database engines,
- **S3(Simple Storage Service)-** Object storage service for storing and retrieving any amount of data, designed for high durability, availability, and scalability.
- **IAM (AWS Identity and Access Management)-** Service for managing user identities, access and permissions to securely control access to AWS resources.
- **CloudWatch-**Monitoring and observability service that collects and tracks metrics, logs and events, allowing for real-time resource and application monitoring.
- **Lambda-**Serverless computing service that allows users to run code without provisioning or managing servers, paying only for the compute time consumed.

Team Members:

- Navya Sahithi Pitani (20B91A12F6)
- Yamini Lakshmi Jenu (21B95A1205)
- Navya Gopisetti(21B95A1204)
- Siri Chandana Varre(20B91A12I3)

Title:

Building an Architecture for Application and Database Deployment

Abstract

This Project presents an architecture for creating a Virtual Private Cloud (VPC) with interconnected public and private subnets, using an Internet Gateway (IGW) and Network Address Translation (NAT). The architecture enables secure connectivity and efficient communication between resources within the VPC and the internet.

Key components of the architecture include VPC configuration, a public subnet connected to an IGW for public access, and a private subnet connected to a NAT gateway or instance for outbound internet connectivity. Routing and security measures are implemented to control traffic flow and enforce security policies.

Implementing this architecture allows organizations to establish a secure and flexible network infrastructure, separating resources based on connectivity needs while ensuring controlled access. The architecture provides scalability and reliability while minimizing exposure to external threats.

This Project serves as a concise guide for building a VPC architecture , connecting public and private subnets with an IGW and NAT. It outlines the steps, considerations, and best practices for designing a secure network setup within the cloud environment, facilitating the deployment of cloud-based applications.

TABLE OF CONTENTS

Services used in the project 3
Skeleton architecture 3
Terminal Blueprint 4
Introduction to Cloud computing 4
Advantages of Cloud 5
Virtualization Architecture 5
Types of Cloud Computing based on service models 6
Cloud Deployment models 6
Cloud service providers 6
Introduction to Amazon Web Services 7
Why Customer move to AWS 7
Gartner Magic Quadrant for Cloud Infrastructure 8
AWS Global Infrastructure 9
What to Consider when Selecting a Region for your Workloads 9
Shared responsibility model 10
AWS Documentation 11
AWS Support Plans 11
Billing Dashboard 11
Ways to access the AWS Cloud Platform 12
AWS Free tier 12
Total Cost of Ownership(TCO) 7
Available AWS Certifications 13
Importance of AWS IAM 14
AWS Root User Vs IAM User 16

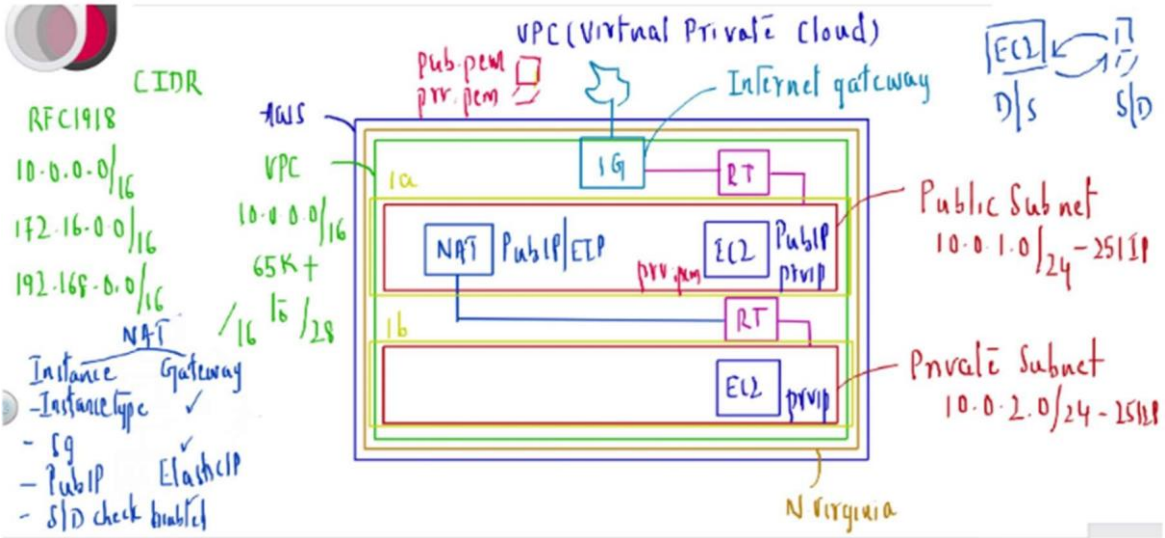
AWS EC2 17
AWS VPC

Execution Steps 28
Conclusion 40

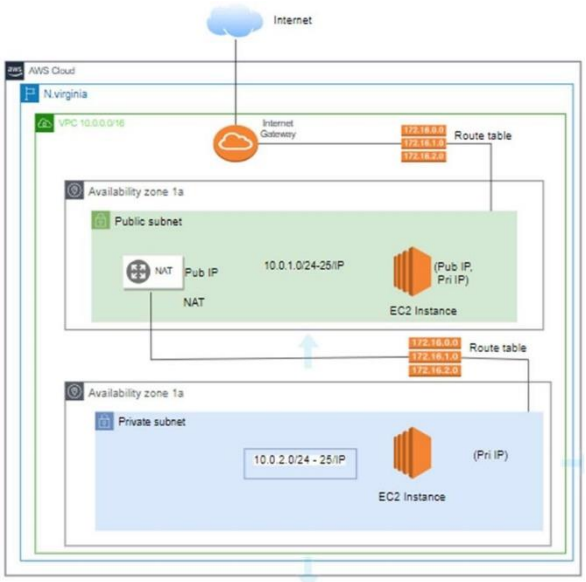
Services used in the project

- ❖ EC2 (Elastic Compute Cloud)
- ❖ VPC (Virtual Private Cloud)

Skeleton architecture



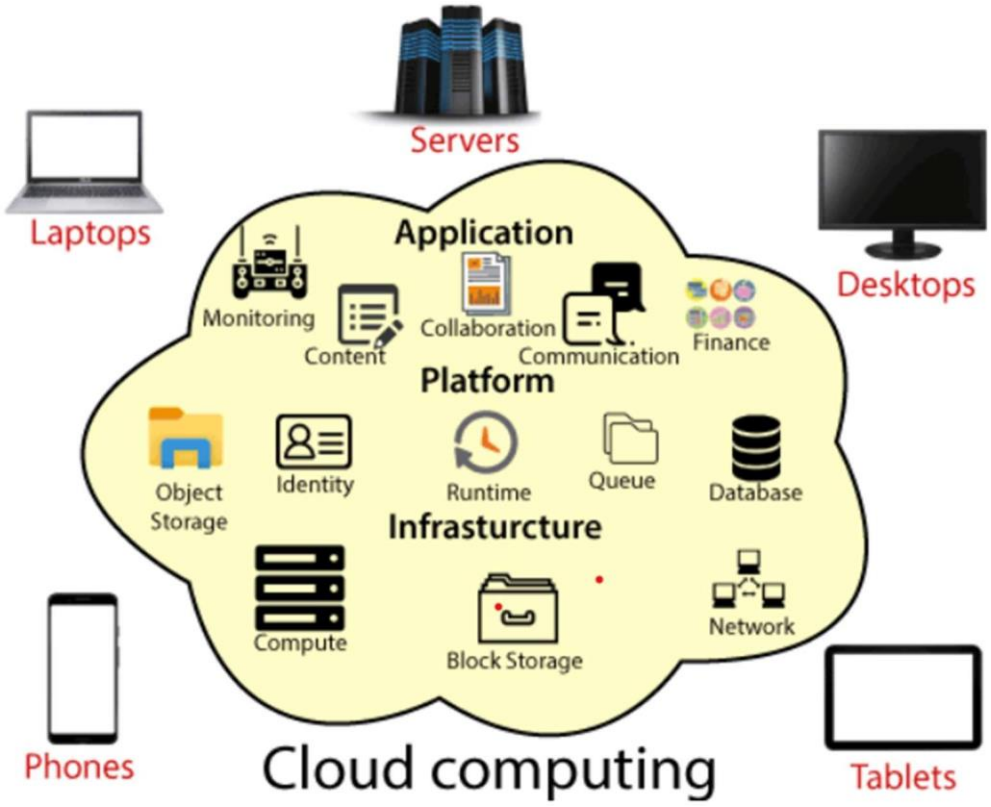
Terminal Blueprint



Introduction to Cloud Computing

Cloud Computing is the delivery of computing services such as servers, storage, databases, networking, software, analytics, intelligence, and more, over the Cloud (Internet).

Cloud Computing provides an alternative to the on-premises datacentre. With an on-premises datacentre, we have to manage everything, such as purchasing and installing hardware, virtualization, installing the operating system, and any other required applications, setting up the network, configuring the firewall, and setting up storage for data. After doing all the set-up, we become responsible for maintaining it through its entire lifecycle.



The cloud environment provides an easily accessible online portal that makes handy for the user to manage the compute, storage, network, and application resources.

Advantages of Cloud

1. **No Capital Investment**

- It reduces the huge capital costs of buying hardware and software.

2. **Only Operational costs with pay as go prices**

3. **Flexibility in Capacity**

4. **Speed and Agility**

- Resources can be accessed in minutes, typically within a few clicks.

5. **Users can concentrate more on their core business processes with no Datacenter maintenance**

6. **Go Global in Minutes**

7. **Productivity**

While using cloud computing, we put less operational effort. We do not need to apply patching, as well as no need to maintain hardware and software. So, in this way, the IT team can be more productive and focus on achieving business goals.

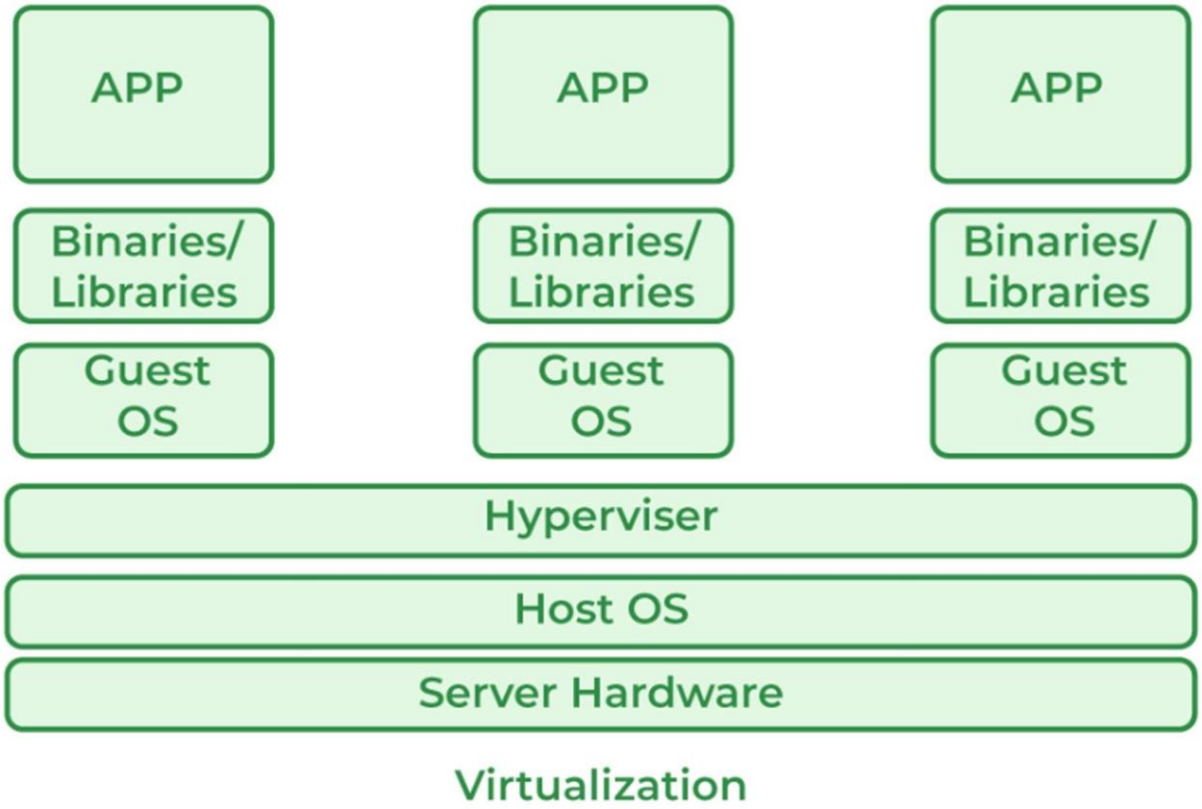
8. **Security**

Many cloud vendors offer a broad set of policies, technologies, and controls that strengthen our data security.

Virtualization Architecture

Virtualization is a technique how to separate a service from the underlying physical delivery of that service. It is the process of creating a virtual version of something like computer hardware. It was initially developed during the mainframe era. It involves using specialized software to create a virtual or software-created version of a computing resource rather than the actual version of the same resource. With the help of Virtualization, multiple operating systems and applications can run on the same machine and its same hardware at the same time, increasing the utilization and flexibility of hardware.

In other words, one of the main cost-effective, hardware-reducing, and energy-saving techniques used by cloud providers is Virtualization. Virtualization allows sharing of a single physical instance of a resource or an application among multiple customers and organizations at one time. It does this by assigning a logical name to physical storage and providing a pointer to that physical resource on demand. The term virtualization is often synonymous with hardware virtualization, which plays a fundamental role in efficiently delivering Infrastructure-as-a-Service (IaaS) solutions for cloud computing. Moreover, virtualization technologies provide a virtual environment for not only executing applications but also for storage, memory, and networking.

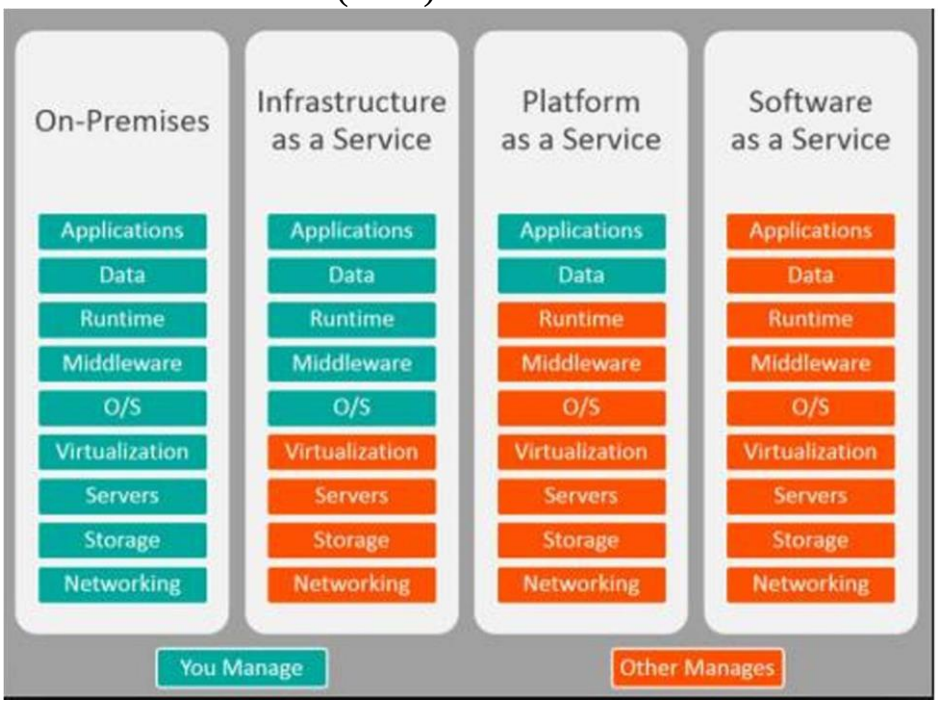


Types of Cloud Computing based on service models

Cloud computing is usually described in one of two ways. Either based on the deployment model, or on the service that the cloud is offering.

Based on Service models, we are speaking of either

- Infrastructure-as-a-Service(IaaS)
- Platform-as-a-Service(PaaS)
- Software-as-a-Service(SaaS)



Comparison

IaaS
Vs
PaaS
Vs
SaaS

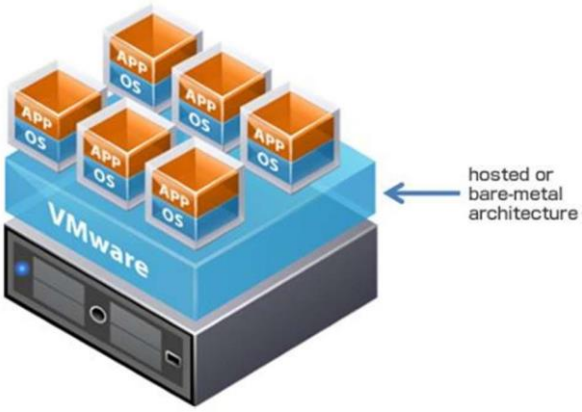
Cloud Deployment models

Based on Deployment models, we have

- Public Cloud
- Private Cloud
- Hybrid Cloud

<u>Public Cloud</u>

Public cloud usually uses shared resources and it also means, if all parts of the application run in the cloud which built on low-level infrastructure prices.



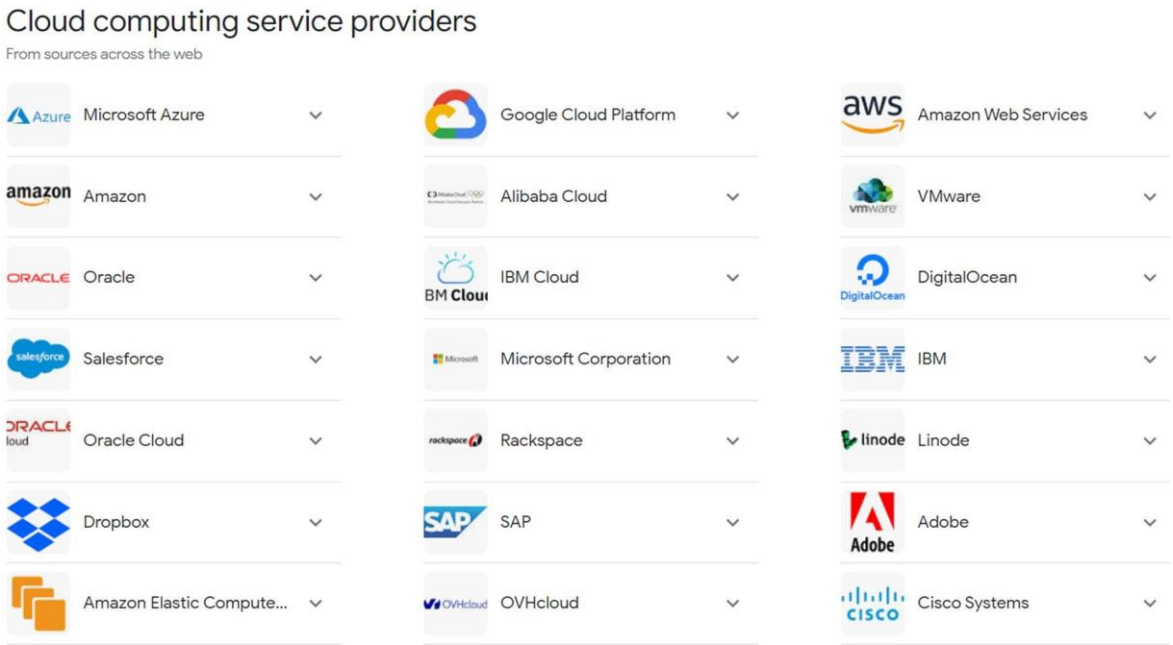
Private Cloud

The deployment of resources on-premises, using virtualization and resource management tools, is sometimes called the “private cloud”. It also means using a cloud infrastructure (network) solely by one customer/organization.

Hybrid Cloud

A hybrid deployment is a way to connect infrastructure and applications between cloud-based resources and existing resources that are not located in the cloud.

Cloud Service providers



Introduction to Amazon Web Services

Amazon Web Services (AWS), a subsidiary of Amazon.com, has invested billions of dollars in IT resources distributed across the globe. These resources are shared among all the AWS account holders across the globe. These accounts themselves are entirely isolated from each other. AWS provides on-demand IT resources to its account holders on a pay-as-you-go pricing model with no upfront cost. Amazon Web services offer flexibility because you can only pay for services you use or you need. Enterprises use AWS to reduce capital expenditure of building their own private IT infrastructure (which can be expensive depending upon the enterprise's size and nature). AWS has its own Physical fiber network that connects with Availability zones, regions, and Edge locations. All the maintenance cost is also bared by the AWS which saves a fortune for the enterprises.

Advantages of Amazon Web Services

- AWS allows you to easily scale your resources up or down as your needs change, helping you to save money and ensure that your application always has the resources it needs.
- AWS provides a highly reliable and secure infrastructure, with multiple data centers and a commitment to 99.99% availability for many of its services.
- AWS offers a wide range of services and tools that can be easily combined to build and deploy a variety of applications, making it highly flexible.
- AWS offers a pay-as-you-go pricing model, allowing you to only pay for the resources you actually use and avoid upfront costs and long-term commitments.

Why Customer move to AWS

Public clouds are an excellent choice for the safest of workloads. Properly moving to cloud can provide a multitude of substantial and extensive benefits. Below are the top seven reasons why so many of our customers are utilizing our AWS services.

1. **Cost**:The pay-per-use is an attractive pricing model for start-ups and enterprises alike. Ultimately, users who may have highly variable computing needs may realize significant savings with their IT costs.

2. **Scalability**:AWS instances can be ramped up or down at the push of a button (given the environment is set up correctly of course). Organizations experiencing a large spike in traffic due to a hot selling product, a major news release or seasonality can truly appreciate this capability

3. **No capital expenses**:AWS works as an Infrastructure-as-a-Service (IaaS) which enables companies to establish their data center in the cloud with no investment in physical hardware or data center space. Rake in the savings or utilize it for your core business objectives, not IT.

4. **Speed & Agility:**Companies can develop and deploy applications faster by acquiring almost instant access to virtually unlimited computing power. Essentially, software providers can deploy their products in a Software-as-a-Service model, eliminating the need for clients to install the software locally or maintain their own internal data centers.

5. **No Latency:**With multiple data centers all around the world, you can be sure there is a data center very close to wherever you or your customers are located.

6. **Self-Service Model:**IT teams can make changes to their environments on the fly- no need for physical hardware upgrades!

7. **Easy Start-Up:**Launching small EC2 instances in AWS may be simple, but maintaining and monitoring these instances may prove to be more of a challenge. Additionally, more companies are utilizing the help of an AWS reseller or a certified consultant to take care of these day-to-day tasks to ensure max uptime.

Gartner Magic Quadrant for Cloud Infrastructure
AWS Global Infrastructure

AWS is a cloud computing platform that is globally available. Global infrastructure is a region around the world in which AWS is based. Global infrastructure is a bunch of high-level IT services which is shown below: AWS is available in 19 regions, and 57 availability zones in December 2018, and 5 more regions 15 more availability zones for 2019.

What to Consider when Selecting a Region for your Workloads

The AWS Cloud is an ever-growing network of Regions and points of presence (PoP), with a global network infrastructure that connects them together. With such a vast selection of Regions, costs, and services available, it can be challenging for startups to select the optimal Region for a workload. This decision must be made carefully, as it has a major impact on compliance, cost, performance, and services available for your workloads.

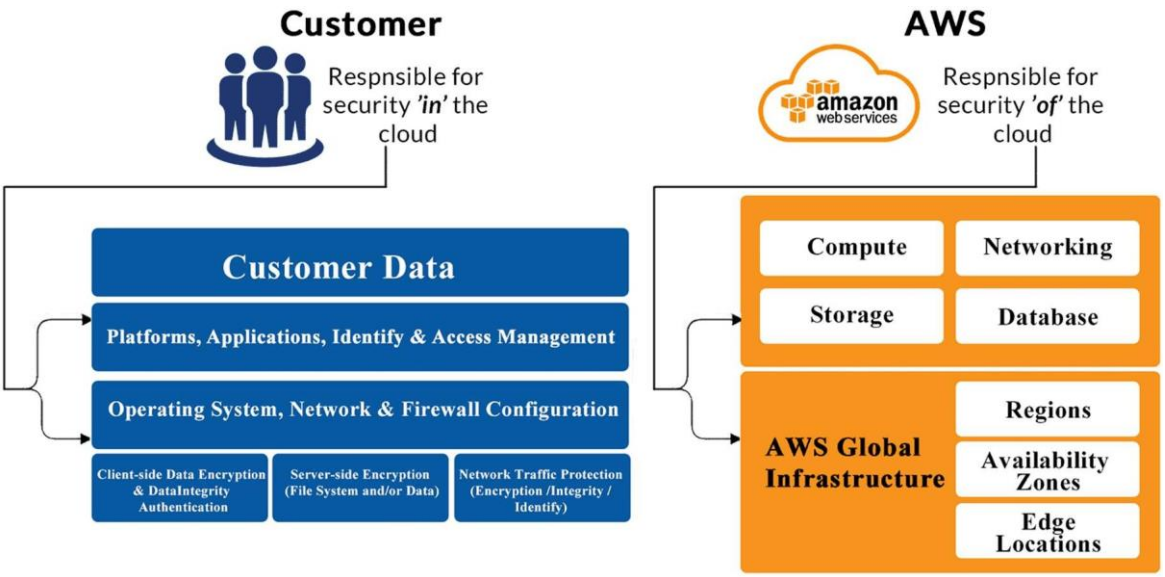
There are four main factors that play into evaluating each AWS Region for a workload deployment:

1. **Compliance**: If your workload contains data that is bound by local regulations, then selecting the Region that complies with the regulation overrides other evaluation factors. This applies to workloads that are bound by data residency laws where choosing an AWS Region located in that country is mandatory.
2. **Latency**:A major factor to consider for user experience is latency. Reduced network latency can make a substantial impact on enhancing the user experience. Choosing an AWS Region with close proximity to your user base location can achieve lower network latency. It can also increase communication quality, given that network packets have fewer exchange points to travel through.
3. **Cost**:AWS services are priced differently from one Region to another. Some Regions have lower costs than others, which can result in a cost reduction for the same deployment.
4. **Services and features**:Newer services and features are deployed to Regions gradually. Although all AWS Regions have the same service level agreement (SLA), some larger Regions are usually first to offer newer services, features, and software releases. Smaller Regions may not get these services or features in time for you to use them to support your workload.

Evaluating all these factors can make coming to a decision complicated. This is where your priorities as a business should influence the decision.

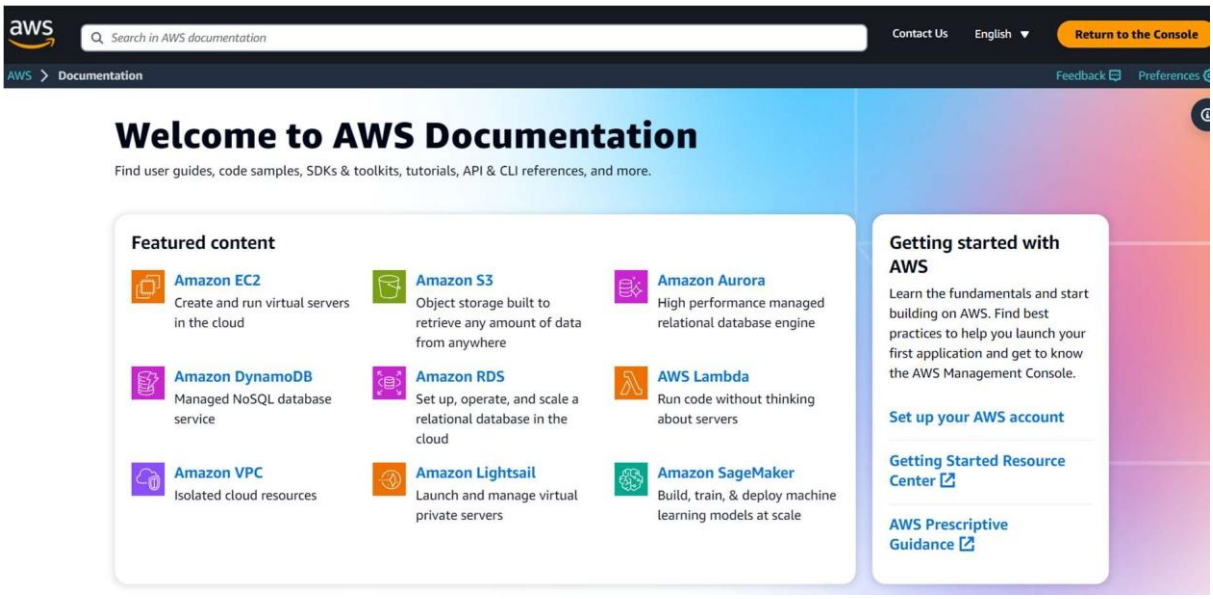
Shared responsibility model

The Shared Responsibility Model is a cloud security structure that specifies the security requirements of cloud service providers and users to ensure accountability. Simply said, a cloud vendor offers a variety of cloud services to its customers. One offers the service, while the other consumes it. The vendor is accountable for the service provided, while the user is responsible for the service consumption.
The biggest challenge that organizations face is the confusion of the responsibilities leading to security compromisation. This confusion gives hackers a blind spot to attack, and many reports claimed the improperly shared security responsibilities as the culprit for various security incidents.
Thus, Amazon Web Service (AWS) established the AWS Shared Responsibility Model to clarify responsibilities.



- **AWS Responsibility:** AWS is responsible for protecting the infrastructure that runs all the AWS services. In other words, AWS control, operate and manage the components from the host operating system and virtualization layer that is down to the physical layer in which the service operates.
- **Customer Responsibility:** The customer's responsibility depends on the AWS service used and the configuration they need to perform to secure that service. In other words, customers need to manage the guest operating system, including security patches and application software. Also, they need to configure the AWS-provided security controls like security groups, network access control, and IAM (Identity and Access Management).

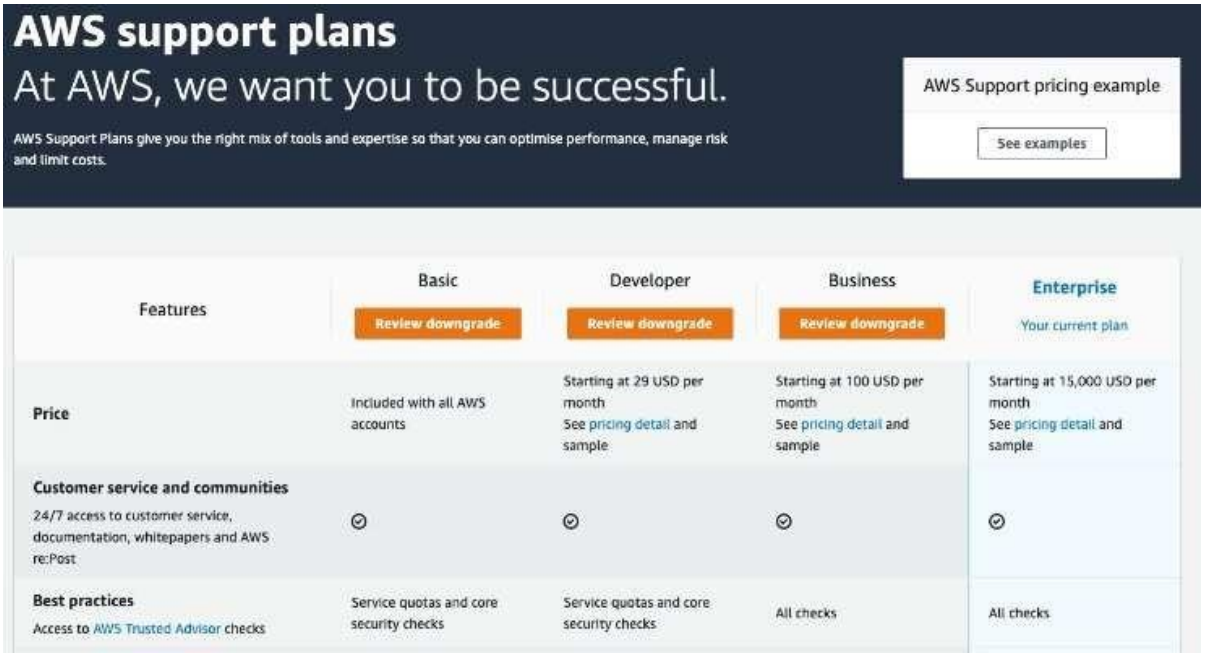
AWS Documentation



AWS Support Plans

AWS offers four different support plans.

- Basic
- Developer
- Business
- Enterprise



Features	Basic Review downgrade	Developer Review downgrade	Business Review downgrade	Enterprise Your current plan
Price	Included with all AWS accounts	Starting at 29 USD per month See pricing detail and sample	Starting at 100 USD per month See pricing detail and sample	Starting at 15,000 USD per month See pricing detail and sample
Customer service and communities 24/7 access to customer service, documentation, whitepapers and AWS re:Post	✓	✓	✓	✓
Best practices Access to AWS Trusted Advisor checks	Service quotas and core security checks	Service quotas and core security checks	All checks	All checks

Billing Dashboard

The AWS Billing console contains features to pay your AWS bills and report your AWS cost and usage. You can also use the AWS Billing console to manage your consolidated billing if you're a part of AWS Organizations. Amazon Web Services automatically charges the credit card that you provided when you sign up for an AWS account. You can view or update your credit card information at any time, including designating a different credit card for AWS to charge.

Some Features of AWS Billing console:

- Manage your account
- View your bill
- Managing your payments
- AWS Purchase Order Management
- AWS Cost Categories
- Payment profile
- Consolidate billing for AWS Organizations

Ways to access the AWS Cloud Platform

AWS Cloud Platform

AWS consists of many cloud services that you can use in combinations tailored to your business or organizational needs

- To access the services, you can use
  - AWS Management Console
  - Command Line Interface (CLI)
  - Software Development Kits (SDKs)
- The **AWS Management Console** is a web application that comprises and refers to a broad collection of service consoles for managing AWS resources.
- The **AWS Command Line Interface (AWS CLI)** is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.
- Our **Software Development Kits (SDKs)** simplify using AWS services in your applications with an Application Program Interface (API) tailored to your programming language or platform.

AWS Free tier

The AWS Free Tier allows customers to explore and try AWS services free of charge up to specified limits for each service. The Free Tier is comprised of three different types of offerings, a 12-month Free Tier, an Always Free offer, and short-term trials.

These tier offers include 12 months of free usage following your initial sign-up date to AWS. When your 12-month free usage term expires, or if your application use exceeds the tiers, you simply pay standard, pay-as-you-go service rates.

Services of aws free tier

- Amazon EC2. Elastic Cloud Computing. 12 MONTHS FREE. ...
- Amazon Lightsail. Simple Virtual Private Servers. 3-MONTH FREE TRIAL. ...
- AWS Lambda. Serverless Computing. ALWAYS FREE. ...
- Amazon Elastic Container Registry. Fully Managed Container Registry. 12 MONTHS FREE. ...
- Elastic Load Balancing. Distribute Network Traffic.

Total Cost of Ownership(TCO)

AWS costs depend on three key components: compute, outbound data transfer, and storage. The AWS pricing model and the product's specific characteristics determine the charges for each of these components.

Data transfer

Generally, AWS does not charge for inter-service data transfer within a region or inbound transfer. However, there are certain exceptions, so it's important to check the data transfer costs before starting. AWS calculates your aggregated outbound data transfer across all services and charges based on the specified rate. The monthly AWS statement specifies this charge under "Data Transfer Out." The amount of data transferred determines the cost per GB.

Compute

AWS charges per hour for compute resources, billing you from when you launch each resource until termination (for on-demand resources). There is also an option to make reservations with predetermined, set costs. AWS offers several compute instance types, such as the Elastic Compute Cloud (EC2) instances, which allow you to optimize your computing costs.

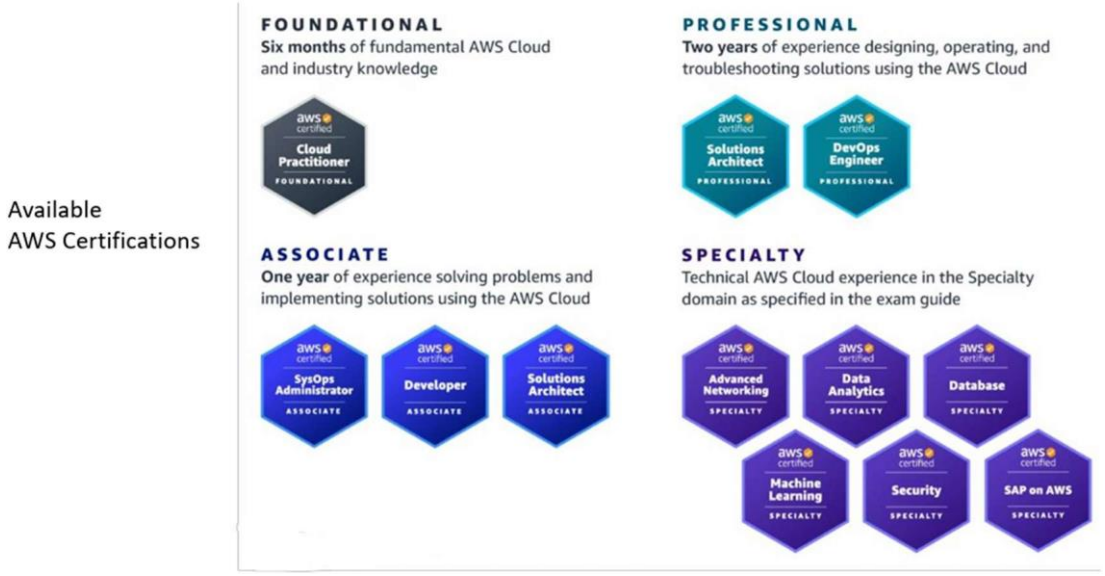
Storage

AWS usually charges for each GB of data storage. There are several storage classes for different use cases and data types, so choosing the right storage class for your needs is important.

The AWS cloud offers fixed and variable pricing models, providing flexibility and scalability at optimized costs. AWS offers several tools for calculating and managing costs, including resources to help you plan budgets, forecast spending, and track your usage and costs. These tools can provide recommendations for optimizing costs.

When calculating your TOC, you must consider the hidden, indirect costs that are difficult to predict, such as the cost of downtime, reduced productivity and other issues. The TOC assessment should include an evaluation of what-if scenarios, such as over-provisioning and changing requirements.

Available AWS Certifications



Importance of AWS IAM

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. With IAM, you can centrally manage permissions that control which AWS resources users can access. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

When you create an AWS account, you begin with one sign-in identity with complete access to all AWS services and resources. This identity is called the AWS account root user and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you don't use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform tasks only the root user can perform.

- **Centralised control of your AWS account:** You can control the creation, rotation, and cancellation of each user's security credentials. You can also control what data in the aws system users can access and how they can access it.
- **Shared Access to your AWS account:** Users can share the resources for the collaborative projects.
- **Granular permissions:** It is used to set a permission that user can use a particular service but not other services.
- **Identity Federation:** An Identity Federation means that we can use Facebook, Active Directory, LinkedIn, etc with IAM. Users can log in to

the AWS Console with the same username and password as we log in with the Active Directory, Facebook, etc.

- **Multifactor Authentication:** AWS provides multifactor authentication as we need to enter the username, password, and security check code to log in to the AWS Management Console.
- **Permissions based on Organizational groups:** Users can be restricted to the AWS access based on their job duties, for example, admin, developer, etc.
- **Networking controls:** IAM also ensures that the users can access the AWS resources within the organization's corporate network.
- **Provide temporary access for users/devices and services where necessary:** If you are using a mobile app and storing the data in an AWS account, you can do this only when you are using temporary access.
- **Integrates with many different aws services:** IAM is integrated with many different AWS services.
- **Supports PCI DSS Compliance:** PCI DSS (Payment Card Industry Data Security Standard) is a compliance framework. If you are taking credit card information, then you need to pay for compliance with the framework.
- **Eventually Consistent:** IAM service is eventually consistent as it achieves high availability by replicating the data across multiple servers within Amazon's data center around the world.
- **Free to use:** AWS IAM is a feature of the AWS account which is offered at no additional charge. You will be charged only when you access other AWS services by using an IAM user.

AWS Root User Vs IAM User

In Amazon Web Services (AWS), there are two different privileged accounts. One is defined as Root User (The account owner) and the other is defined as an IAM User.

The AWS Root User is the first cloud service identity created by default when you create your cloud service provider account. It is important to note—all cloud service providers have some form of root account. Depending on the provider, you may be allowed to have more than one such account. The more root accounts you have, the larger the risk surface. Because of its omnipotence, the AWS root account is a prime target for threat actors because it can control everything in your instance and in your account profile.

An AWS IAM user granted administrative privileges can do almost everything a root user can do, except for a few tasks that are restricted to the root account. If you delete the root account, which some security professionals

may advocate, these changes will be permanently inaccessible. The following list highlights some of the more critically important privileges only available to root:

- Closing your cloud service provider account. This is particularly important if you plan to change providers, or the account is ephemeral for testing or development
- Changing many of your cloud service provider account settings, like the root email address (the root account should never be a person's email address, but rather always a generic alias email account managed by the business)
- Changing your support plan and billing information (using a credit card or other method that will change due to an expiration date can cause issues if the root account is deleted)
- Enabling or disabling specific security controls, like MFA, to manage key runtime parameters, like deletion. (these settings should be implemented upfront and never be changed)

Services used in the Project

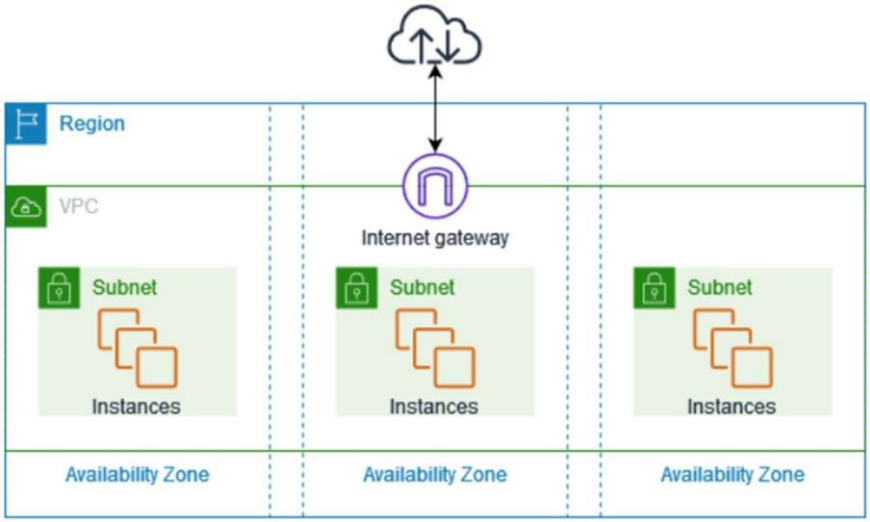
➢ AWS EC2(Elastic Compute Cloud)

Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) provides on-demand, scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 reduces hardware costs so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. You can add capacity (scale up) to handle compute-heavy tasks, such as monthly or yearly processes, or spikes in website traffic. When usage decreases, you can reduce capacity (scale down) again.
Amazon EC2 supports the processing, storage, and transmission of credit card data by a merchant or service provider, and has been validated as being compliant with Payment Card Industry (PCI) Data Security Standard (DSS).

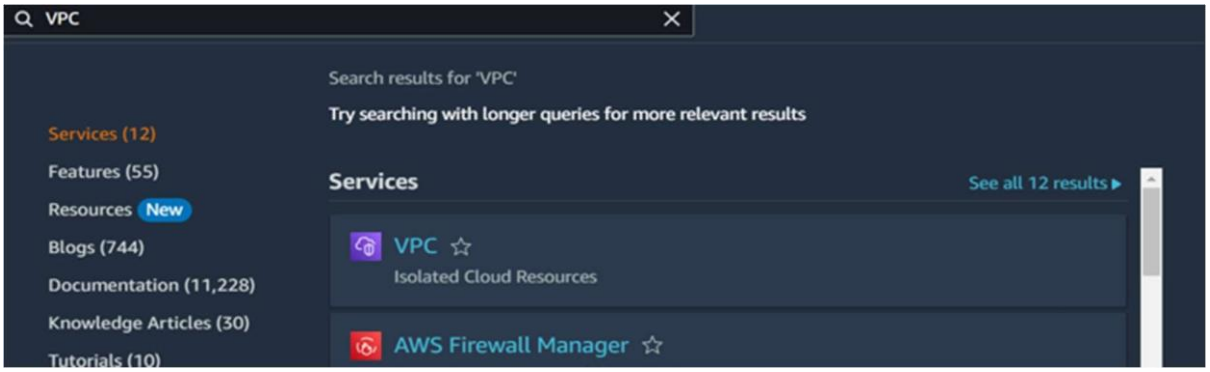
➢ AWS VPC(Virtual Private Cloud)

With Amazon Virtual Private Cloud (Amazon VPC), you can launch AWS resources in a logically isolated virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

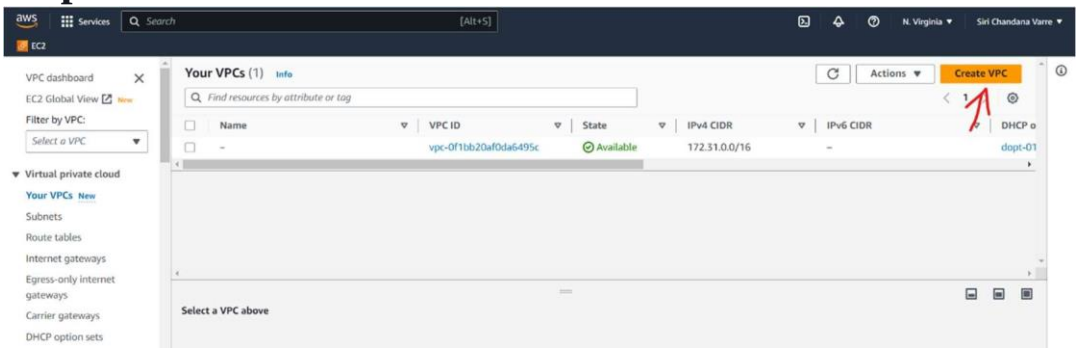


The following diagram shows an example VPC. The VPC has one subnet in each of the Availability Zones in the Region, EC2 instances in each subnet, and an internet gateway to allow communication between the resources in your VPC and the internet.

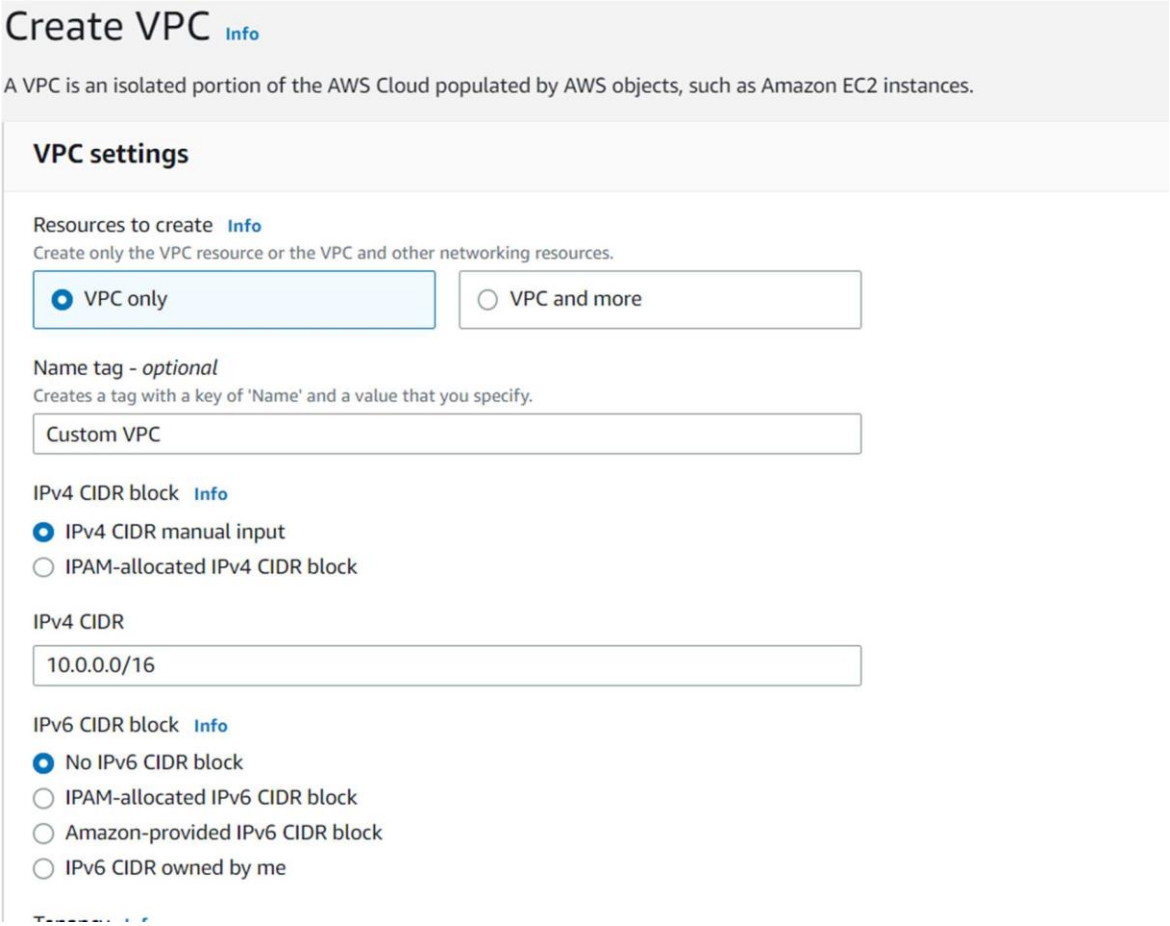
Step 1: Select VPC Service

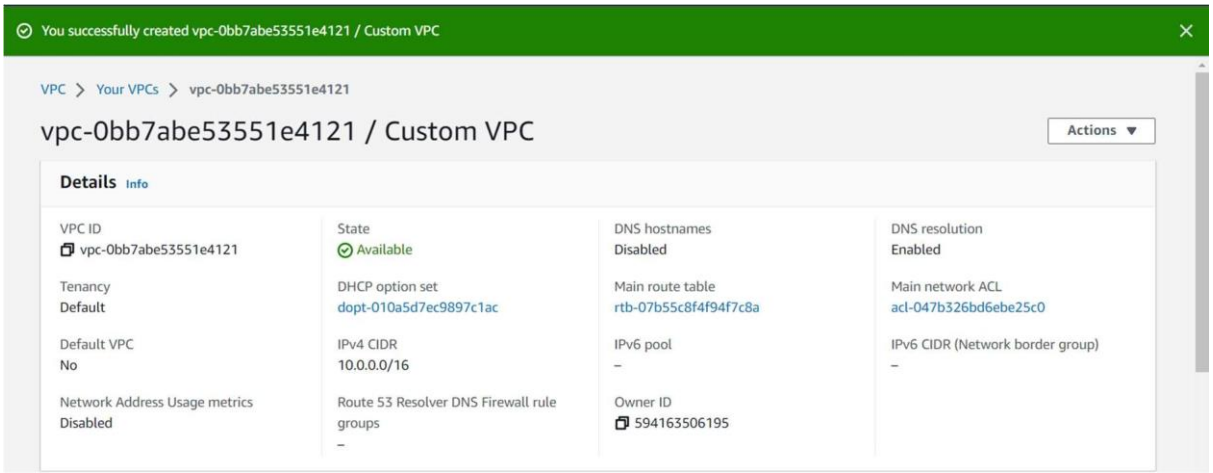


Step 2: Create new VPC

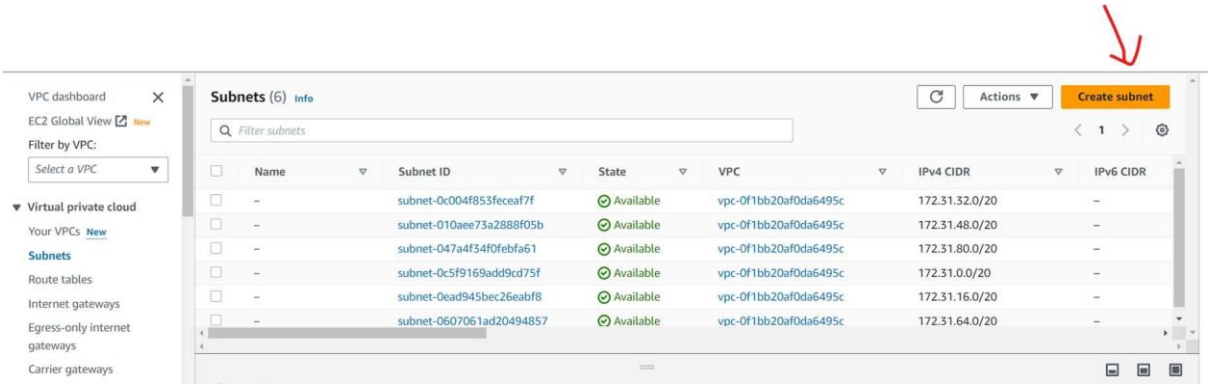


- Name the vpc and give IPV4 CIDR and keep the Tenancy as Default and then create the VPC.



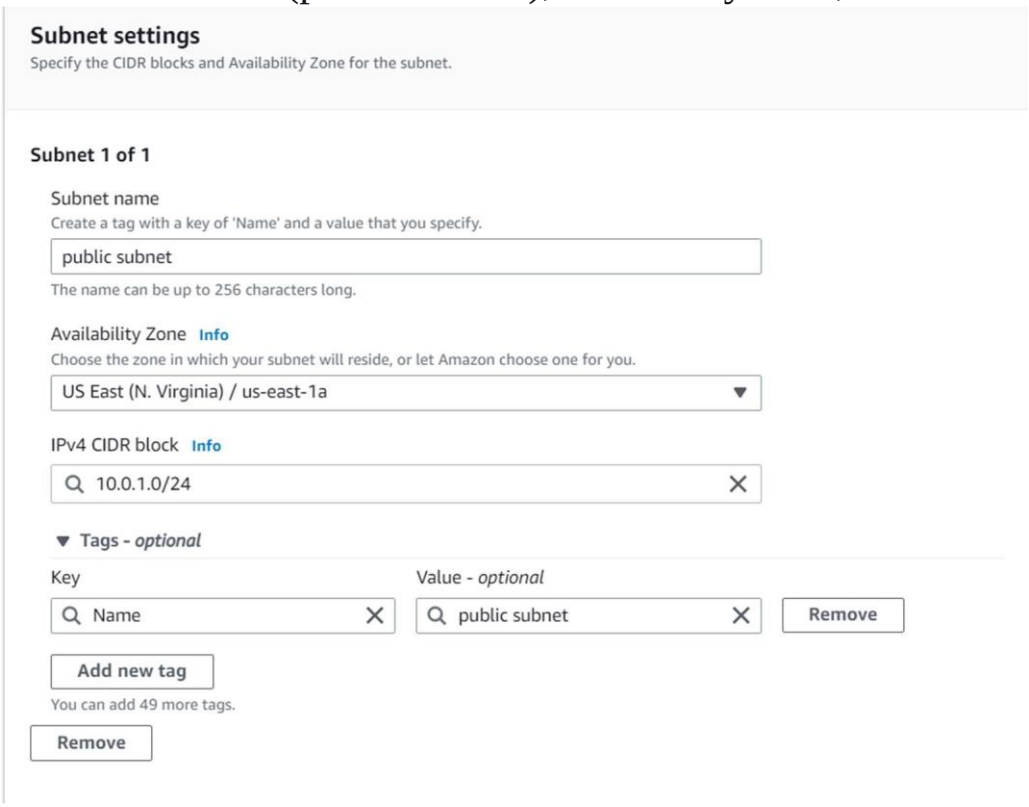


Step 3: Go to Subnets>click on create Subnet(creating public subnet)

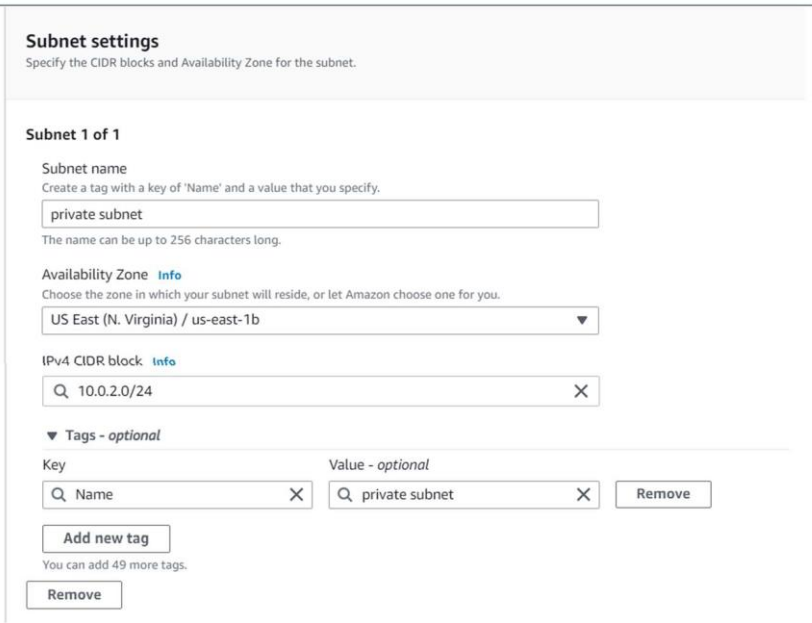


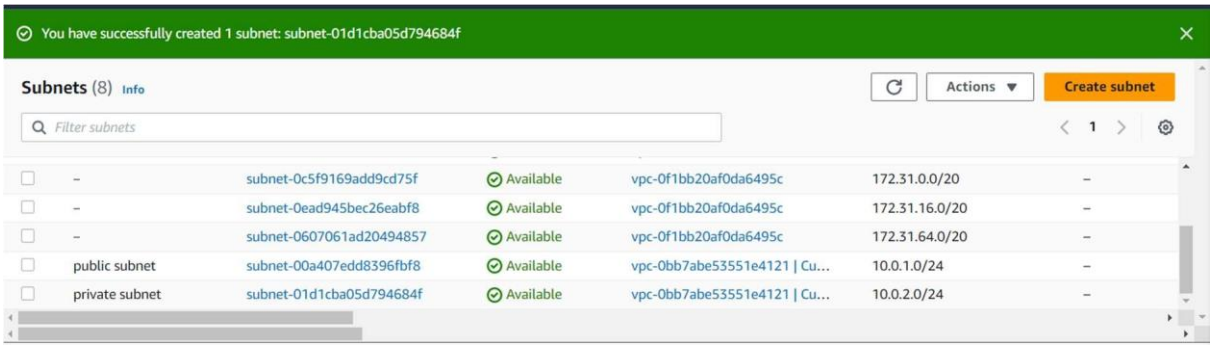
- Select VPC that we created(Custom VPC)

- Give subnet name(public subnet),Avalability zone,IPV4 CIDR

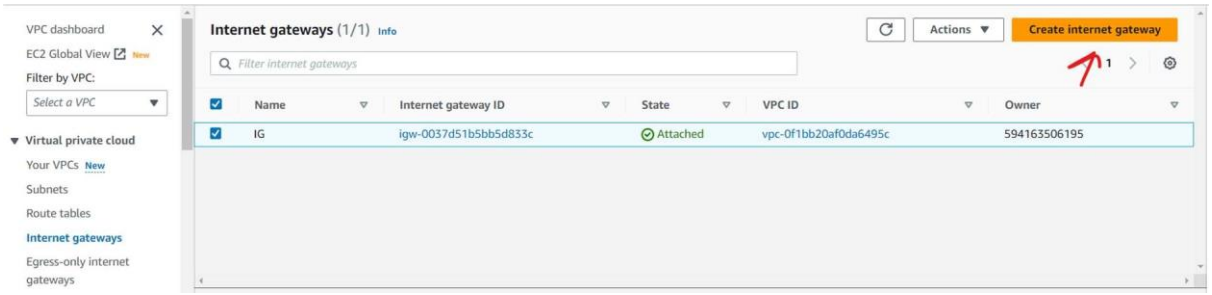


- Creating another subnet(private subnet)

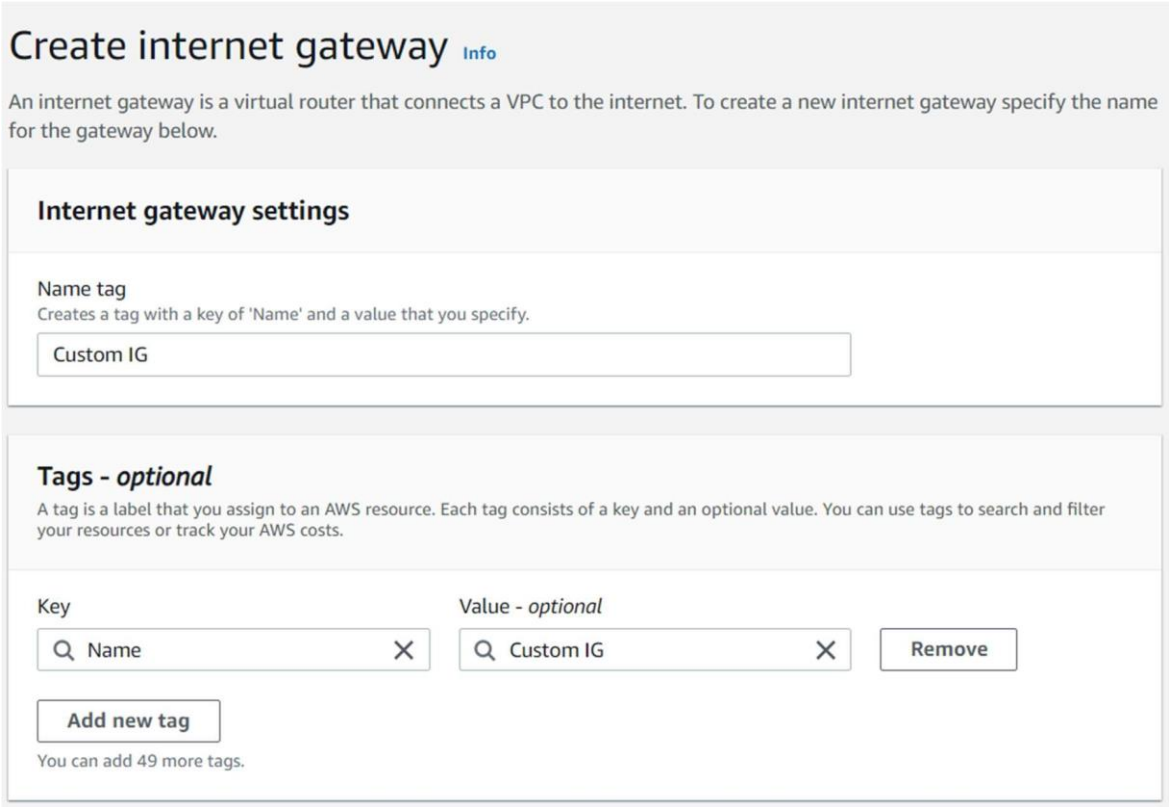




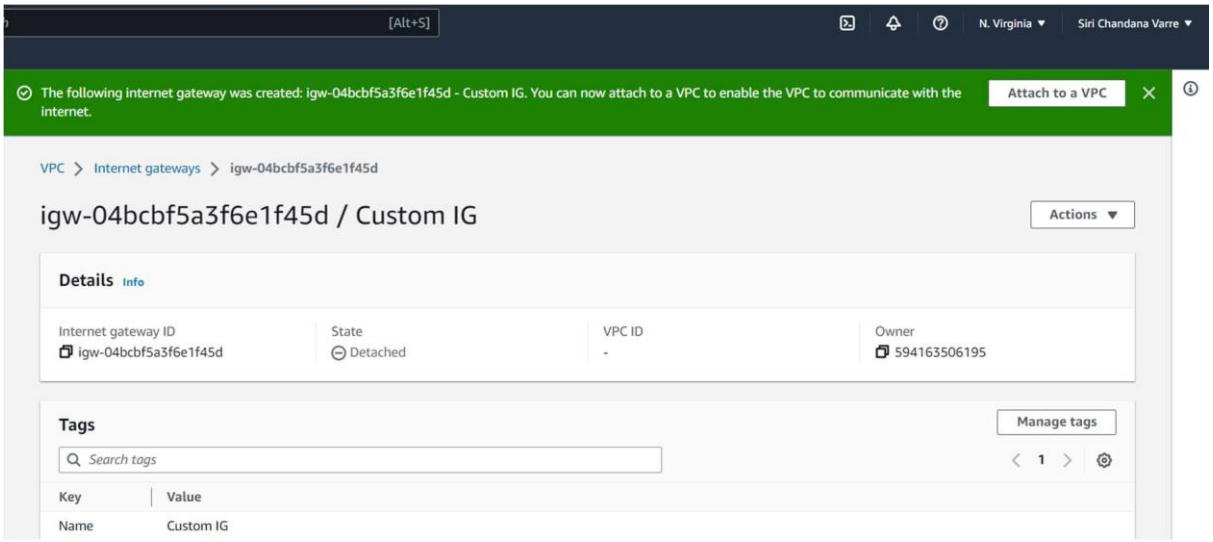
Step 4: Go to Internet Gateway>click on create Internet Gateway



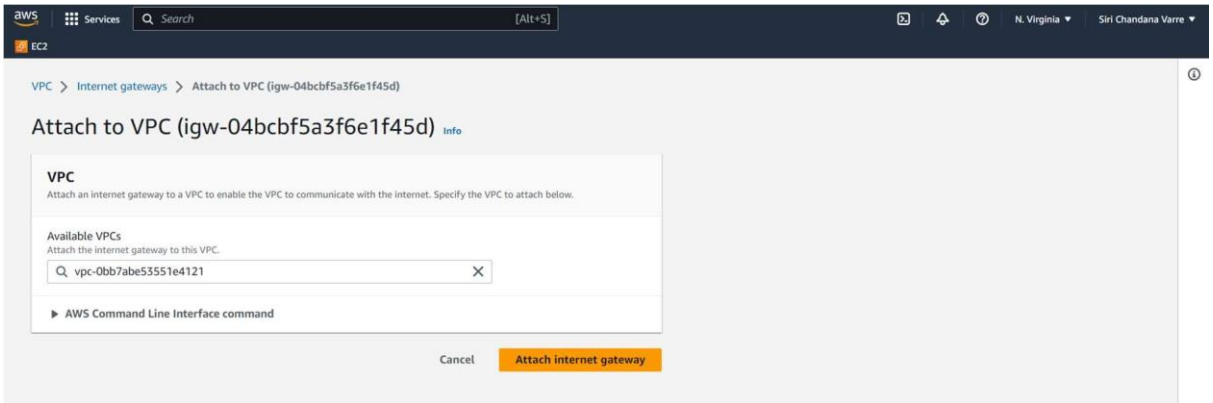
- Give Name to it(Custom IG)>create



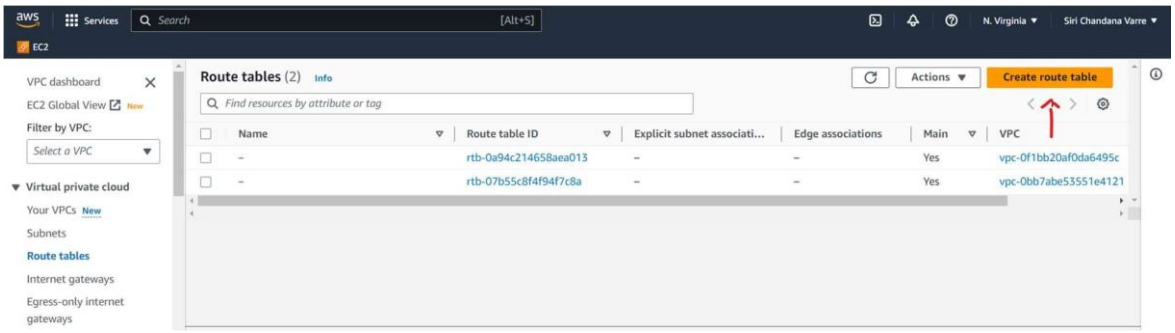
Step 5: Attach to VPC(Click on Attach to VPC)



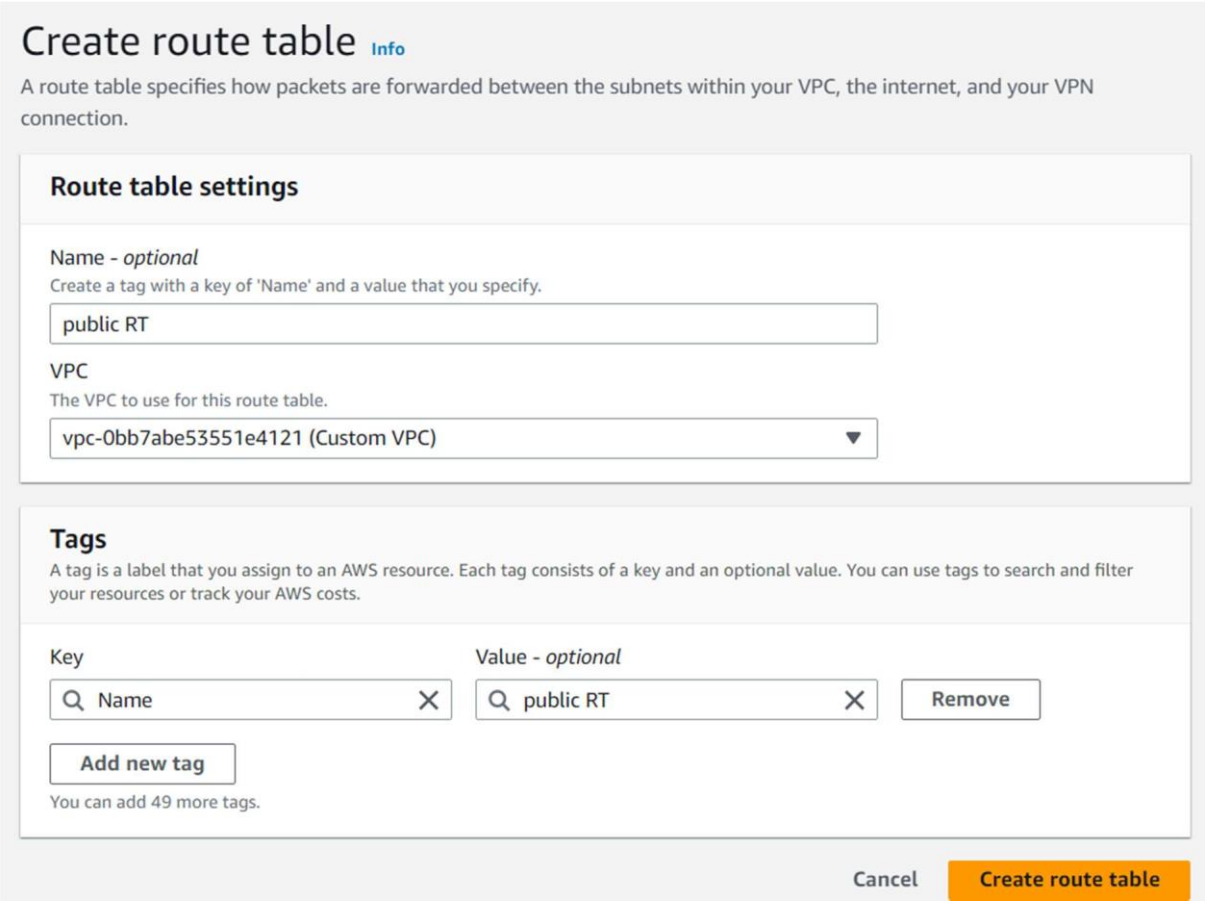
- Select corresponding created VPC



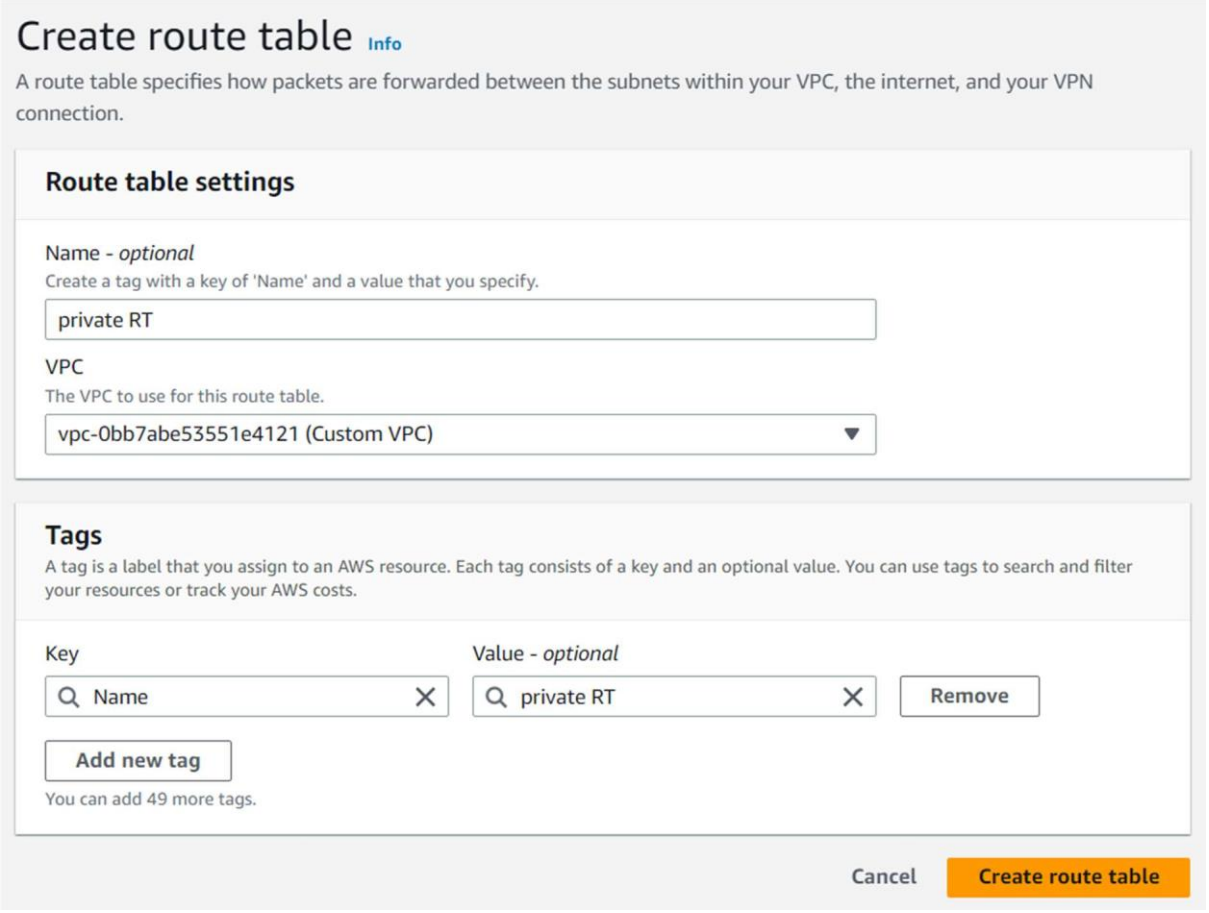
Step 6: Create Route Table(name-public RT)

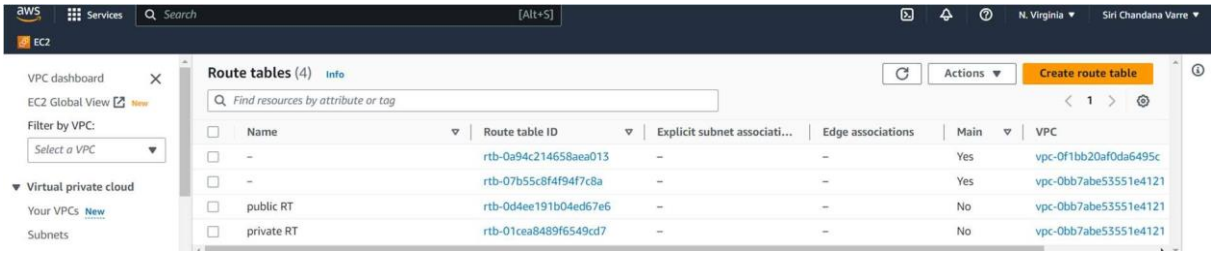


- Give Name to it(Public RT) and select Custom VPC >create route table

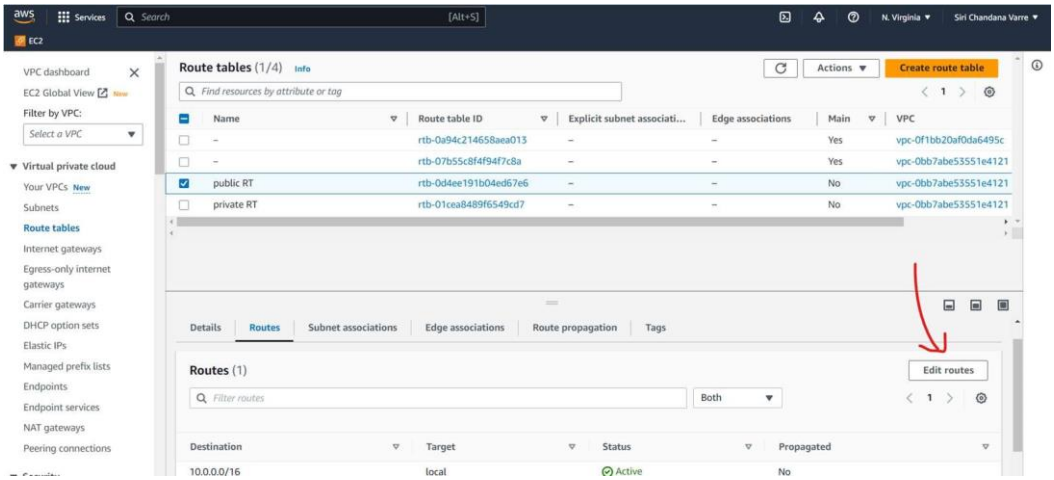


- Create another Routing Table(private RT)

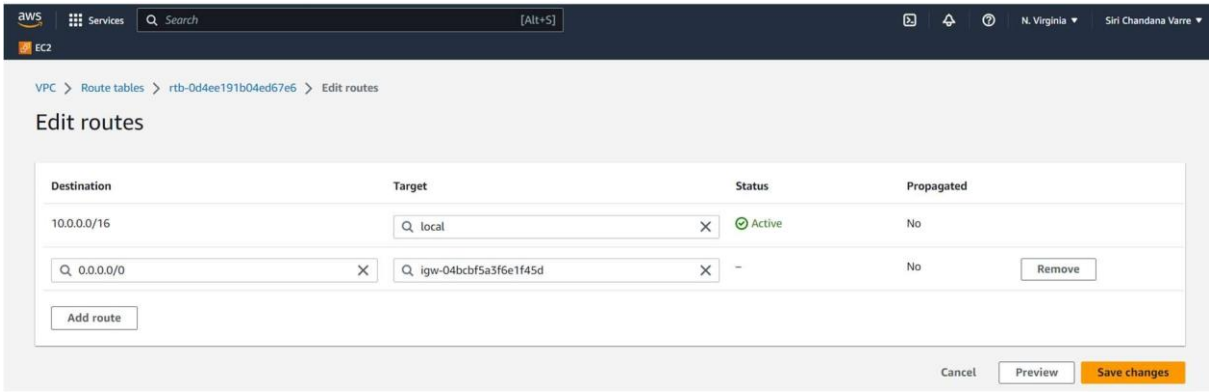




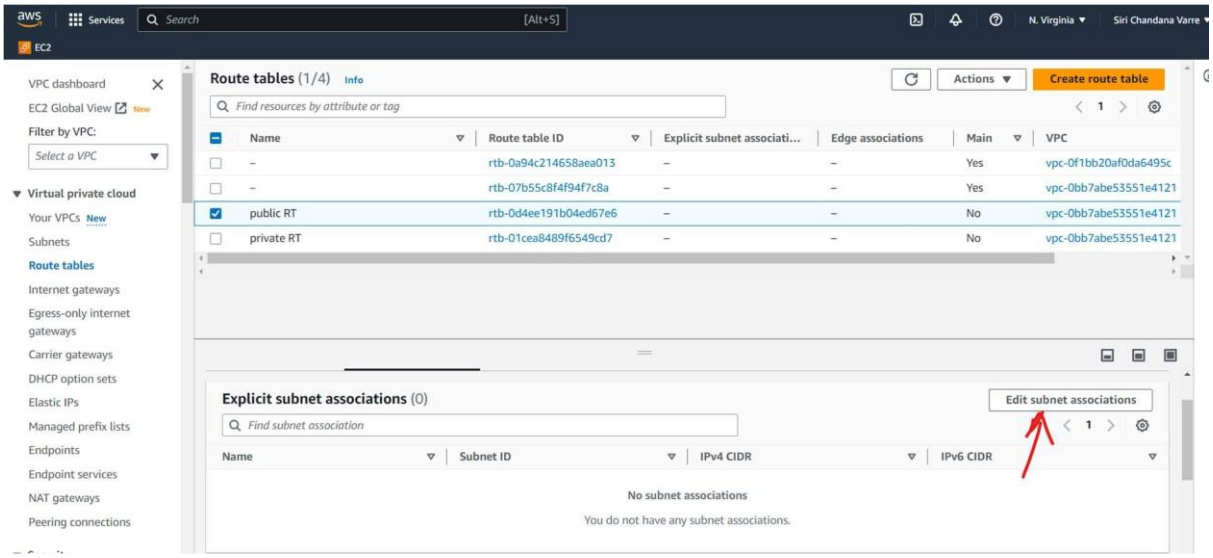
Step 7: Go to Route tables>public RT>Routes>Edit routes



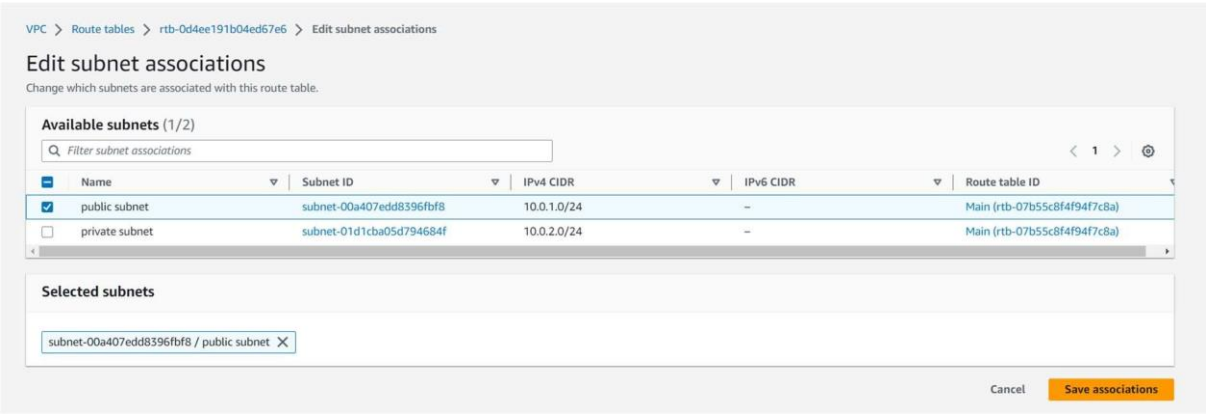
- Edit routes>Add Route>Destination-0.0.0.0/0,Target-Custom IG(which we created)>Save Changes.



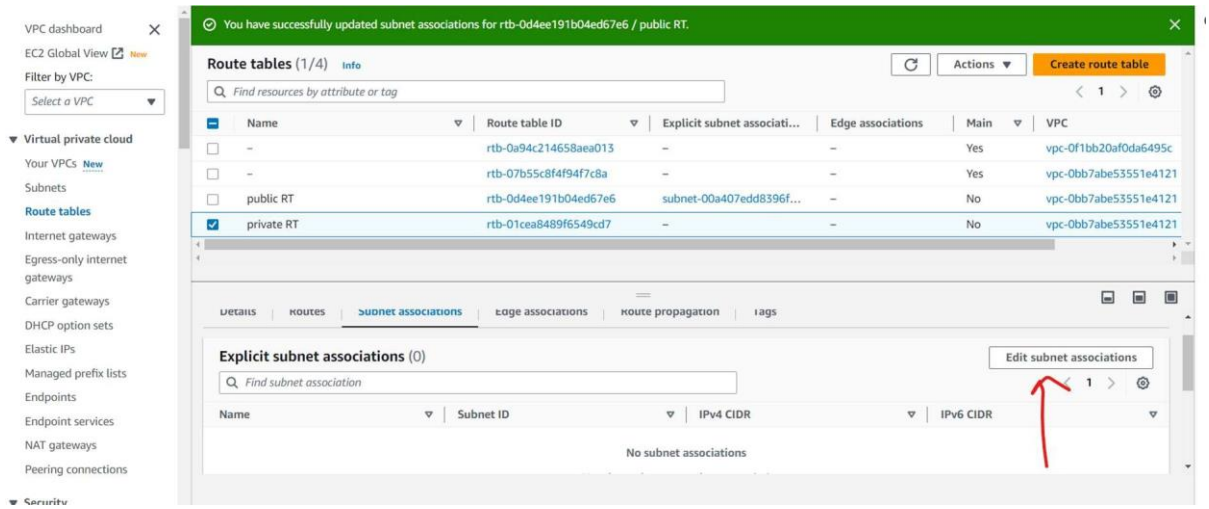
Step 8: Go to Route Tables>select Public RT>subnet Associations>Edit



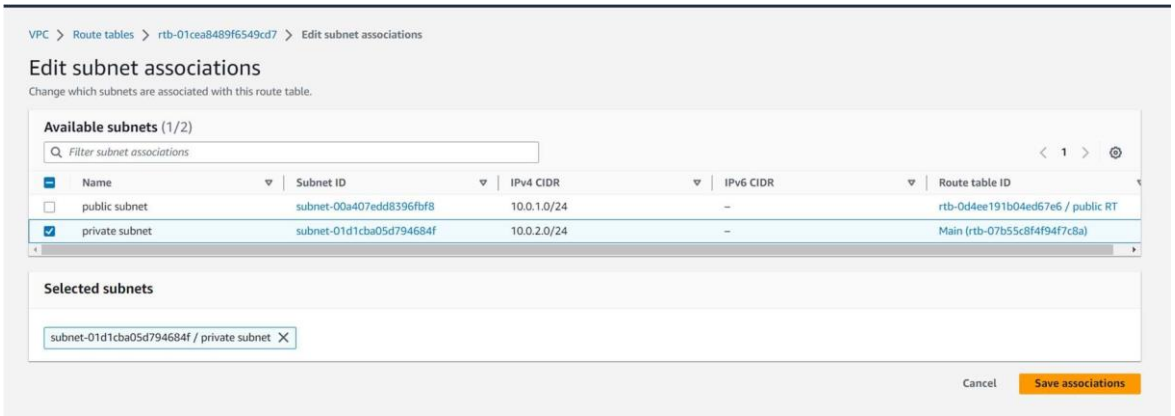
- Edit subnet associations>select public subnet>save associations



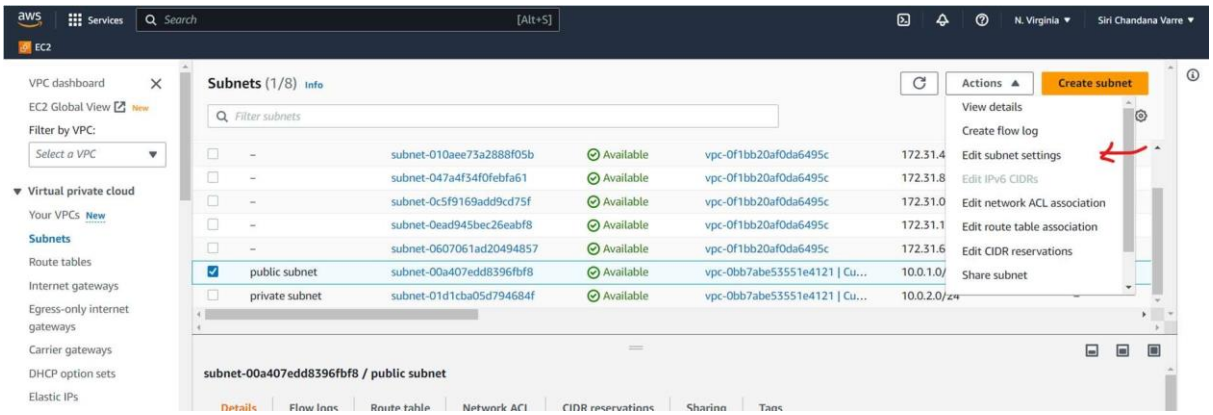
- Similarly,Go to Route Tables>Select private RT>Subnet Associations>Edit



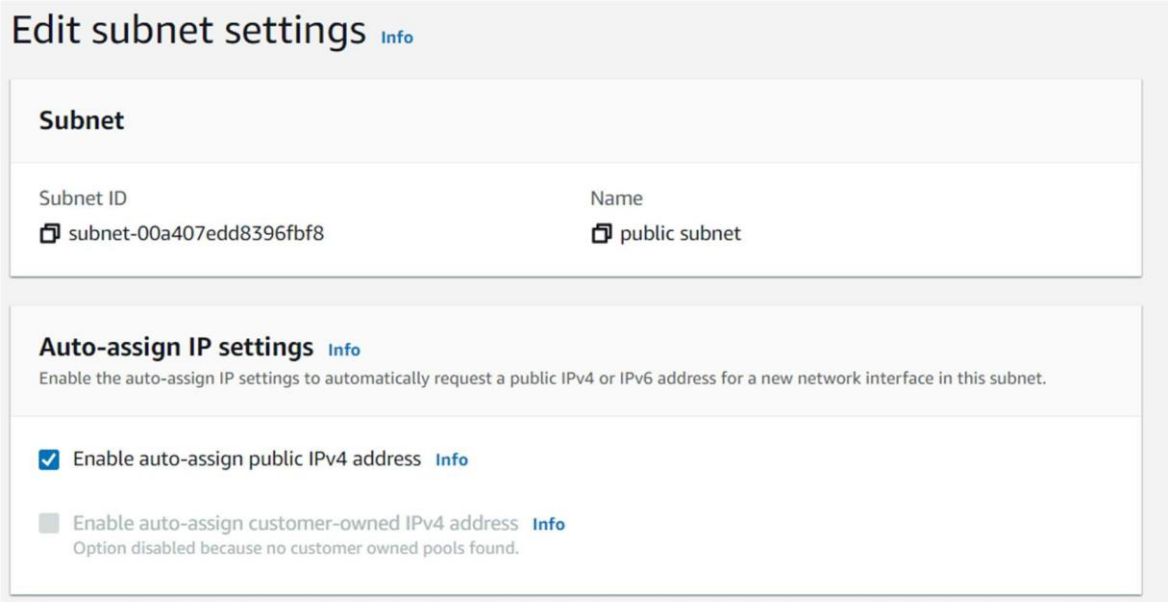
- Edit routes>select private subnet>save associations



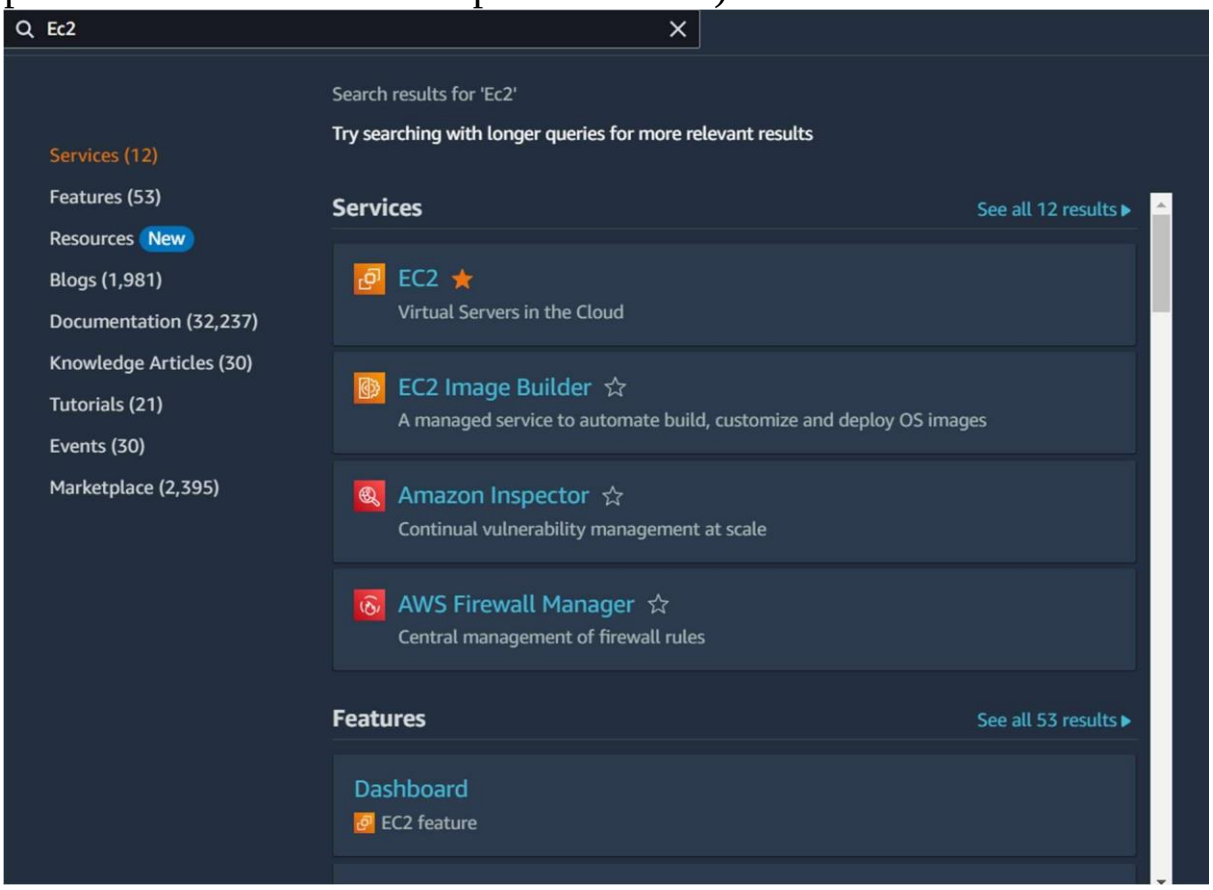
Step 9: Go to Subnets>select public subnet>Actions>Edit Subnet Settings



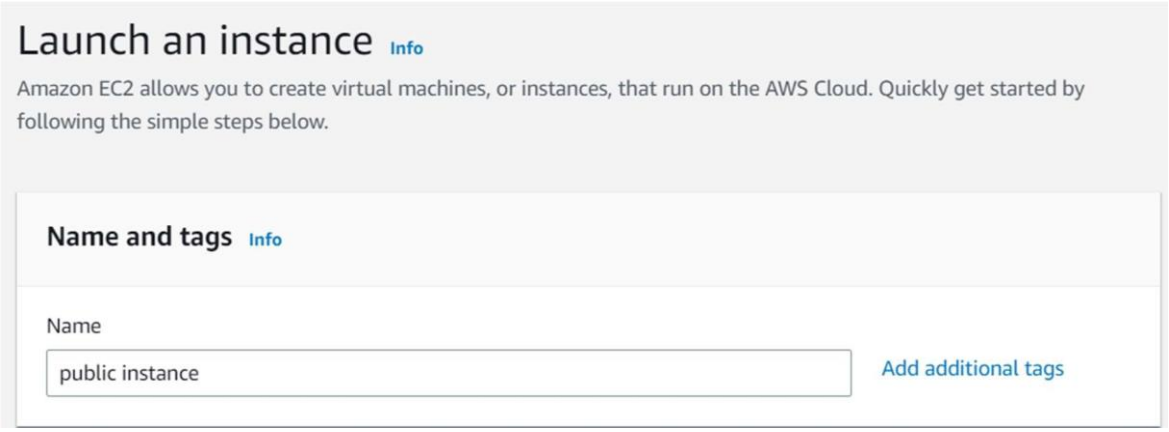
- Edit Subnet settings>Enable Auto-assign IP settings>Save



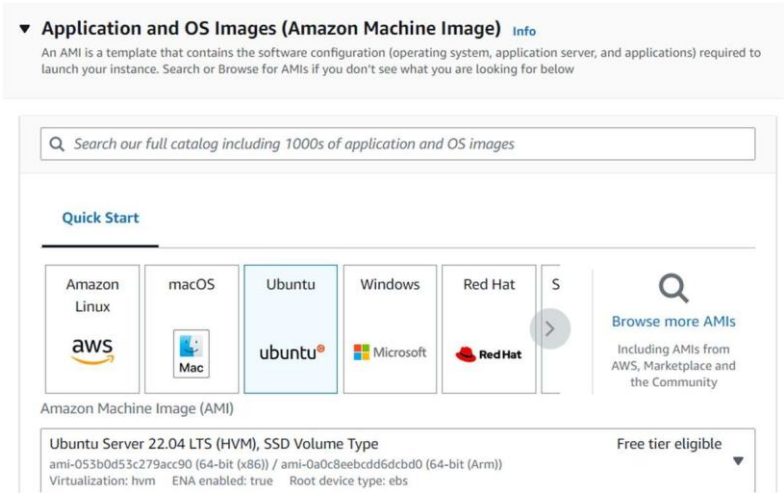
Step 10: Go to EC2 Service and Launch two instance(one for public subnet and other for private subnet)



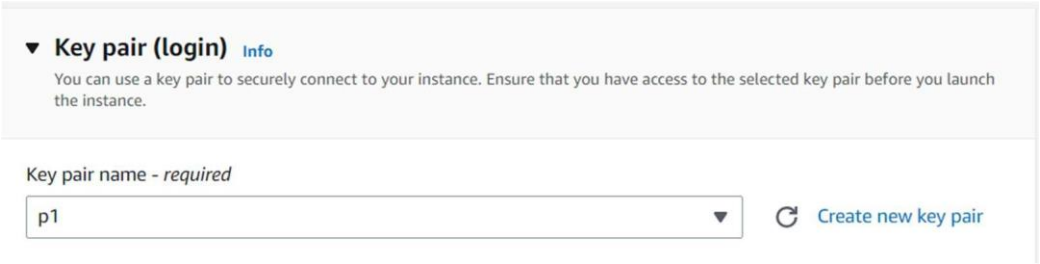
- Launching an Public instance
- Launch an instance and give name as public instance



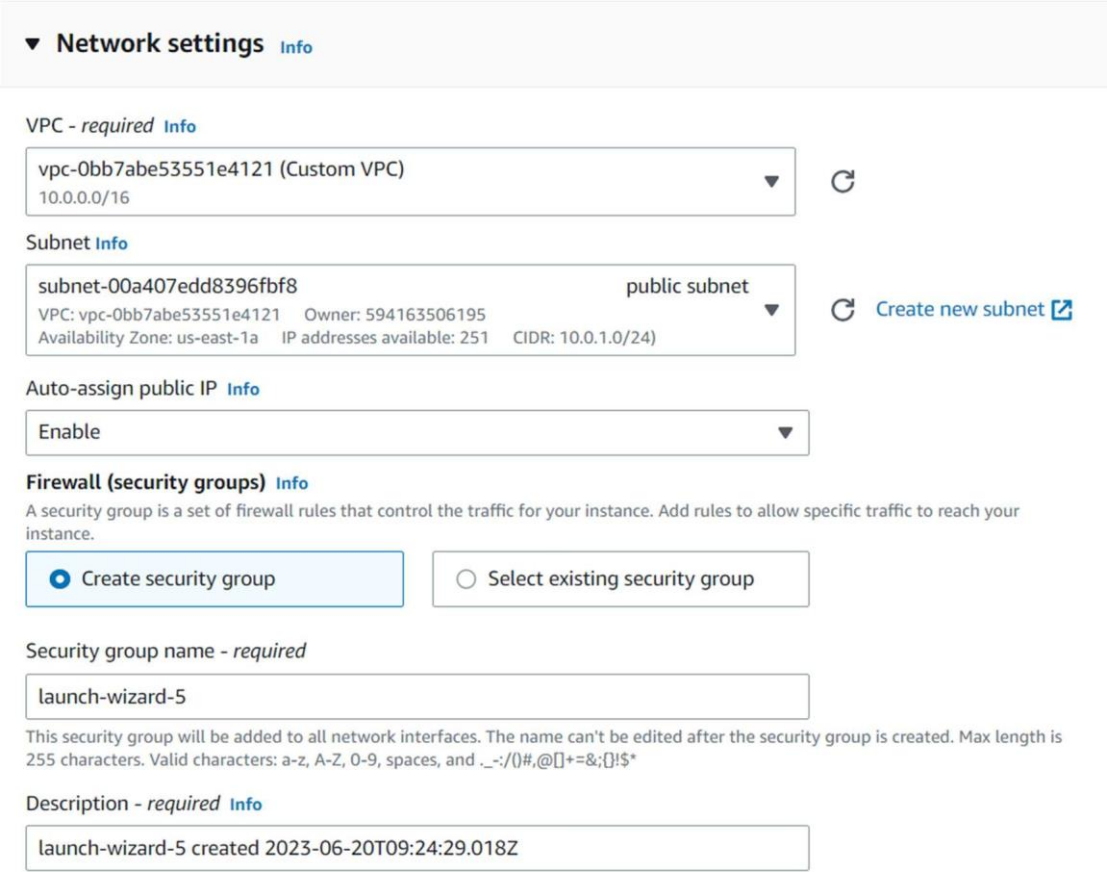
- Give Ubuntu as an AMI



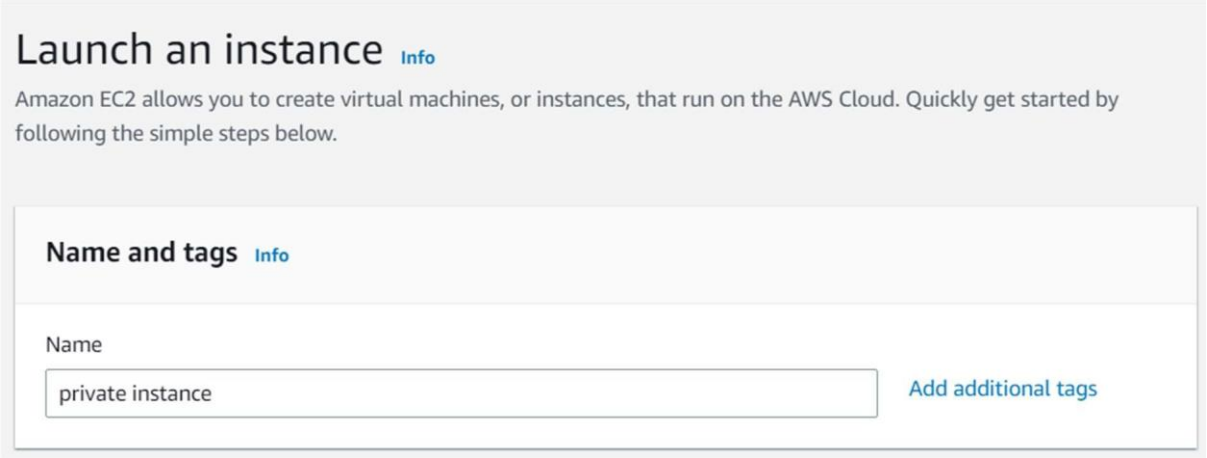
- Give an existing KeyPair(as our wish)



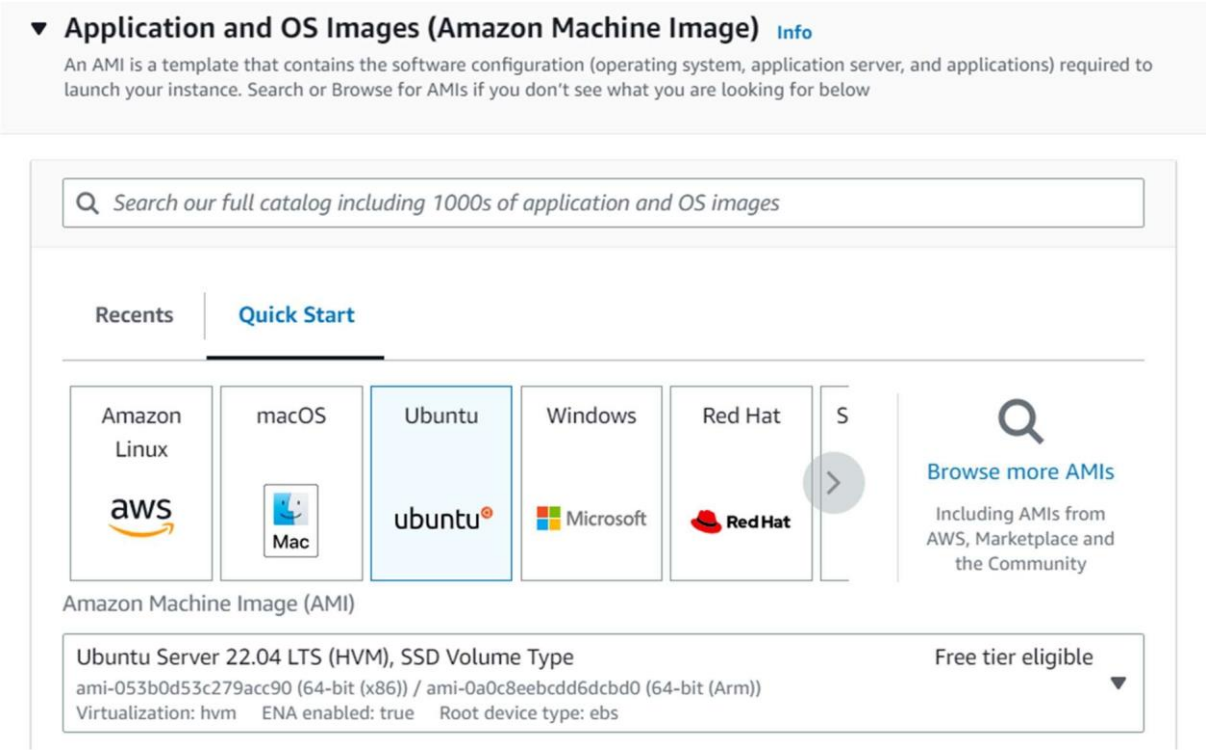
- Network Settings>give created VPC>give public subnet>Launch instance



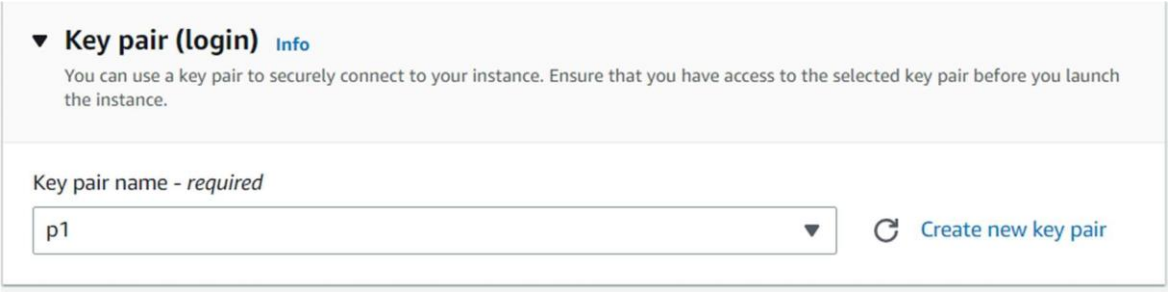
- Lauching private instance
- Launch an instance and name it as private instance



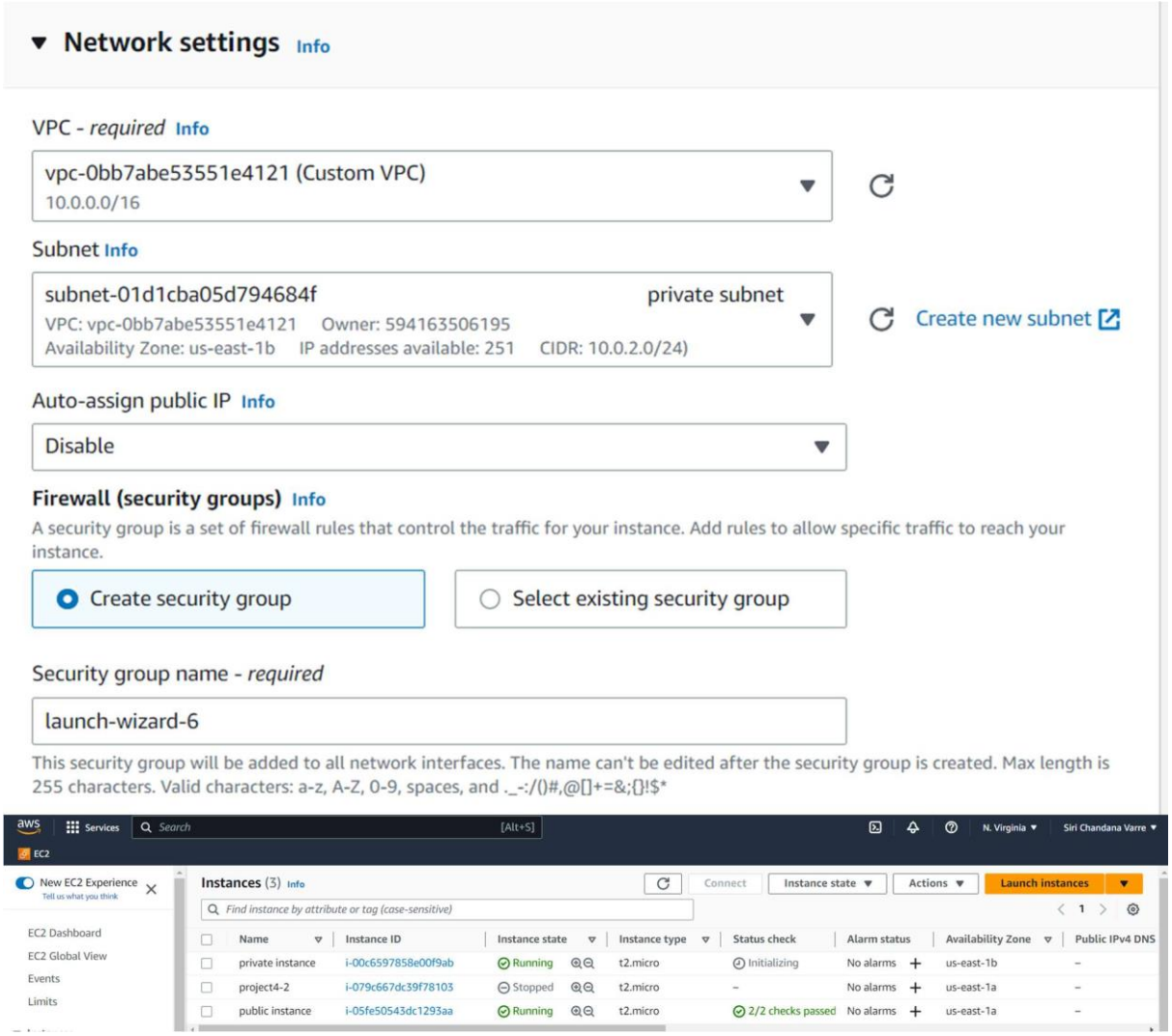
- Give Ubuntu as an AMI



- Give an existing KeyPair(as our wish)



- Network Settings>give created VPC>give private subnet>Launch instance.



Step 11: Open the folder which contains the pem file which we created keypair >right click on mouse>show more options>use Git Bash here>Run the following Commands on the shell.

- chmod 400 [PEM file name]
- ssh -i [PEM file name] ubuntu@[PUBLIC IP]
- sudo apt-get update
- sudo su
- chmod 777 /opt
- exit
- exit
- scp -i public.pem ./private.pem ubuntu@publicip:/opt
- ssh -i public.pem ubuntu@publicip
- cd /opt
- ls
- chmod 400 private.pem

- ssh -i private.pem ubuntu@privateip
- ping google.com (u will not be able to connect to internet as private instance is in the private subnet if you don't create NAT Instance)

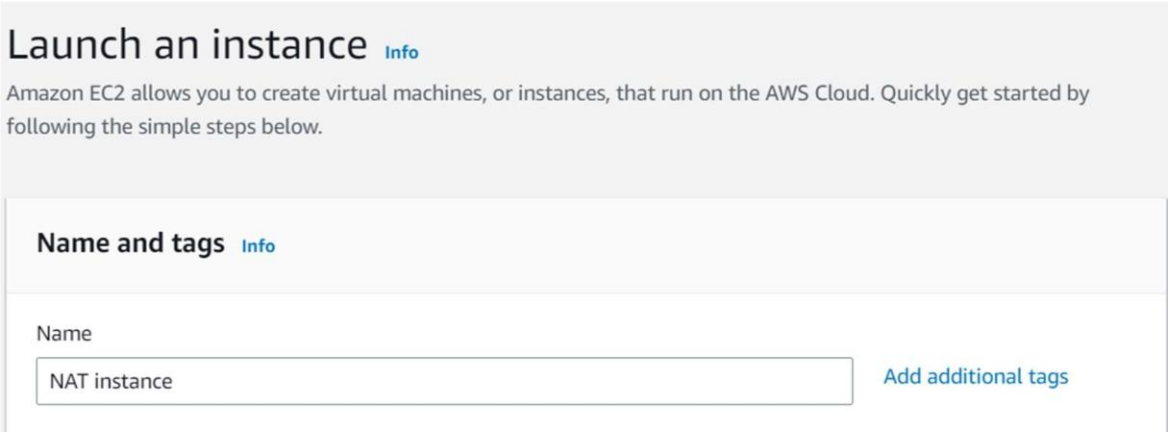
```
Asus@DESKTOP-U8NRV6M MINGW64 ~/OneDrive/Desktop/aws
$ ls
kpubuntu.pem  nfs.pem  p1.pem  p2.pem

Asus@DESKTOP-U8NRV6M MINGW64 ~/OneDrive/Desktop/aws
$ chmod 400 p1.pem

Asus@DESKTOP-U8NRV6M MINGW64 ~/OneDrive/Desktop/aws
$ ssh -i p1.pem ubuntu@3.92.192.118
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue Jun 20 10:00:14 UTC 2023

  System load:  0.0               Processes:             97
  Usage of /:   23.1% of 7.57GB   Users logged in:       0
  Memory usage: 24%               IPv4 address for eth0: 10.0.1.253
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

70 updates can be applied immediately.
47 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Tue Jun 20 09:52:54 2023 from 175.101.94.8
ubuntu@ip-10-0-1-253:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
ubuntu@ip-10-0-1-253:~$ sudo su
root@ip-10-0-1-253:/home/ubuntu# chmod 777 /opt
root@ip-10-0-1-253:/home/ubuntu# exit
exit
ubuntu@ip-10-0-1-253:~$ exit
logout
Connection to 3.92.192.118 closed.
```

```
Asus@DESKTOP-U8NRV6M MINGW64 ~/OneDrive/Desktop/aws
$ scp -i p1.pem ./p2.pem ubuntu@3.92.192.118:/opt
p2.pem                                        100% 1674     7.3KB/s   00:00

Asus@DESKTOP-U8NRV6M MINGW64 ~/OneDrive/Desktop/aws
$ ssh -i p1.pem ubuntu@3.92.192.118
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue Jun 20 10:02:55 UTC 2023

  System load:  0.0               Processes:             98
  Usage of /:   23.1% of 7.57GB   Users logged in:       0
  Memory usage: 24%               IPv4 address for eth0: 10.0.1.253
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

70 updates can be applied immediately.
47 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Tue Jun 20 10:00:15 2023 from 175.101.94.8
ubuntu@ip-10-0-1-253:~$ cd /opt
ubuntu@ip-10-0-1-253:/opt$ ls
p2.pem
```

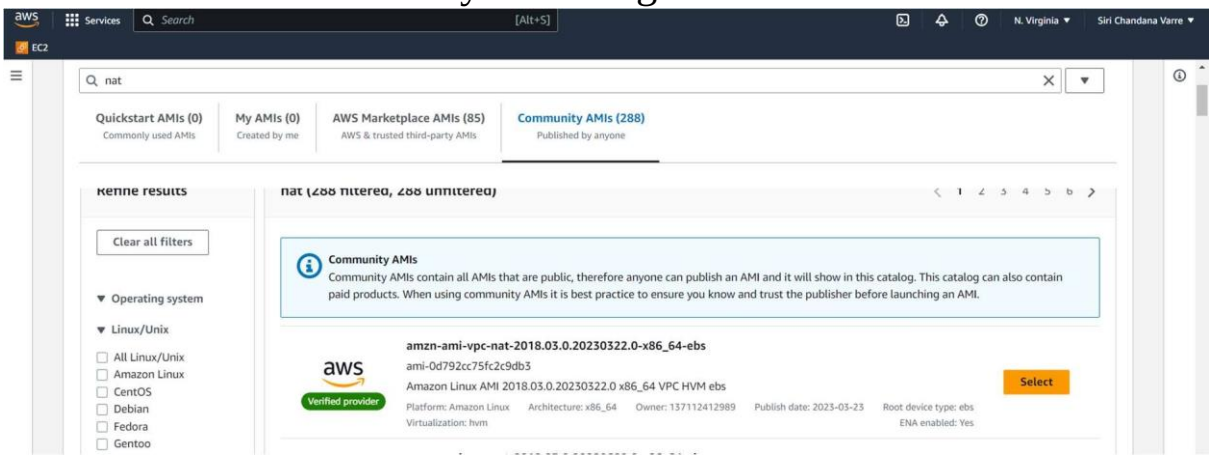
```
Last login: Tue Jun 20 10:00:15 2023 from 175.101.94.8
ubuntu@ip-10-0-1-253:~$ cd /opt
ubuntu@ip-10-0-1-253:/opt$ ls
p2.pem
ubuntu@ip-10-0-1-253:/opt$ chmod 400 p2.pem
ubuntu@ip-10-0-1-253:/opt$ ssh -i p2.pem ubuntu@10.0.2.54
The authenticity of host '10.0.2.54 (10.0.2.54)' can't be established.
ED25519 key fingerprint is SHA256:lqQ1+QKeTN/ndsPOS1nhoQitBBt4f7hGrqRga29OmHA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.54' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue Jun 20 10:04:30 UTC 2023

  System load:  0.080078125       Processes:             94
  Usage of /:   20.6% of 7.57GB   Users logged in:       0
  Memory usage: 23%               IPv4 address for eth0: 10.0.2.54
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-2-54:~$ ping google.com
PING google.com (172.253.63.101) 56(84) bytes of data.
```

- inorder to provide internet access to private instance ,we create NAT(Network Address Translator)

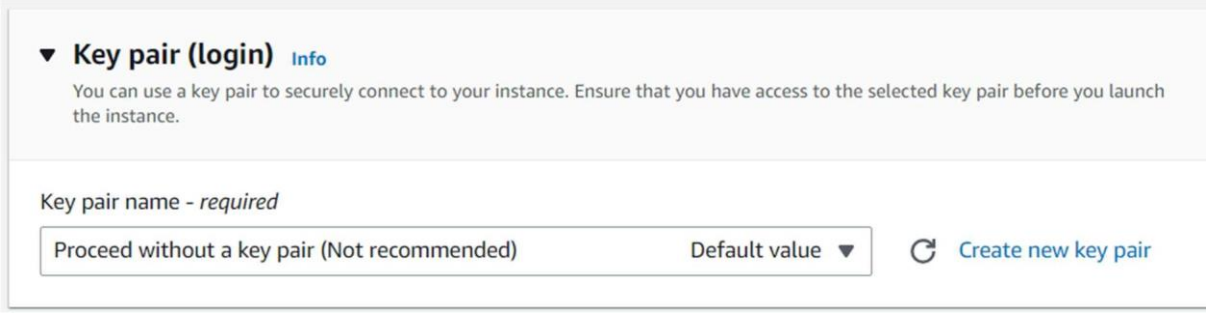
Step 12: Launch instance(NAT instance)



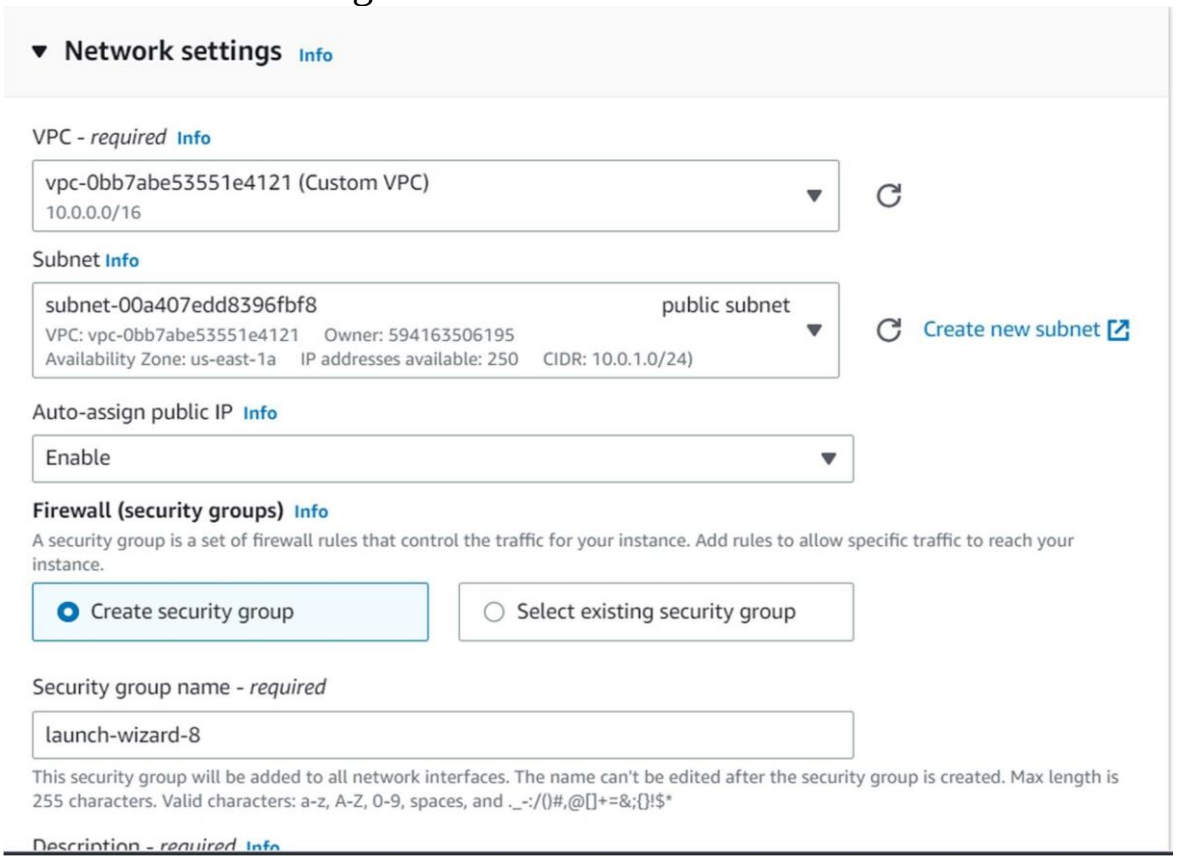
- Select aws-NAT AMI by Browsing AMI's



- Key pair-proceed without keypair

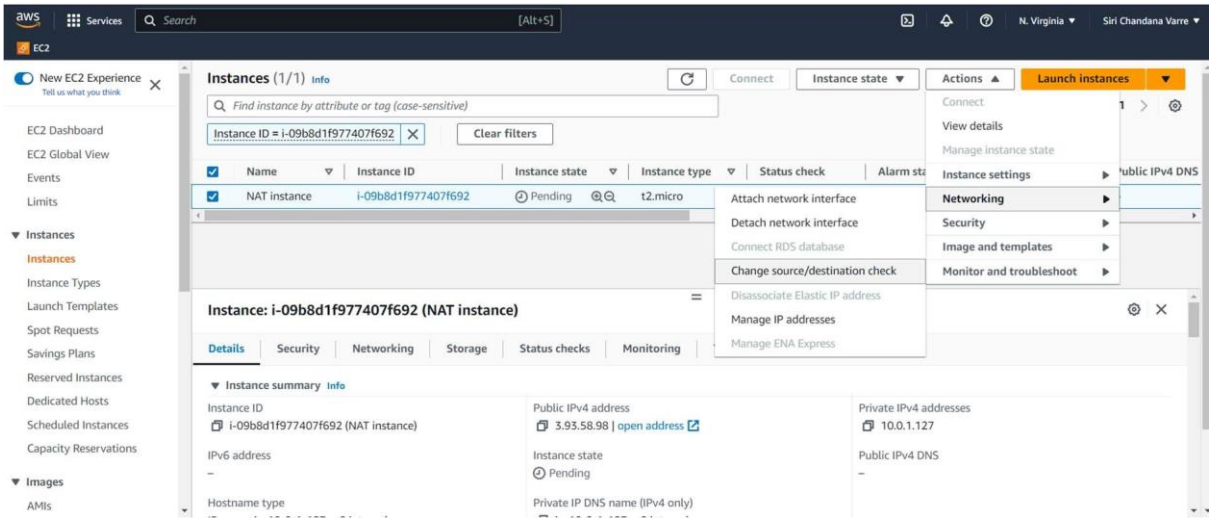


- As we creating NAT in Public Subnet ,select public subnet in Network settings

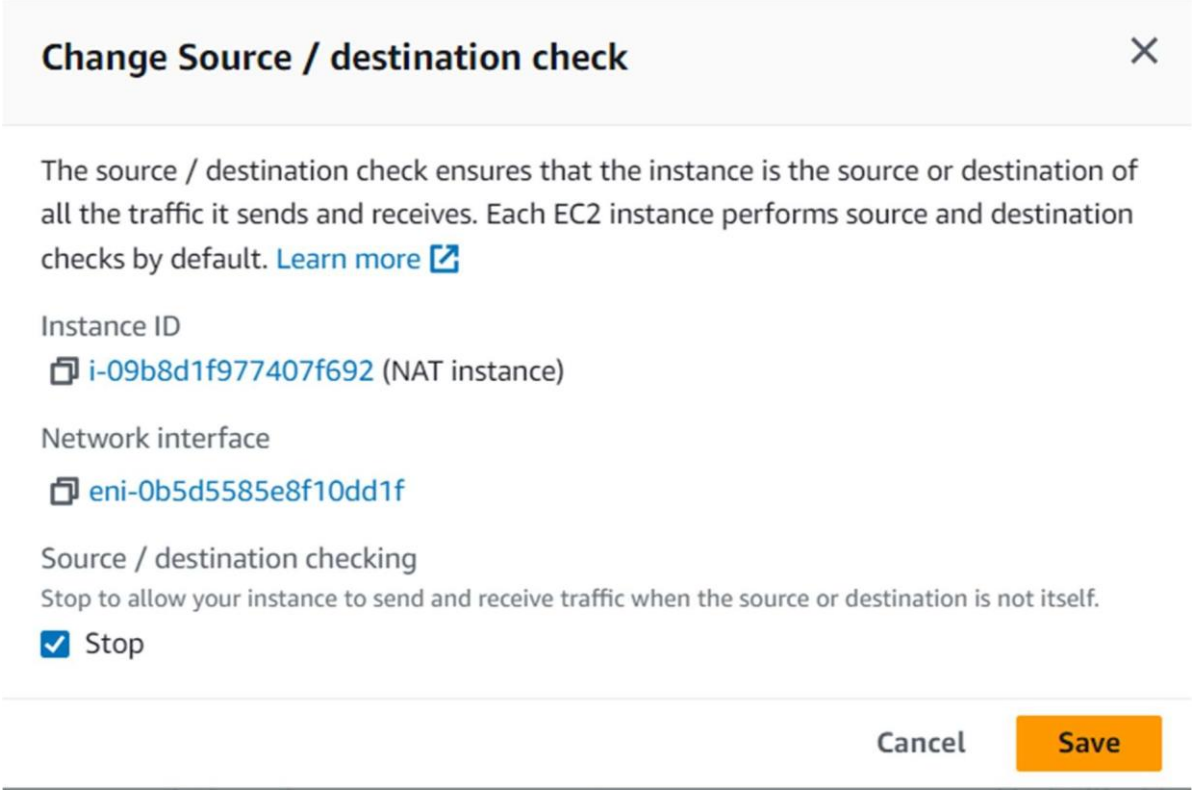


- Inbound security rules>Type-All Traffic>Launch Instance

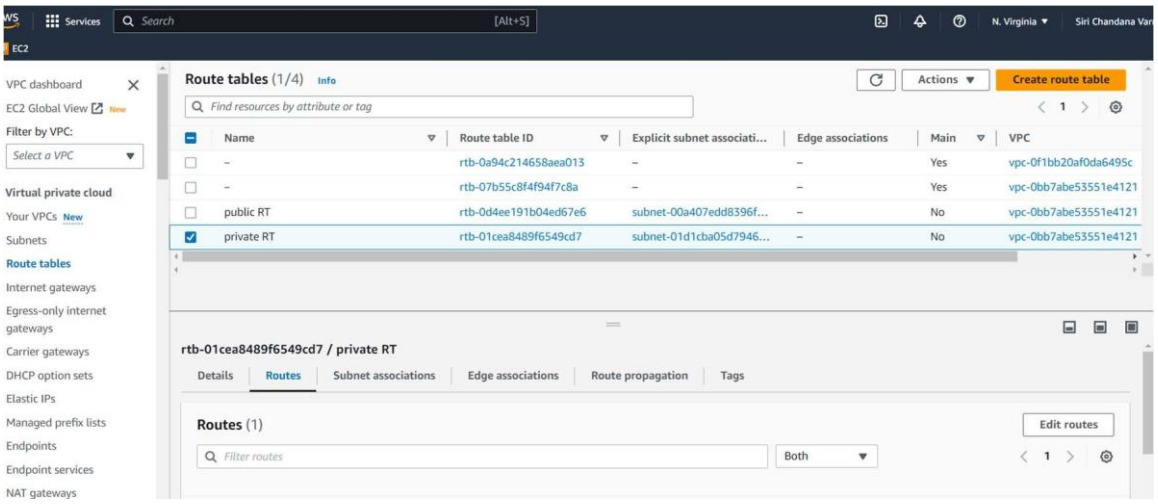
Step 13: Select NAT instance>Actions>Networking>Change source/destination check



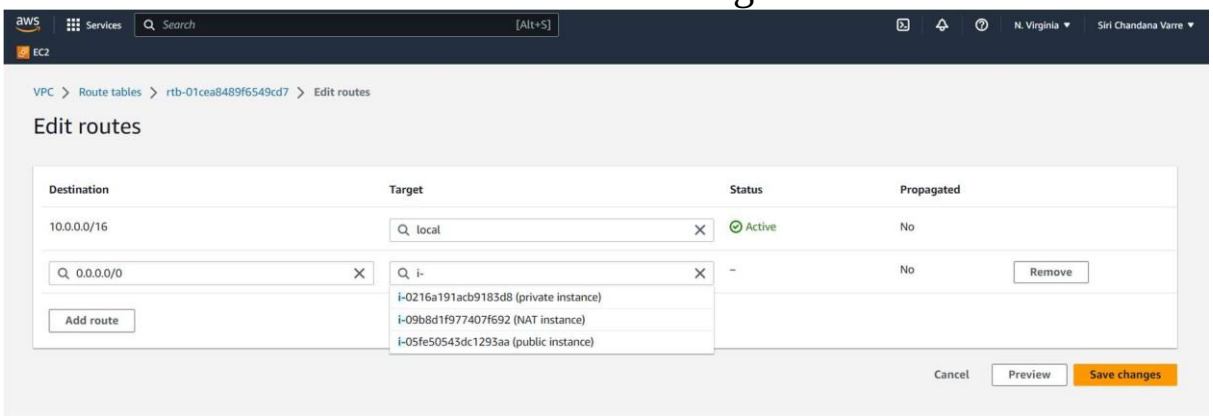
- Click on stop and save



Step 14: Go to Route Tables in VPC Service>Select private RT>Routes>Edit routes



- Edit routes>ADD route>destination-0.0.0.0/0,Target-select NAT instance from instance>Save changes



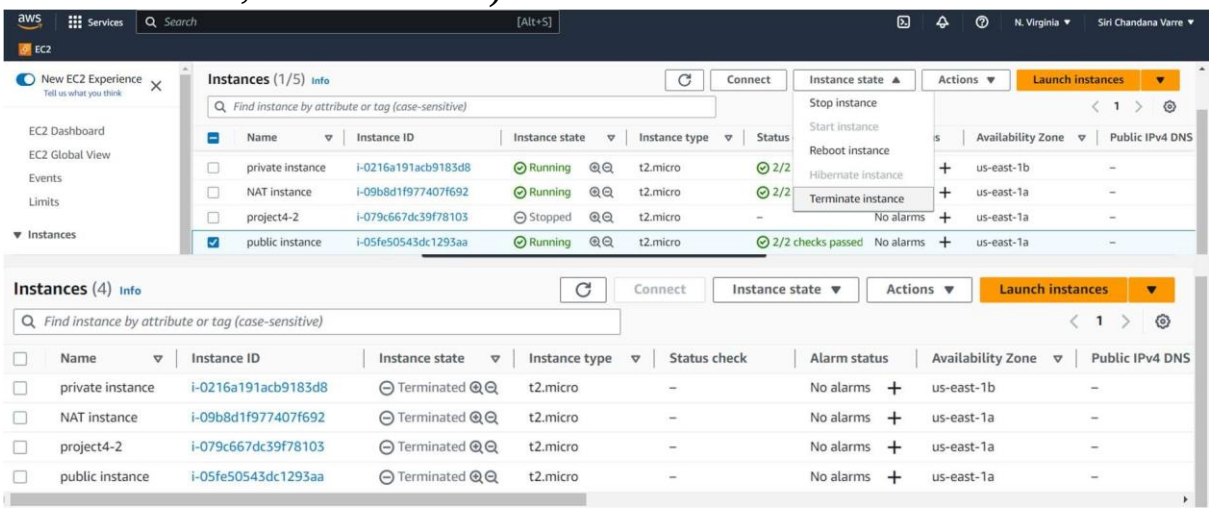
- Now go to Git Bash and run the Command (ping google.com)

```
--- google.com ping statistics ---
1686 packets transmitted, 34 received, 97.9834% packet loss, time 1724639ms
rtt min/avg/max/mdev = 1.898/2.223/4.788/0.575 ms
ubuntu@ip-10-0-2-54:~$ ping google.com
PING google.com (172.253.115.113) 56(84) bytes of data.
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=1 ttl=48 time=2.67 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=2 ttl=48 time=2.62 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=3 ttl=48 time=2.70 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=4 ttl=48 time=2.67 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=5 ttl=48 time=2.65 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=6 ttl=48 time=2.62 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=7 ttl=48 time=2.61 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=8 ttl=48 time=2.65 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=9 ttl=48 time=2.64 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=10 ttl=48 time=2.66 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=11 ttl=48 time=2.69 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=12 ttl=48 time=2.65 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=13 ttl=48 time=2.64 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=14 ttl=48 time=2.67 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=15 ttl=48 time=2.59 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=16 ttl=48 time=2.87 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=17 ttl=48 time=2.69 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=18 ttl=48 time=2.68 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=19 ttl=48 time=2.65 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=20 ttl=48 time=2.71 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=21 ttl=48 time=2.64 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=22 ttl=48 time=2.68 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=23 ttl=48 time=2.63 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=24 ttl=48 time=2.62 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=25 ttl=48 time=3.75 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=26 ttl=48 time=2.62 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=27 ttl=48 time=2.69 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=28 ttl=48 time=3.09 ms
64 bytes from bg-in-f113.1e100.net (172.253.115.113): icmp_seq=29 ttl=48 time=3.12 ms
```

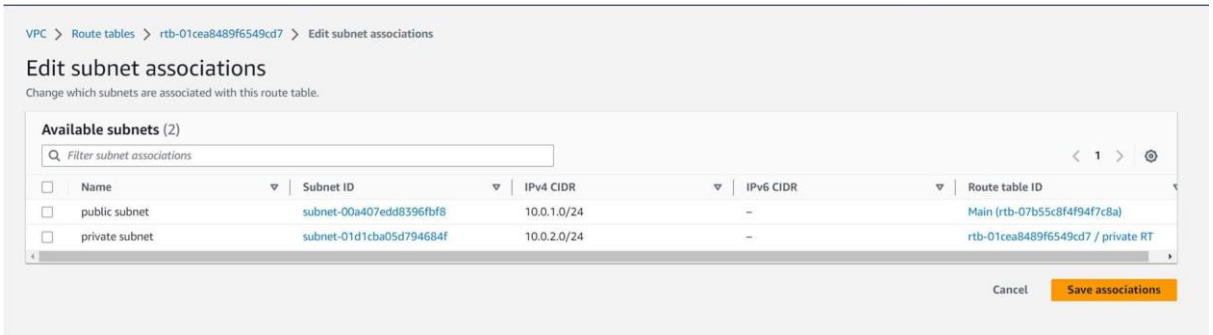
Now,private instance an access to internet Through NAT

Step 16: CleanUp

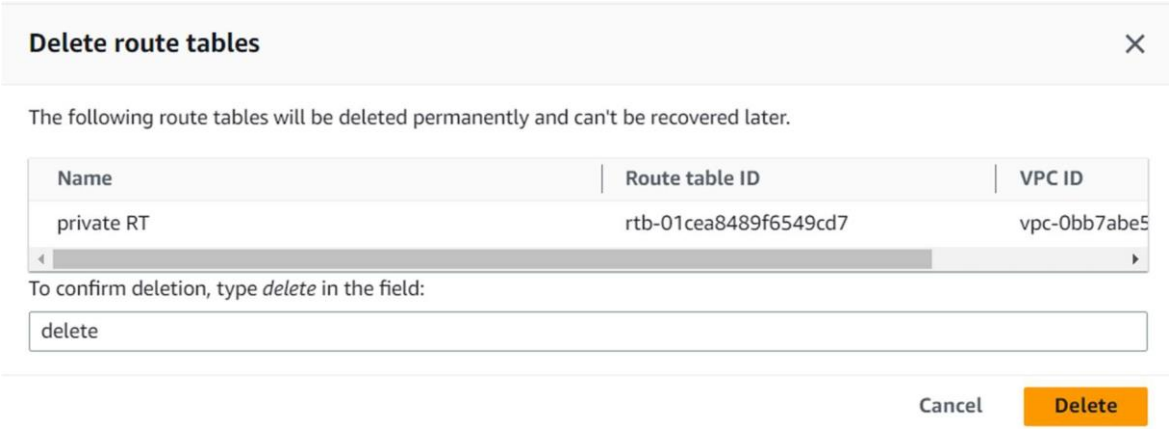
- Deleting Instances
  - Go to EC2 Service>instances>select>instance state>Terminate instance(terminate private instance,public instance,NAT instance)



- Deleting Route Tables
  - Go to VPC Service
  - Go to Route Tables>subnet assocciations>Edit>deselect the subnet and save associations

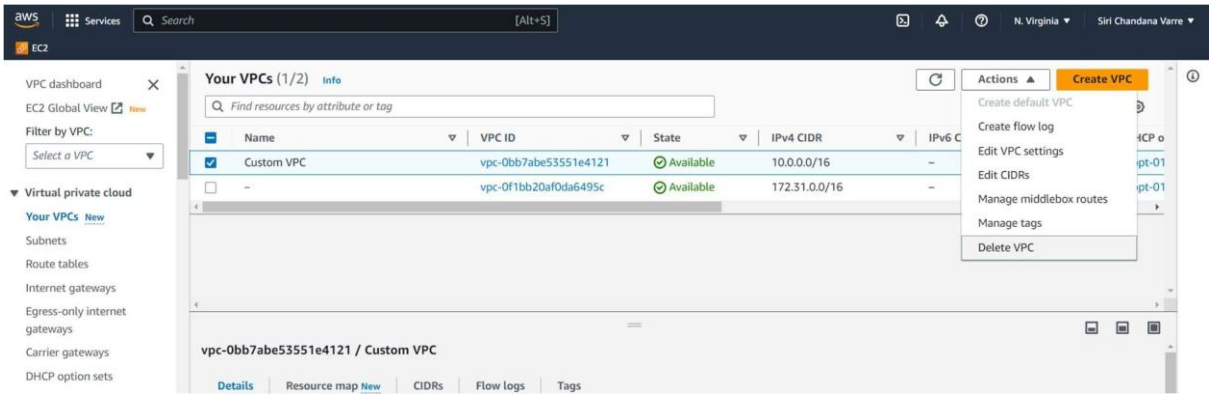


- Now Select Route Table >Actions>delete Route Table>enter text ‘delete’>Delete

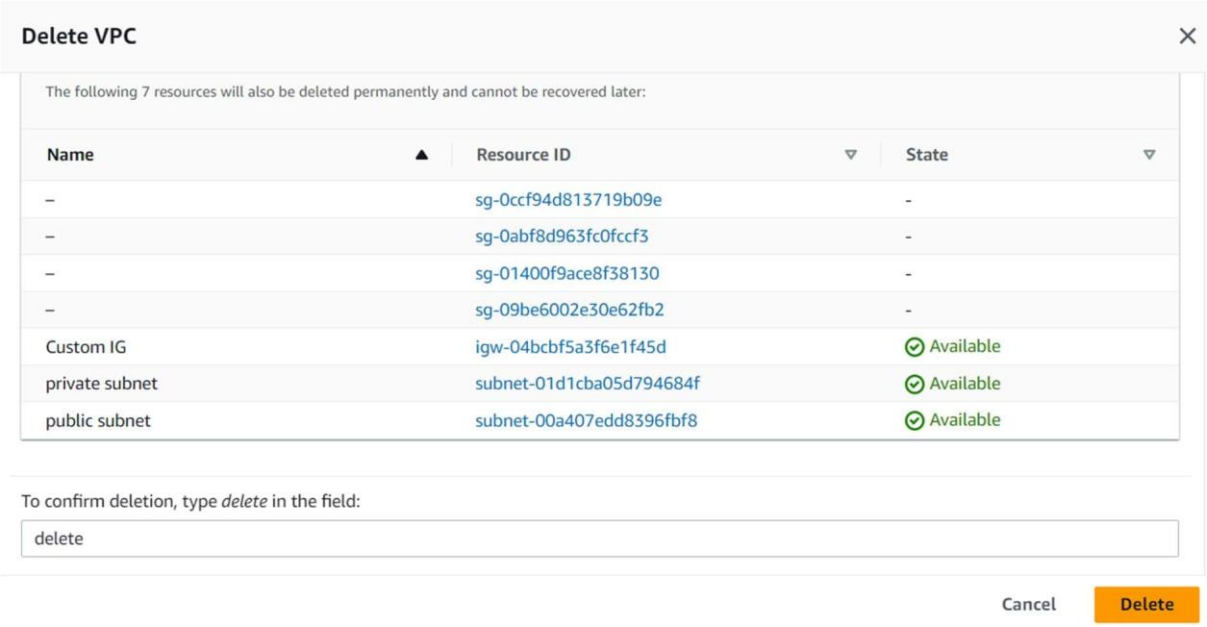


➢ Deleting VPC

- Go to your vpc's >select VPC to delete>Actions>Delete VPC



- Type text 'delete',then VPC will be deleted



Conclusion

In conclusion, By implementing the proposed architecture, organizations can separate resources based on connectivity requirements, with the public subnet allowing public access through the IGW and the private subnet enabling outbound internet connectivity via the NAT gateway or instance. Routing and security measures ensure controlled traffic flow and enforce security policies, enhancing data privacy and mitigating external threats.

By following this architecture, organizations can confidently deploy their applications in the cloud, achieving a well-designed VPC setup that meets their connectivity and security needs while leveraging the advantages of cloud infrastructure.