Exploring AWS Cloud Services

What are Aws?

AWS (Amazon Web Services) is a comprehensive and widely used cloud computing platform provided by Amazon. It offers a vast array of cloud services, including computing power, storage solutions, databases, networking, machine learning, analytics, security, and more. AWS allows businesses and individuals to access computing resources on-demand, without the need for upfront investments in physical hardware. It provides scalability, flexibility, and cost-effectiveness, making it a popular choice for hosting applications, websites, and managing various IT infrastructure needs.

What is Cloud Computing?

Cloud computing refers to the delivery of computing services over the internet. Instead of owning and maintaining physical servers and infrastructure, cloud computing allows users to access resources such as storage, processing power, and applications on-demand from cloud service providers. This model offers scalability, flexibility, cost-effectiveness, and accessibility, enabling businesses and individuals to focus on their core activities without the burden of managing complex IT infrastructure.

Explain the benefits of using AWS, such as scalability, reliability, cost-effectiveness, and global reach.

Scalability: AWS allows you to easily scale your resources up or down based on your needs. If your website or application suddenly gets a lot of traffic, AWS can handle the increased demand without any issues.

Reliability: AWS offers high reliability and uptime for your services. They have multiple data canters around the world, so even if one goes down, your data and services remain accessible.

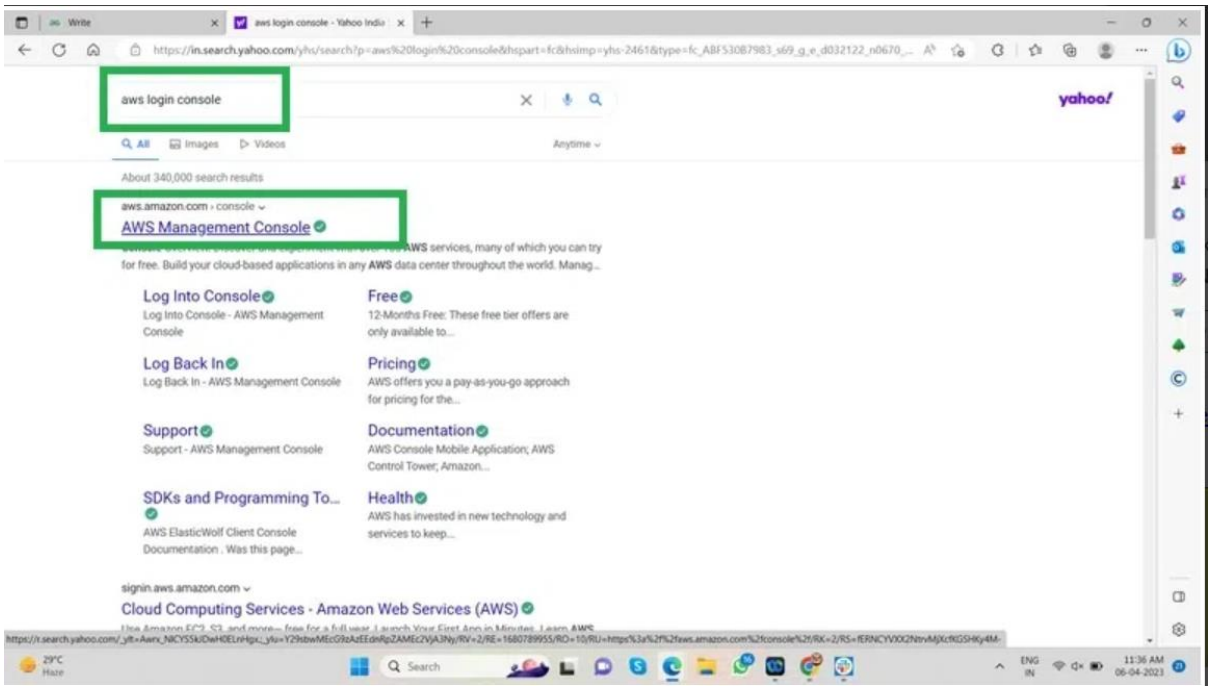
Cost-effectiveness: With AWS, you only pay for what you use. There are no upfront costs, and you can adjust your resources as needed, which helps you save money compared to maintaining your own hardware.

Global Reach: AWS has a global infrastructure, meaning you can deploy your applications and services closer to your users, reducing latency and improving performance worldwide.

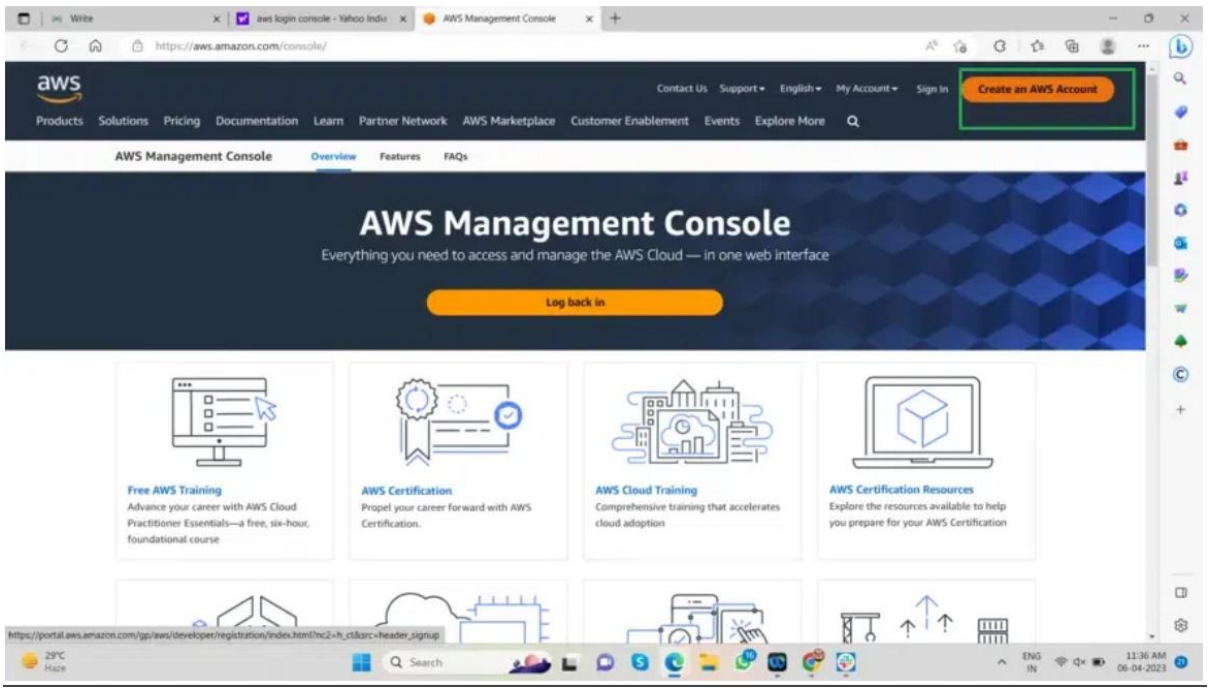
How Creating and Log-in to AWS account.

Creating an AWS Account:

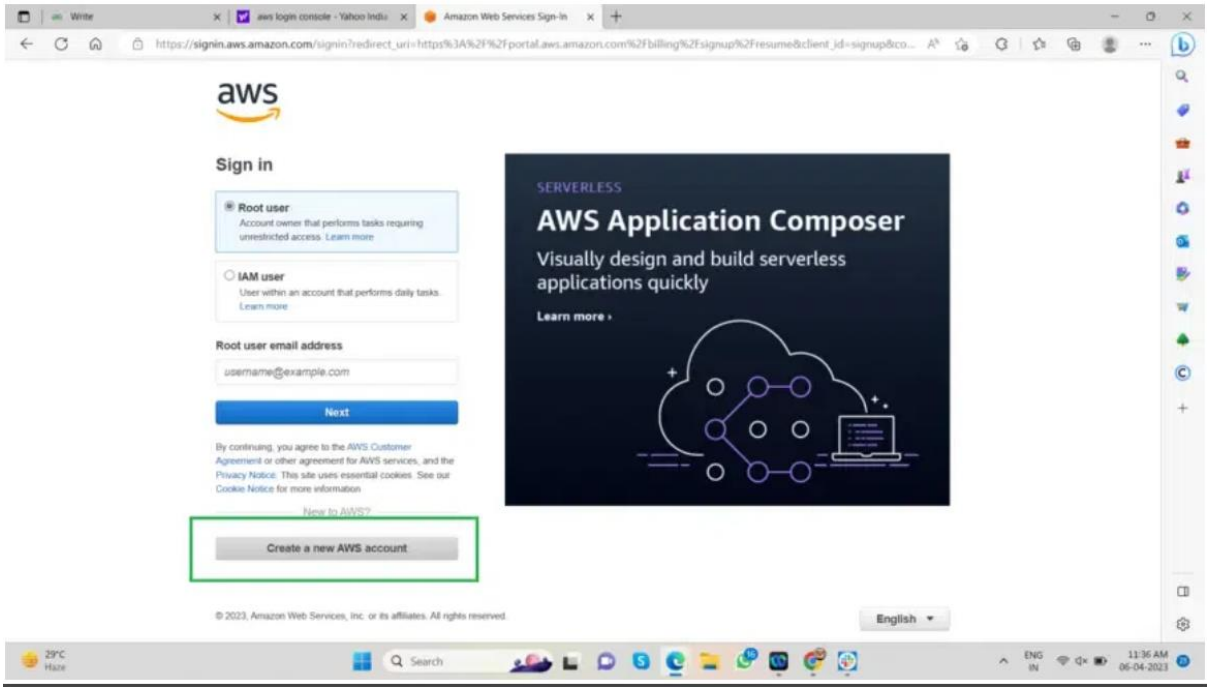
Step 1: First Open your web browser and search for AWS Login Console and click on the first link. As shown in the picture below



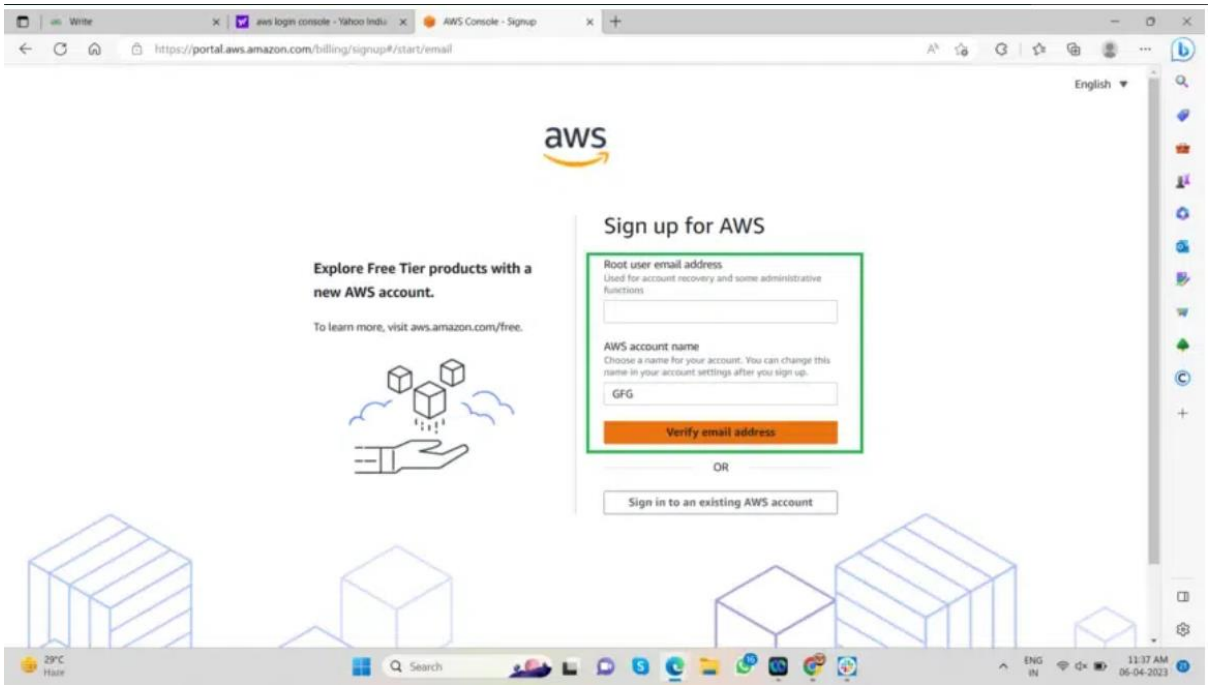
Step 2: An AWS Login Console page will open now click on Create an AWS Account.



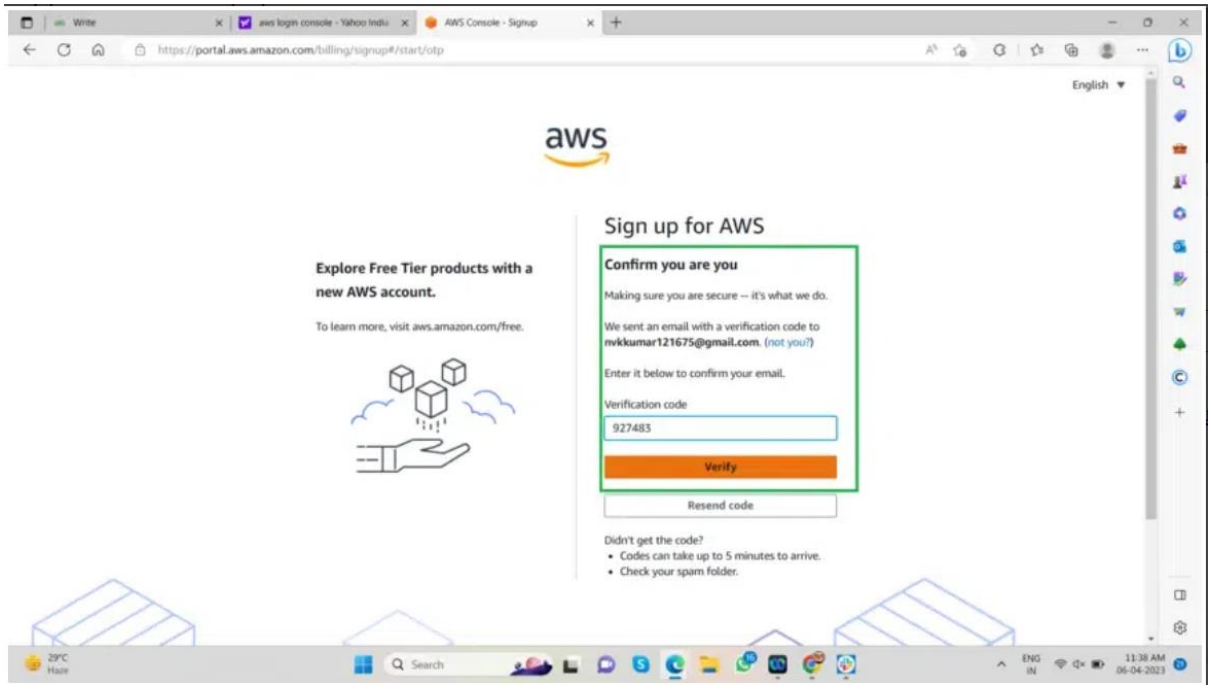
Step 3: A new AWS sign-in page will now open after selecting Create an AWS Account**.** Choose to Create a new AWS account. As shown in the image below



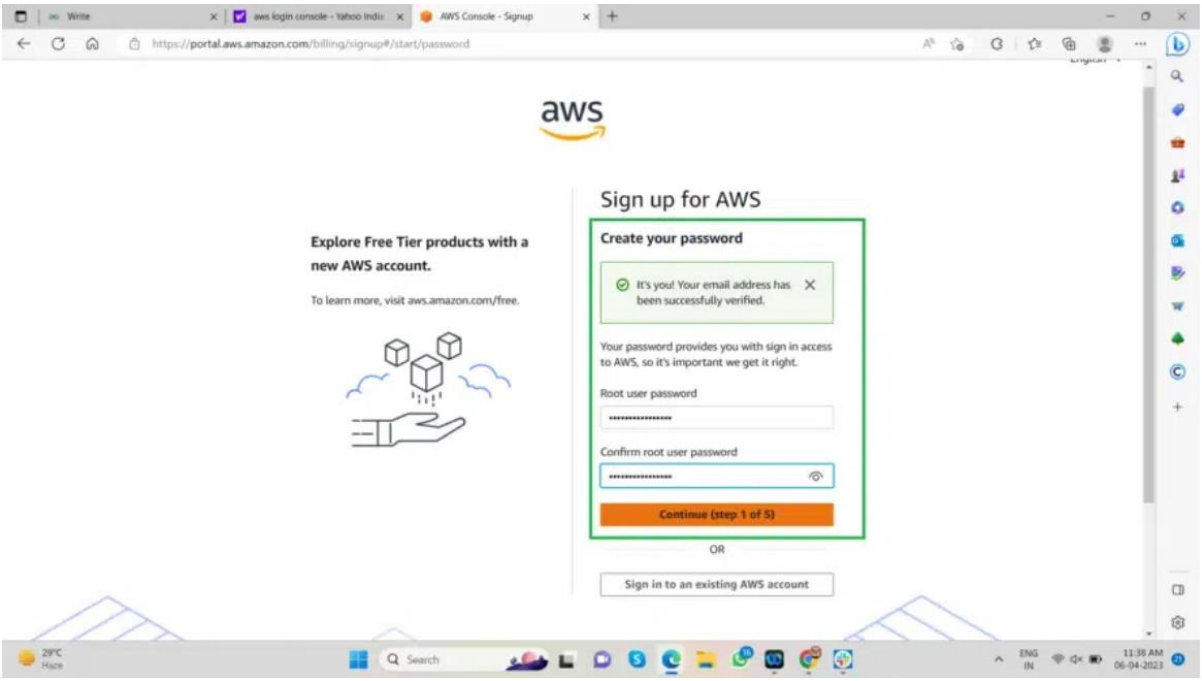
Step 4: In order to use the feature to log into an AWS Free Tire Account**,** we must validate the email address and have to provide the AWS account name in this stage. After clicking on **"**Verify Email Address**,"** you will receive a verification code at the address you provided. Next, you must create a password for this account. Finally, click **"**Continue**"** to move on to the next stage. The pictures below show every step of the process.



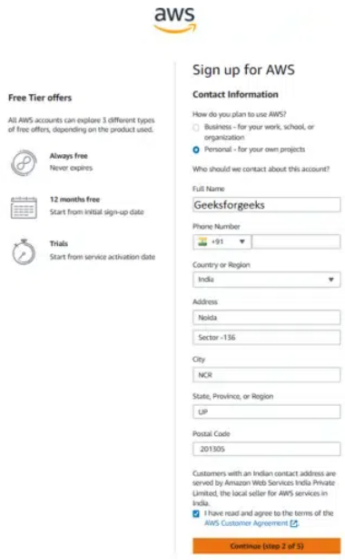
After clicking on **"**Verify Email Address**,"** you will receive a verification code at the address you provided.



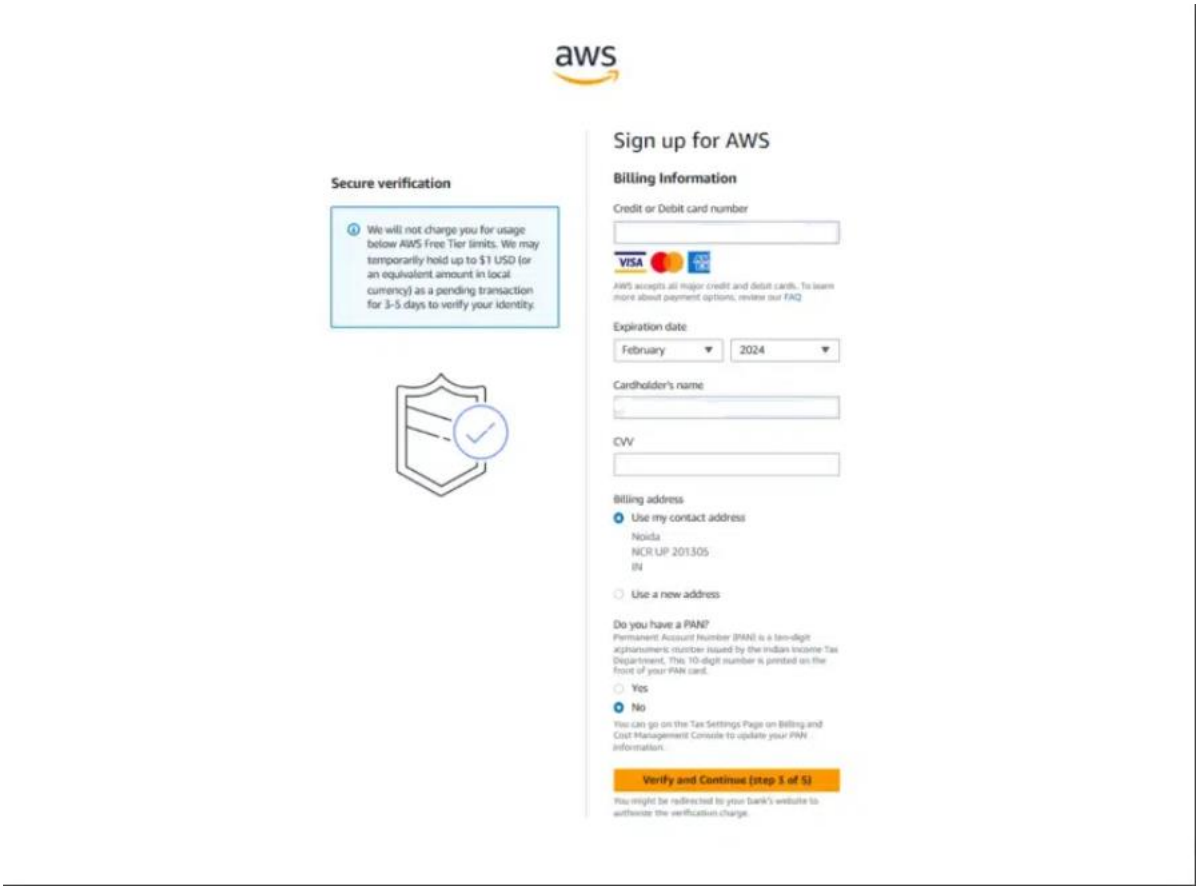
Next, you must create a password for this account. Finally, click **"**Continue**"** to move on to the next stage.



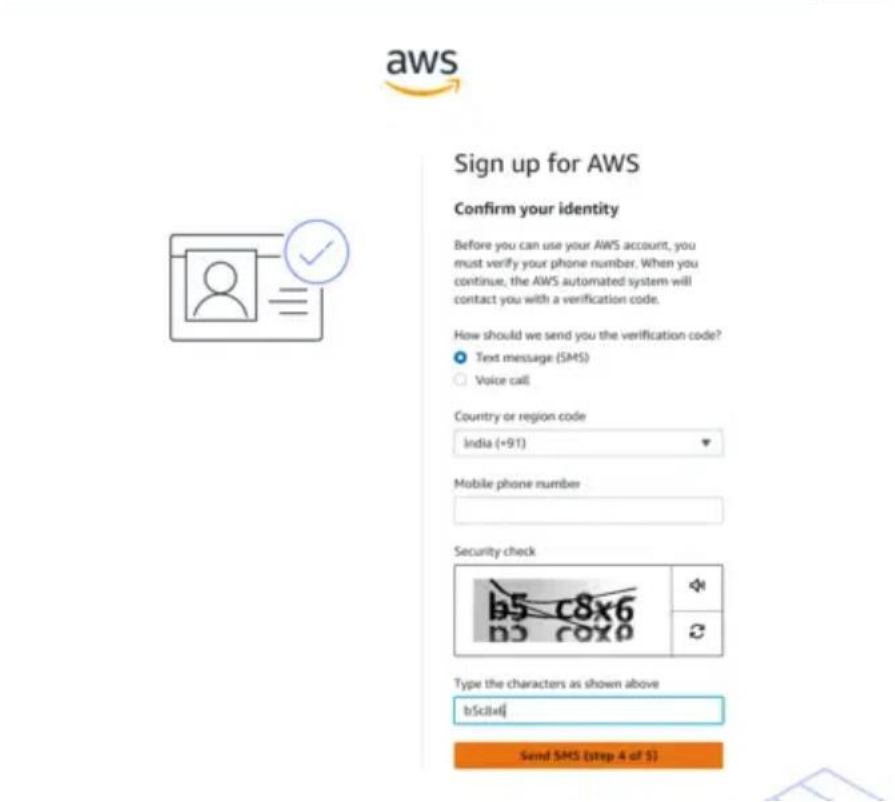
Step 5: We must include all of our contact information in this phase to make it easier for Amazon support personnel to get in touch with us about our AWS Account and any feature references. As shown in the image below.



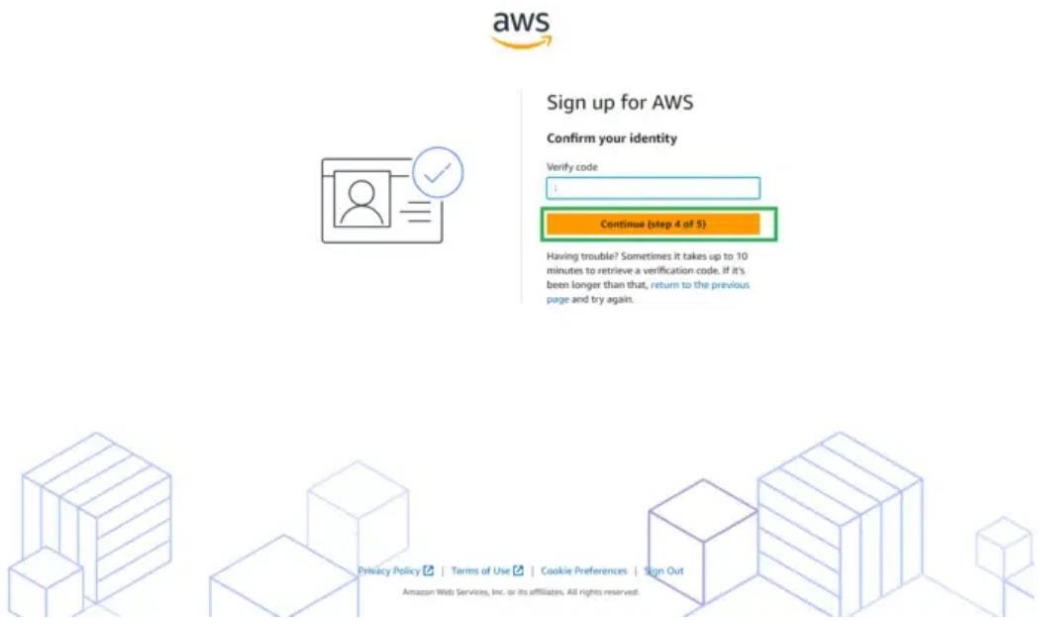
Step 6: We must provide the credit/debit card information in this step. There is no reason to panic at this time. AWS won't deduct any amount unless you pay it on your own. AWS may temporarily keep your identification that they will charge you only 2 Indian rupees.



Step 7: We have to verify our phone number in this phase. As seen in the image below, select “TEXT or Voice call**”** as the method for receiving your verification number, then complete the captcha by clicking on “Send SMS.” You will be sent to a screen where you must confirm the verification code you have received and click continue to proceed to the following stage. As seen in the pictures below.



Step 8: Enter the verification code you received on your mobile device, validate it, and then click Continue to move on to the following stage.

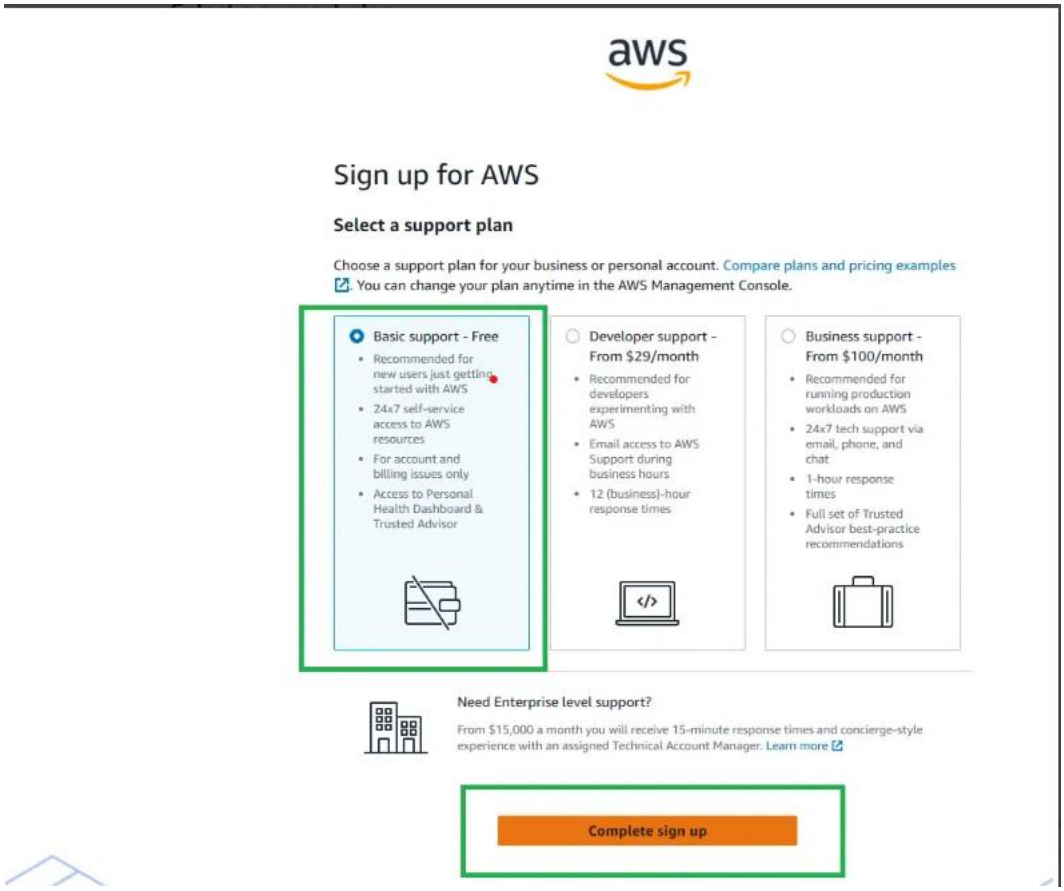


Step 9: Choose the support strategy you want to use. We are setting up an AWS Free Tier Account so select the Basic Support option, which is cost-free and which AWS also suggests for new customers. The Basic Support Plan includes following

- 24*7 self-service access to AWS resources
- Can access personal health dashboard
- It is free of cost

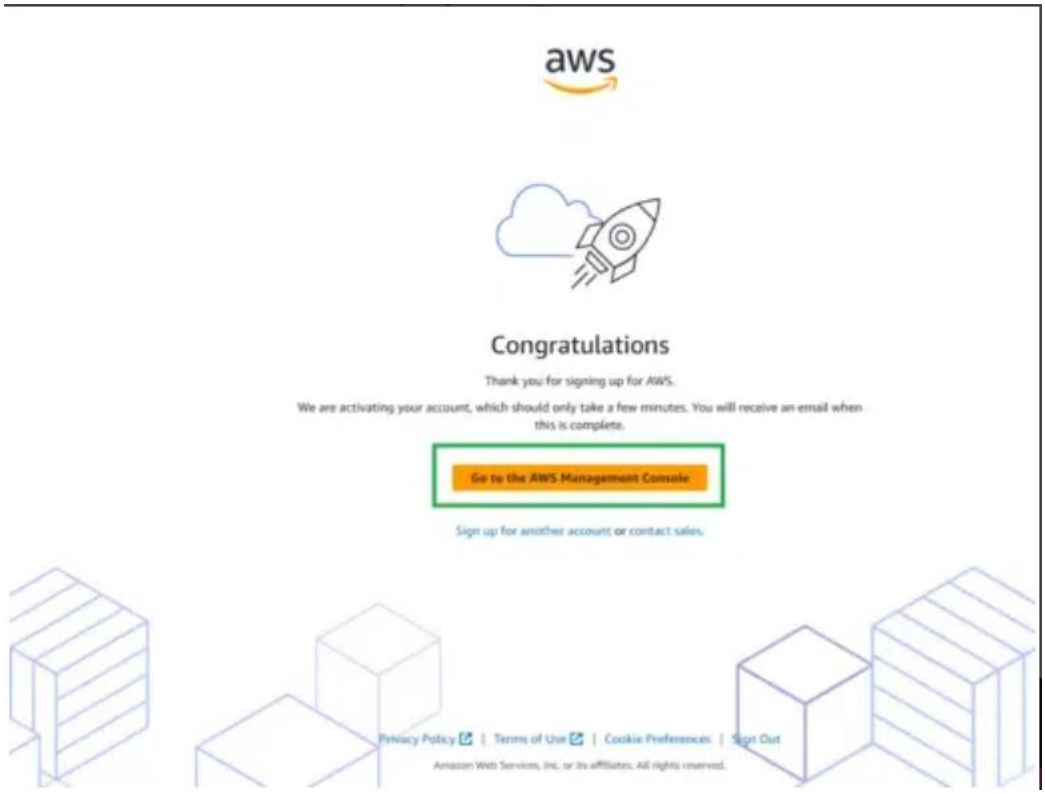
Step 9: Choose the support strategy you want to use. We are setting up an AWS Free Tier Account so select the Basic Support option, which is cost-free and which AWS also suggests for new customers. The Basic Support Plan includes following

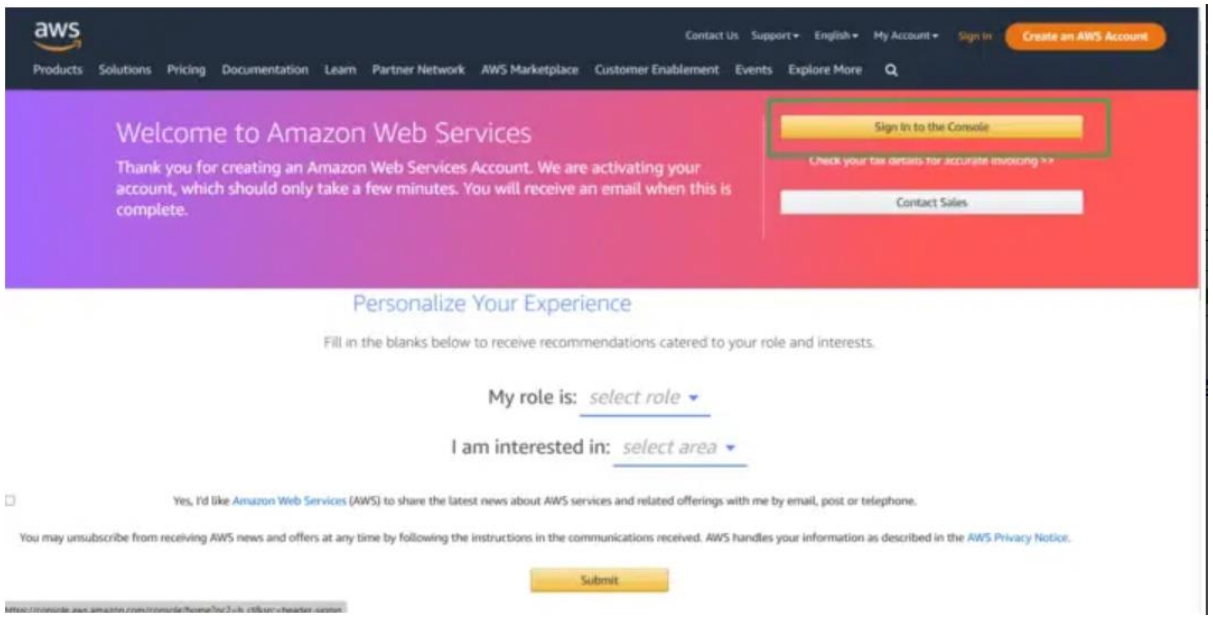
- 24*7 self-service access to AWS resources
- Can access personal health dashboard
- It is free of cost



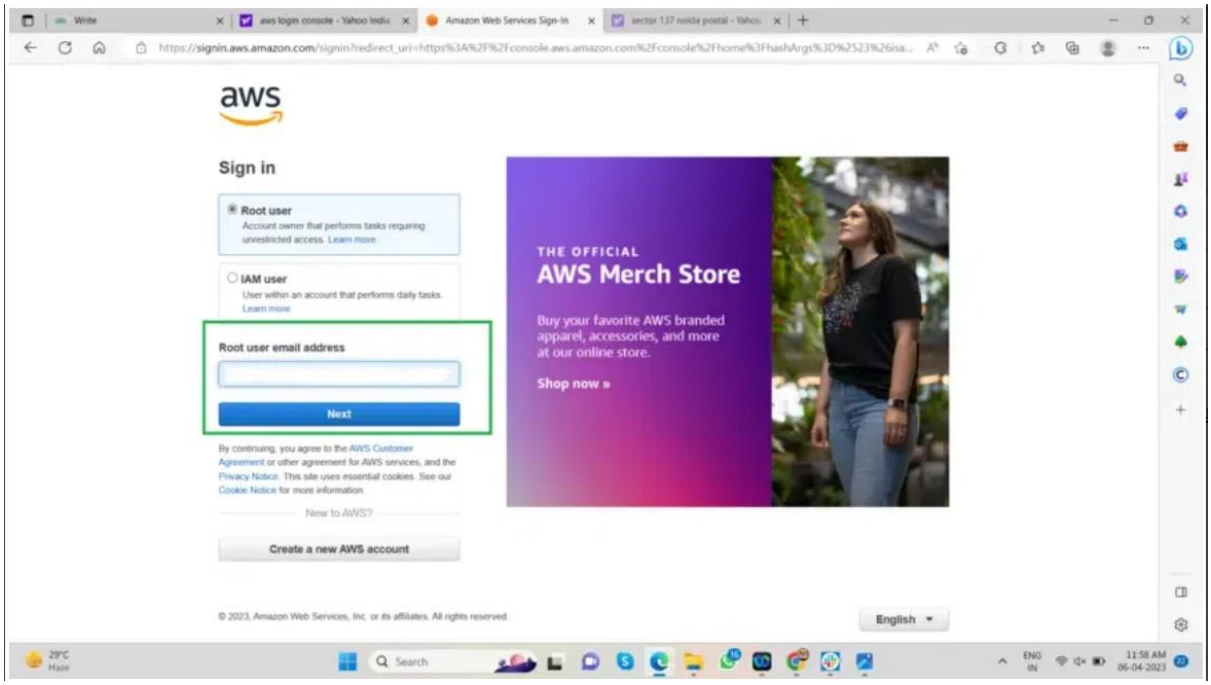
After selecting a plan, click **"**Complete the sign up**"** as shown in the image.

Step 10: "Congratulation" Upon the creation of your AWS account, you can sign in by clicking Click Sign into the console once more, input the email address that you provided, your password, and then click Sign in as shown in the accompanying image, where you can see AWS Management Console's home page for certain of its offering services.

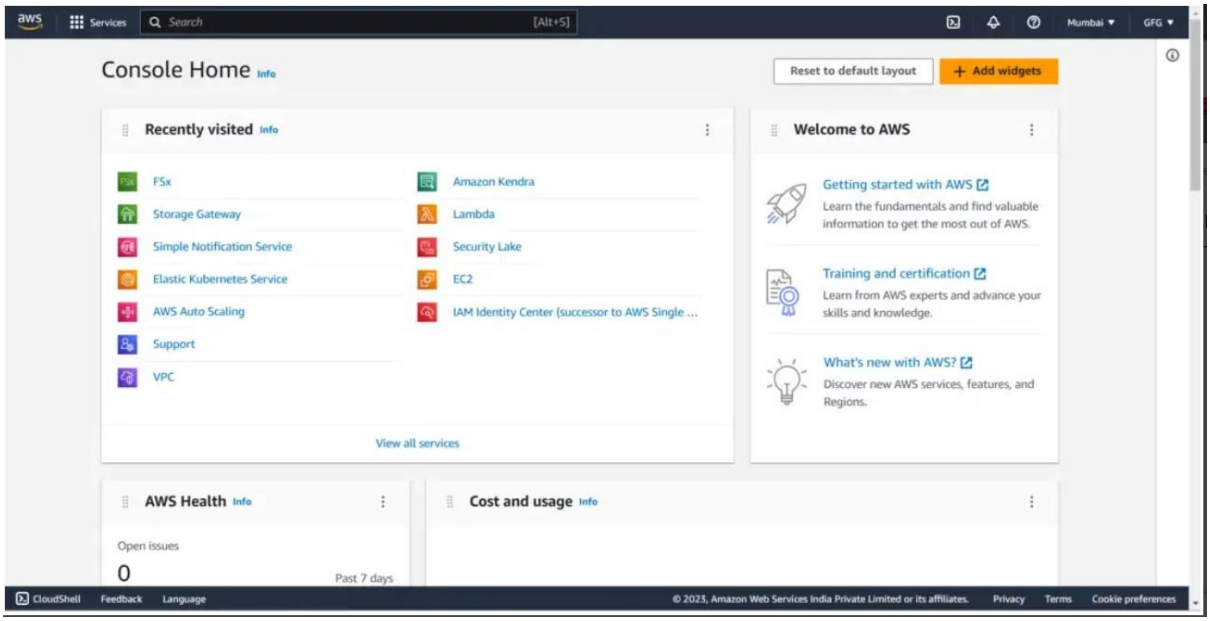




Enter your **email address and previously-configured password**.



And this is the Amazon Console Home page**,** where you may access some of the most popular AWS services, including **EC2, VPC, AUTOSCALING,** etc.



After setting up our AWS Free Tier account, we are now ready to begin using the services that AWS offers.

How to create EC2 instance.

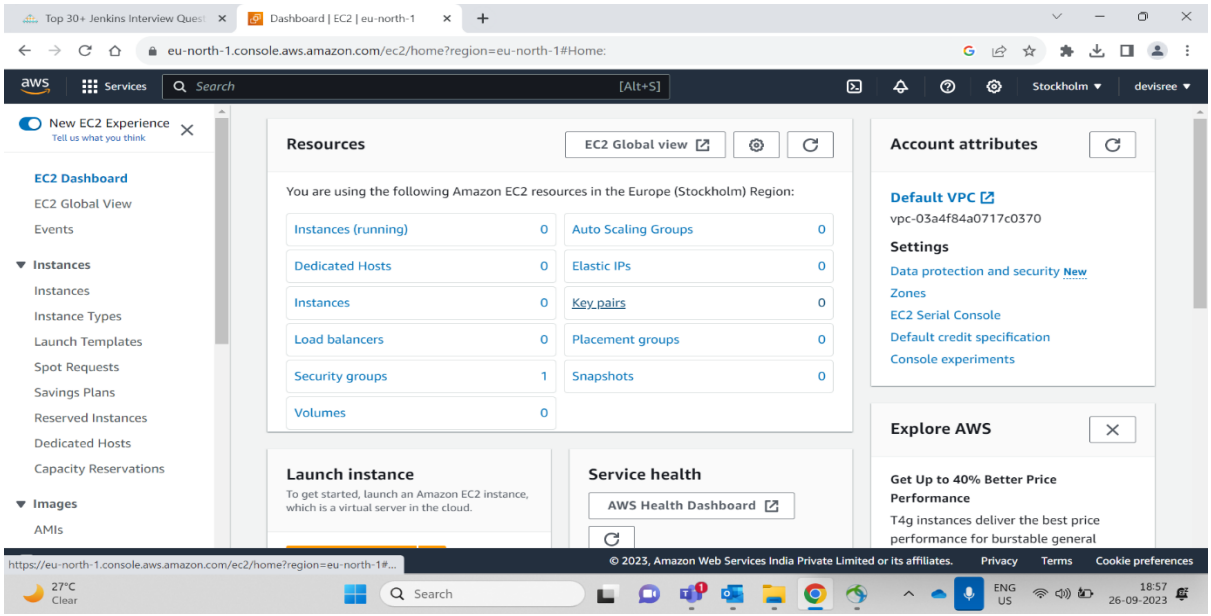
Create Instance in AWS

1) Create the AWS Account

2) Login with your aws account

3) Click on Services

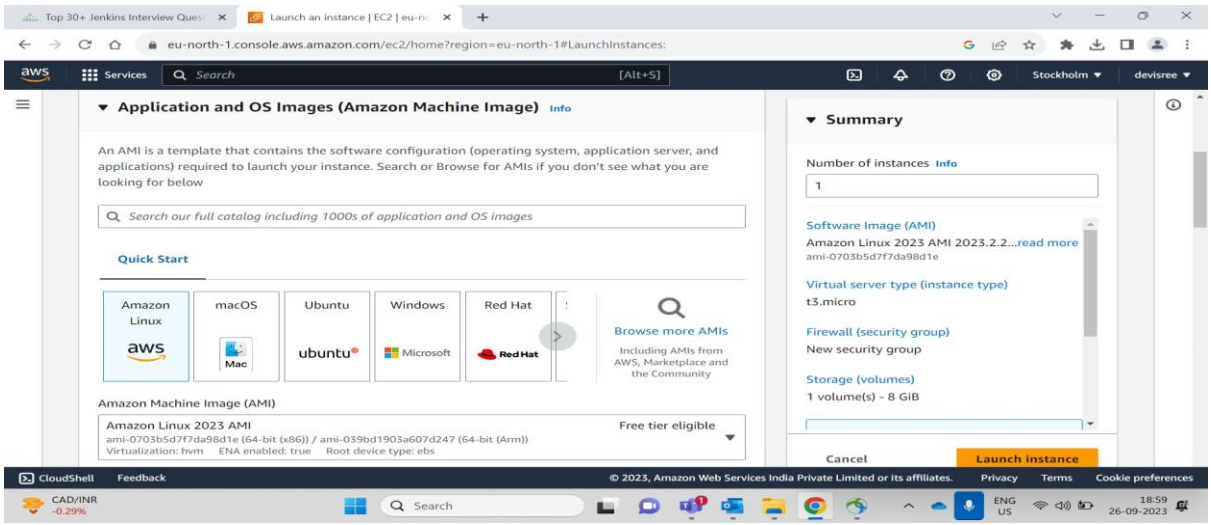
4) Click on EC2



5) Click on Instance

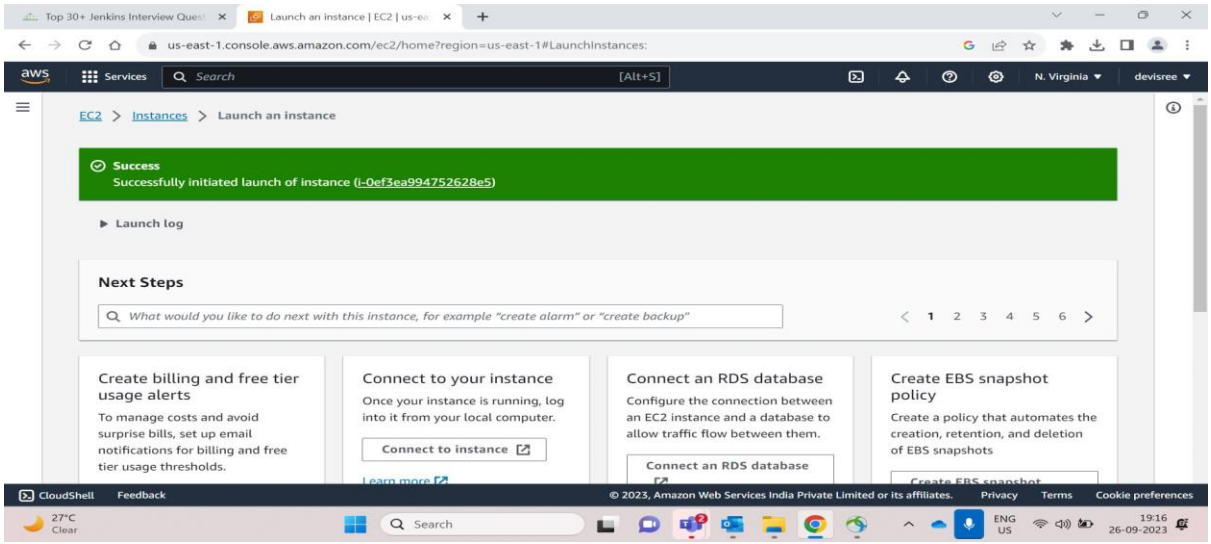
6) Click on launch instance

1. Create an EC2 instance using the AWS Management Console, and configure it according to your requirements, such as selecting the appropriate instance type, region, and networking settings.
2. Give the name of your Instance.
3. Then in Quick Start section choose option Linux Aws



Here we can see the in the instance was created successful.

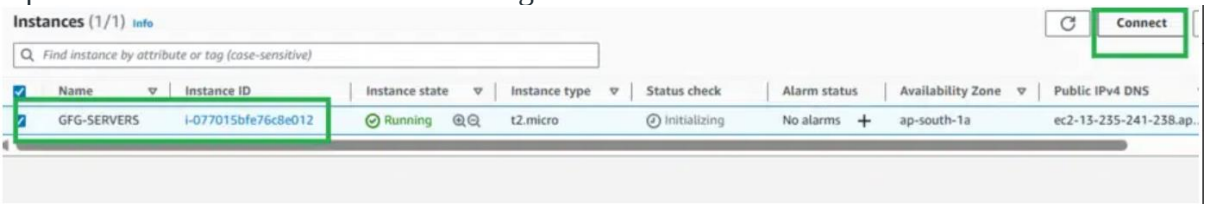
Create a Key- value pair.(A File will be downloaded)



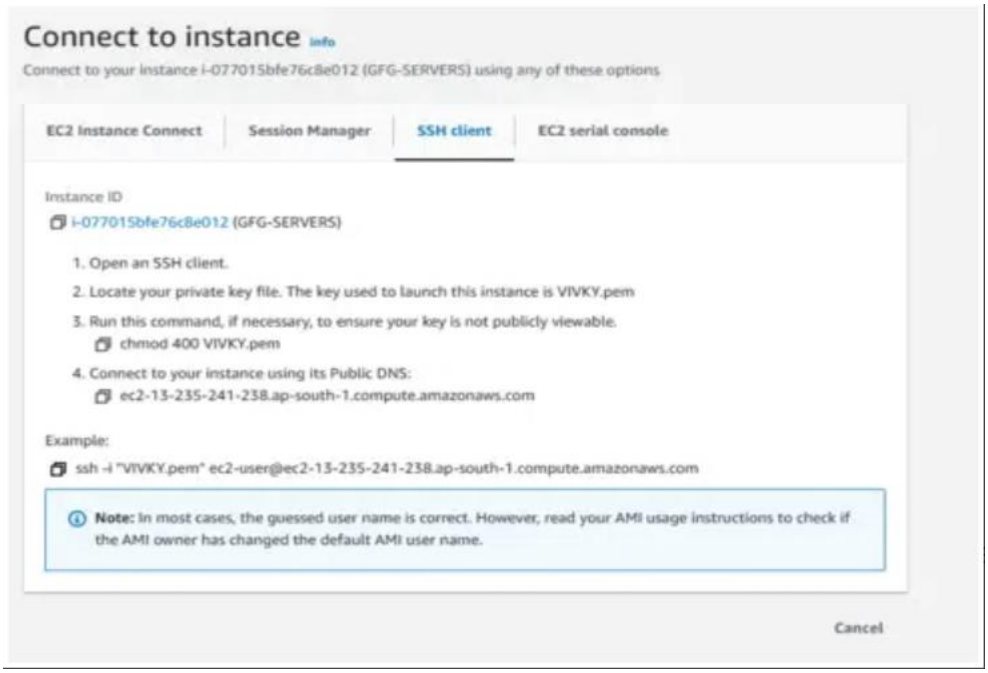
Now launch the instance as EC2 is a block to store data in AWS verify it's working after launching it successfully.

Steps To Connect Terminal Using SSH-Key

Step 1: Select the server to which you want to connect and click on the connect button at the top of that instance as shown in the image below.



Step 2: Copy the SSH key which is right following the example it will acct as a key-pair to connect to EC2-Instance.



Step 3: Open the terminal and go to the folder where your .pem file is located and paste the key that you have copied in AWS and paste it in the terminal.

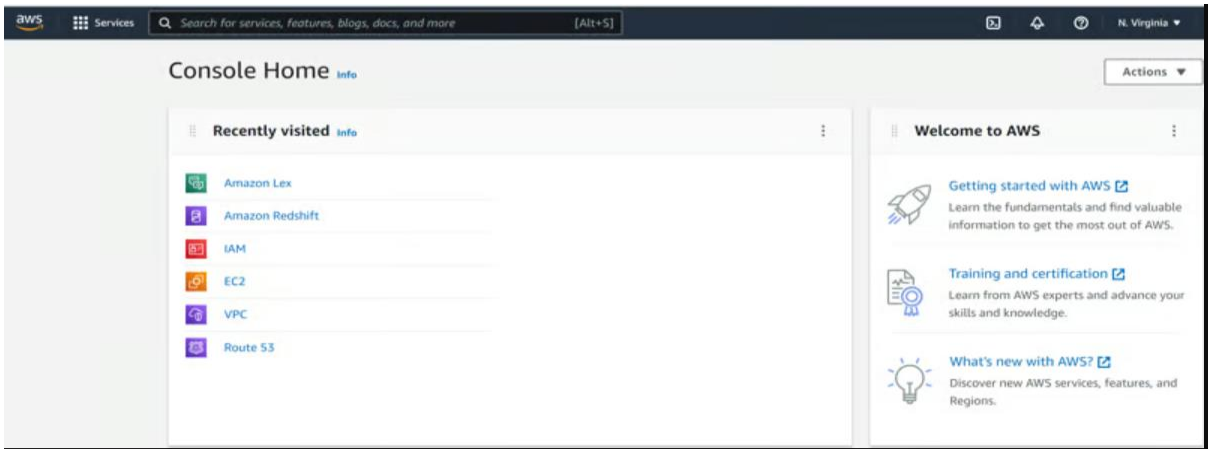
```
PS C:\Users\rknav\Downloads> ssh -i "VIVKY.pem" ec2-user@ec2-13-235-241-238.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-13-235-241-238.ap-south-1.compute.amazonaws.com (13.235.241.238)' can't be established.
ED25519 key fingerprint is SHA256:5VxqQUp4UBe9rUMXvZ1uL9UnzRNfpSFk8DjMybXVoyE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? YES
Warning: Permanently added 'ec2-13-235-241-238.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
[ec2-user@ip-172-31-34-45 ~]$
```

To know whether you connected to EC2-Instance perfectly or not you can check the IP-Adrees of the instance if the IP is displaying then you have connected successfully.

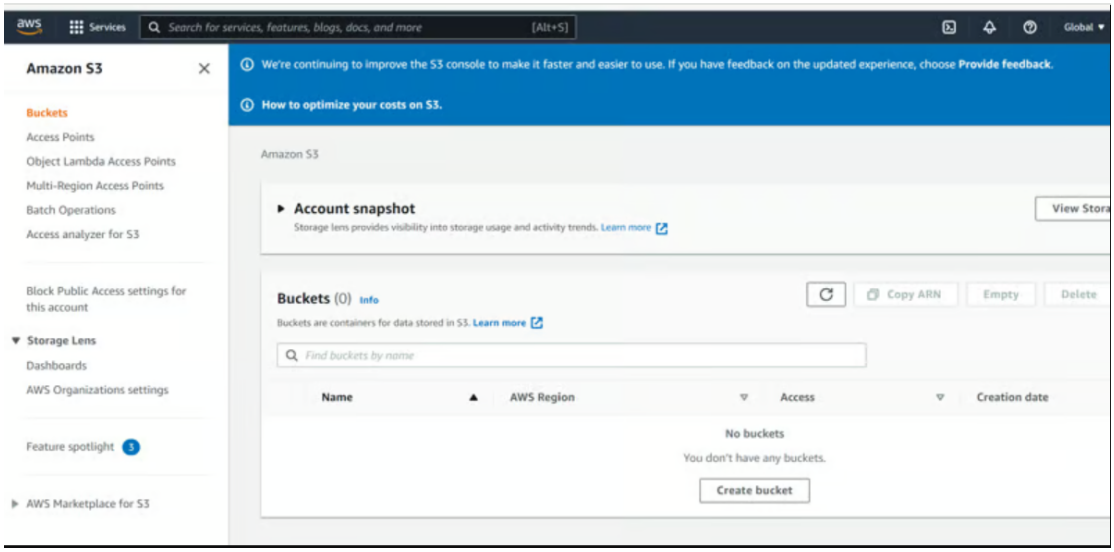
How to Creating an Amazon S3 bucket.

Step 1: Sign into the AWS Management Console using your IAM user credentials that you have already created. If you don't have IAM user credentials you can learn how to create one here.

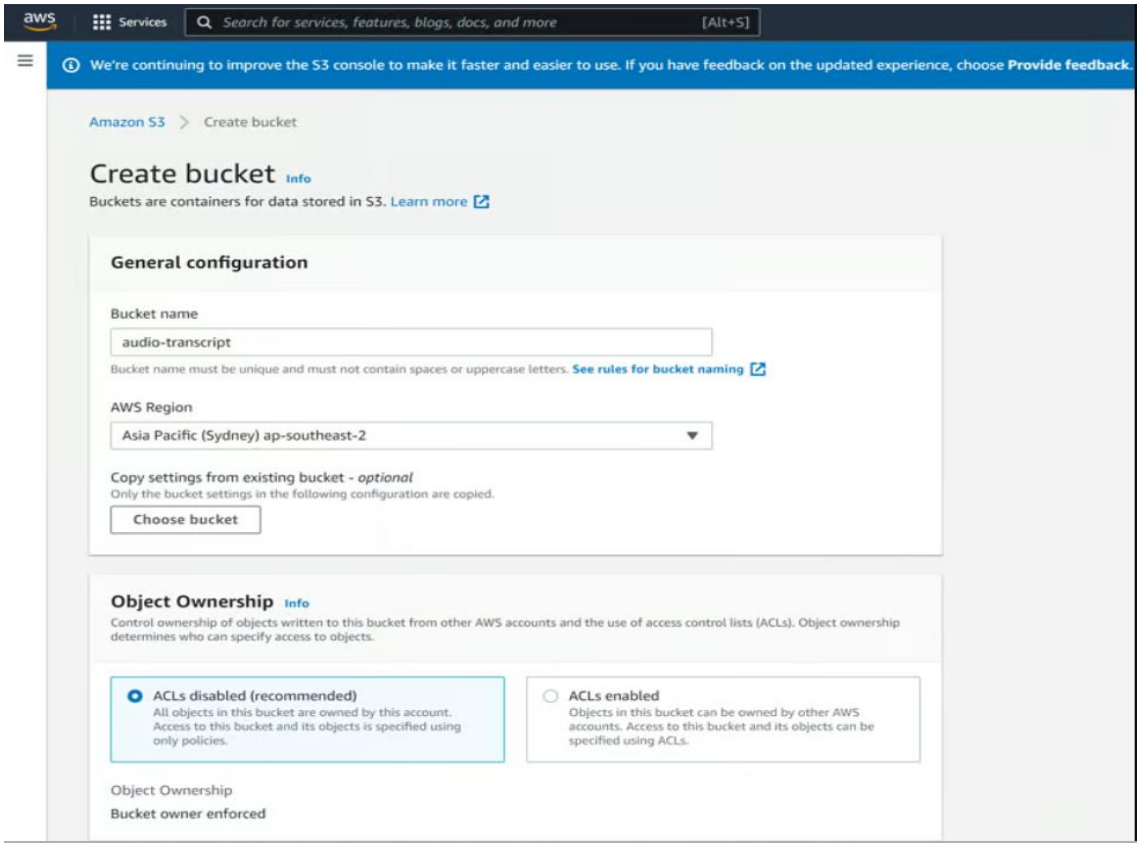
Login into the AWS Management with your IAM role.

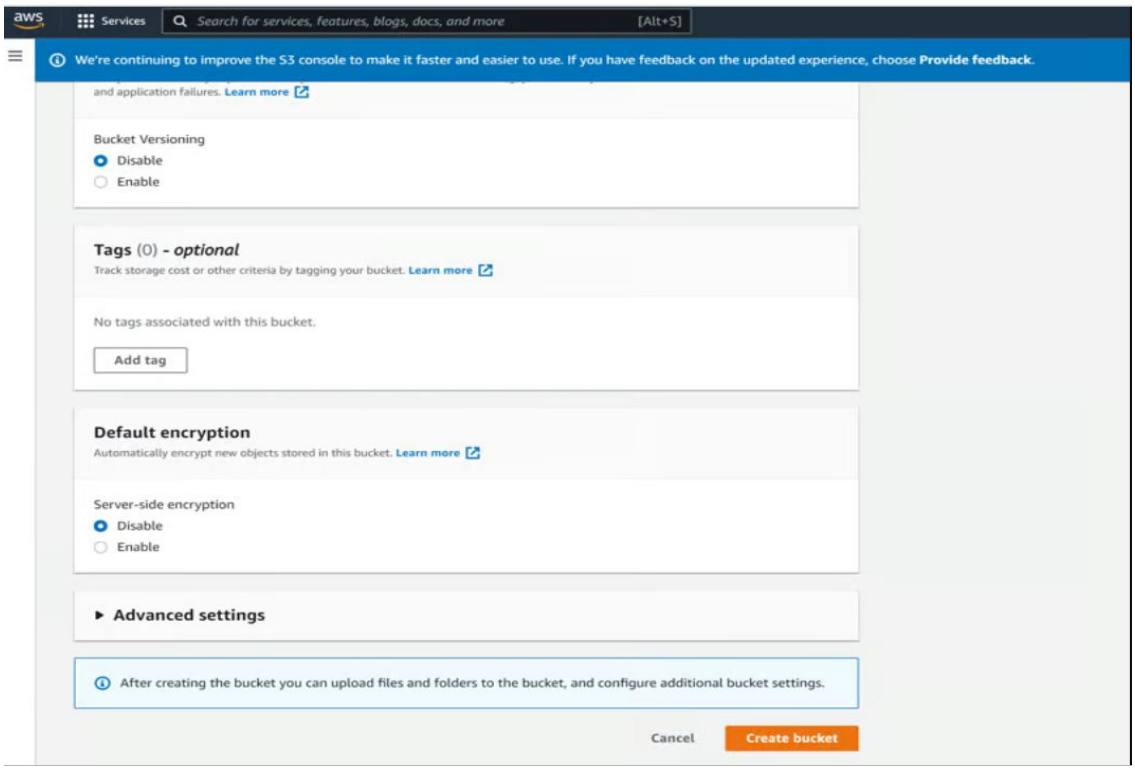


Step 2: Select S3 under the AWS Services navigation search bar On the S3 main screen click 'Create Bucket'

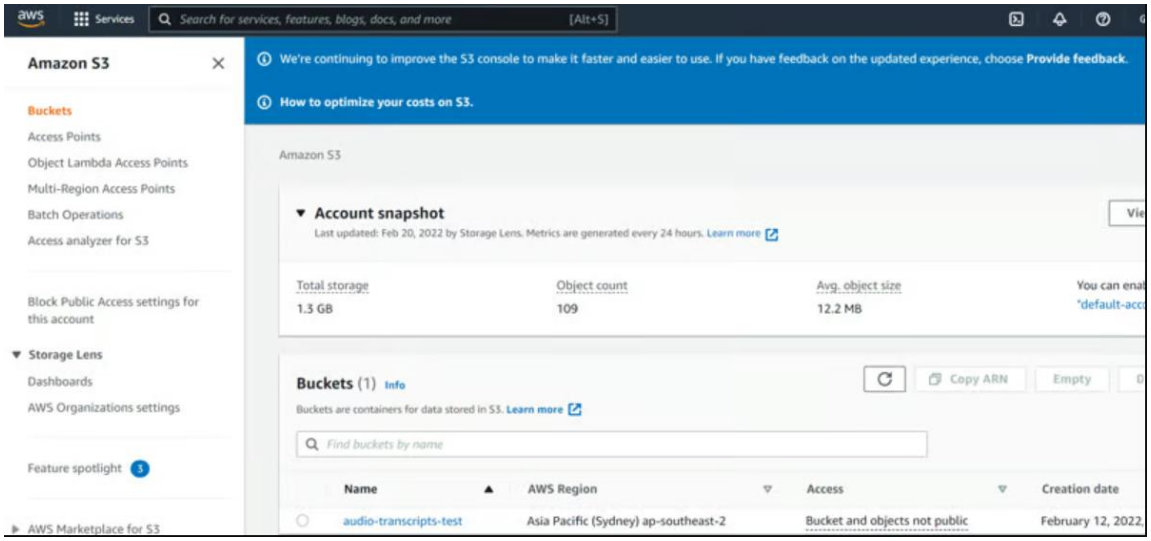


Step 3: Provide a descriptive name for your S3 folder and scroll to the bottom of your screen and click on 'Create Bucket'

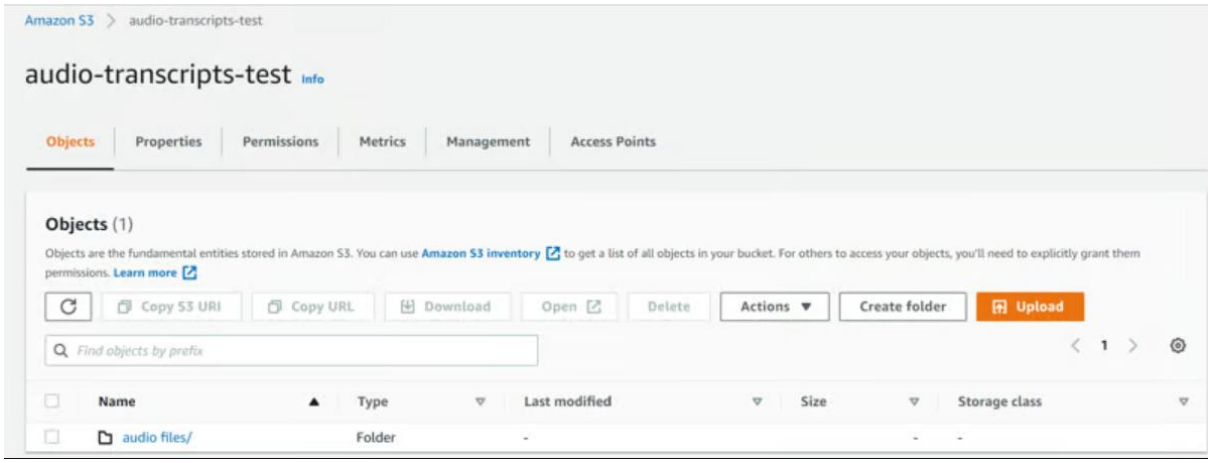




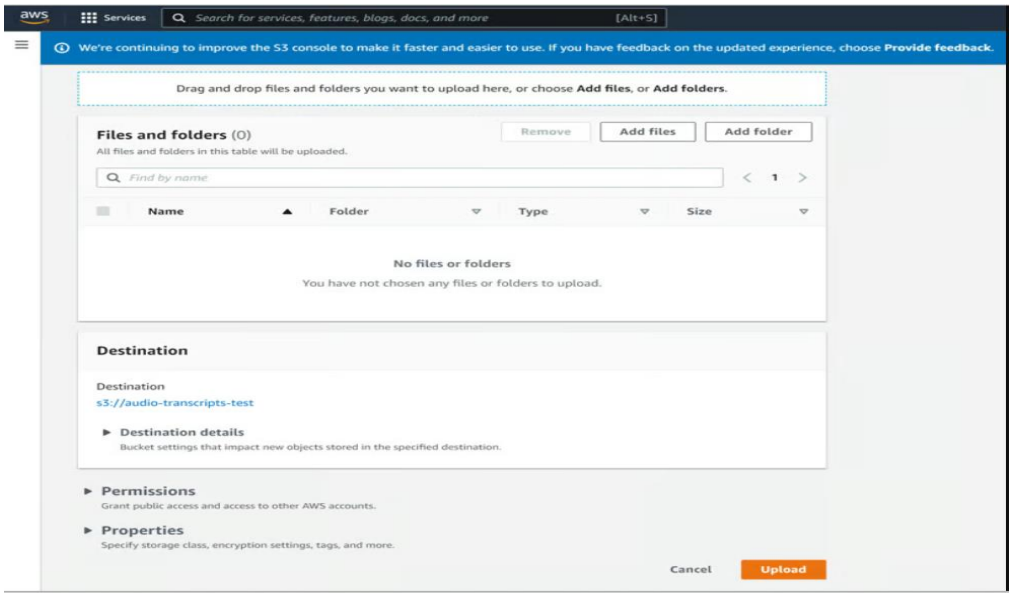
Step 4: Click into the hyperlink of the newly created S3 bucket and create a new folder e.g. called audio files.



Step 5: After the folder is created, click Upload

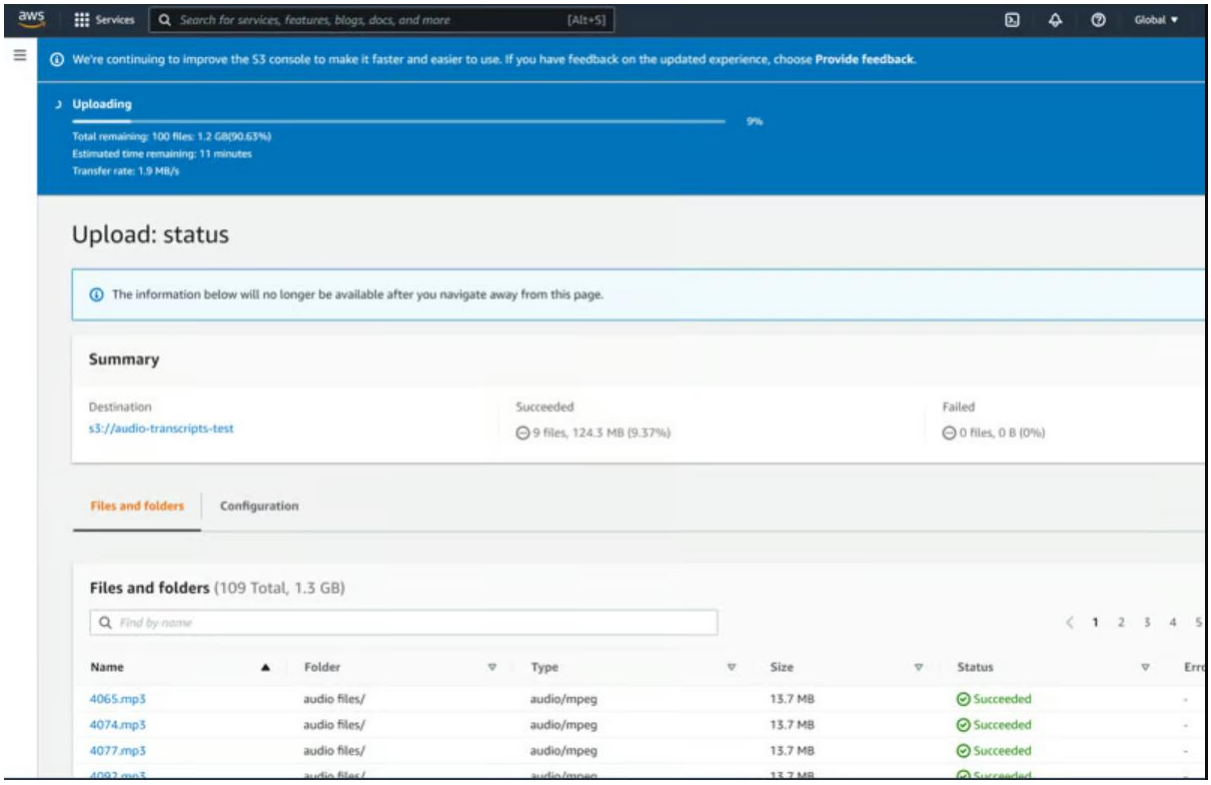


Step 6: Click 'Add files' or 'Add Folder' if you are uploading a folder of objects and finally click 'Upload'.

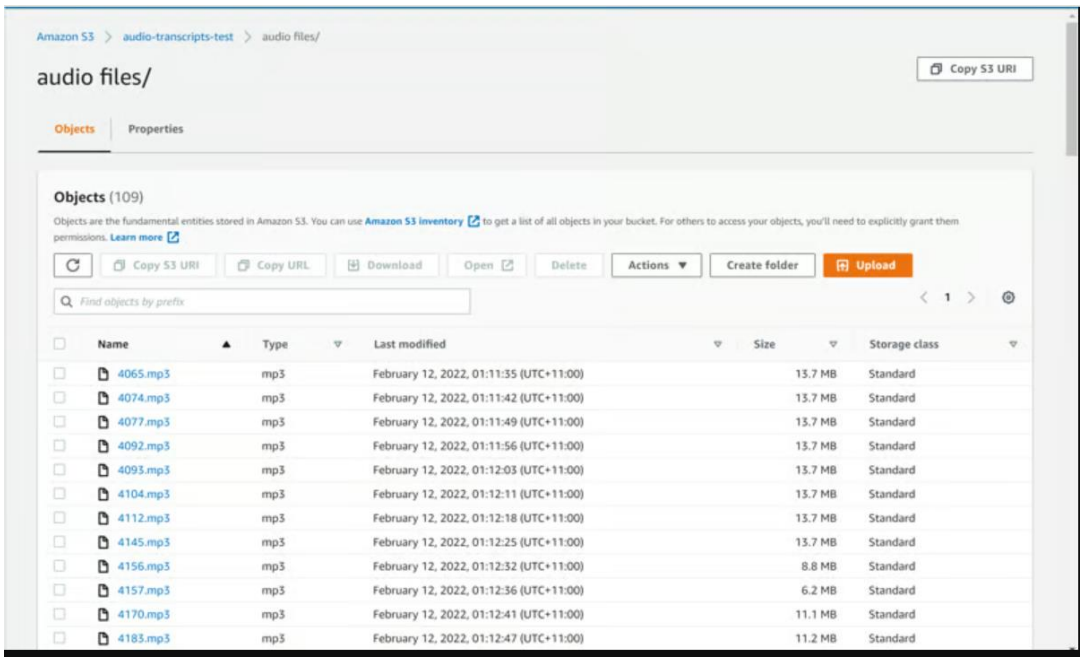


Step 7: Receive confirmation of file upload success

After the objects are uploaded into the S3 bucket, a confirmation message indicates the status of success.



Step 8: Click on the S3 bucket folder and inspect that all objects have been upload successfully

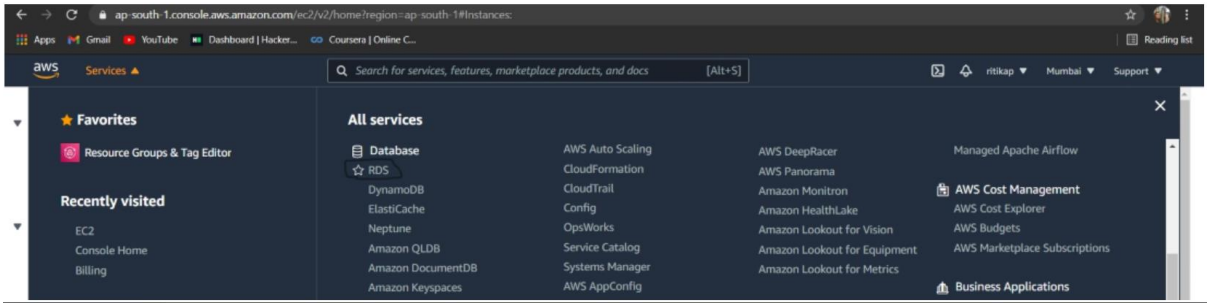


How to create Amazon RDS.

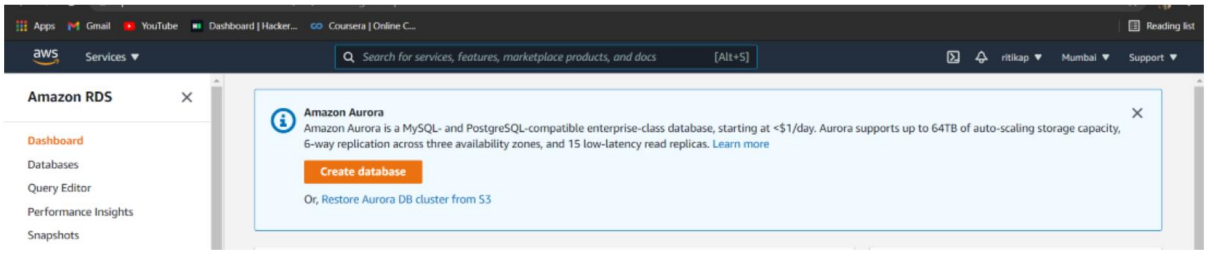
Amazon RDS or Amazon Relational Database Service makes the manipulation of databases much easier and handy for its users. It is very much similar to the traditional

relational databases along with numerous facilities of the cloud as a platform. Due to these extra advantages, people prefer AWS for managing their data.

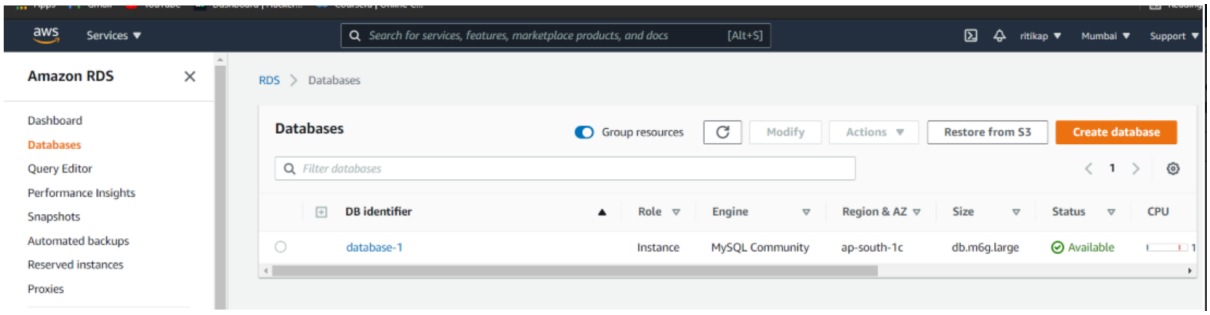
Step 1: First login into your AWS account and once your primary screen is loaded. Click on **services** which is written on the left and from the drop-down menu under **Database** there is an option **RDS**. Click on that and wait for the page to be loaded. Here is the image for better understanding.



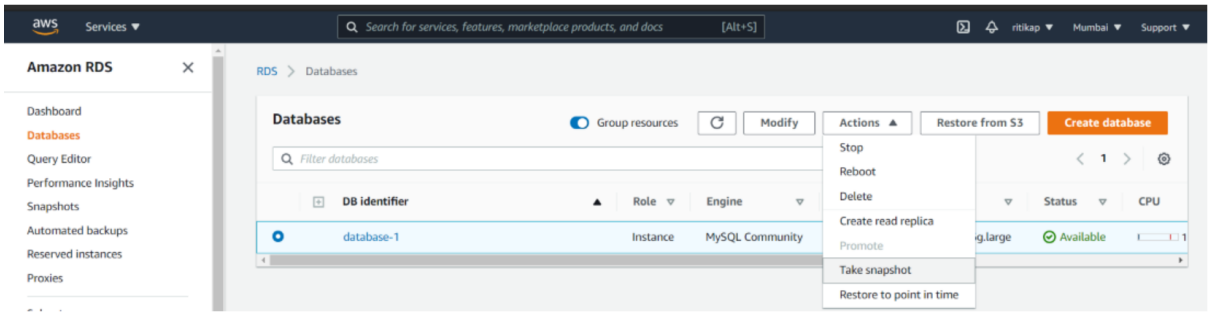
Step 2: Once the screen is available click on **create database.** And create your database as per your choice. The image to refer to is attached ahead.



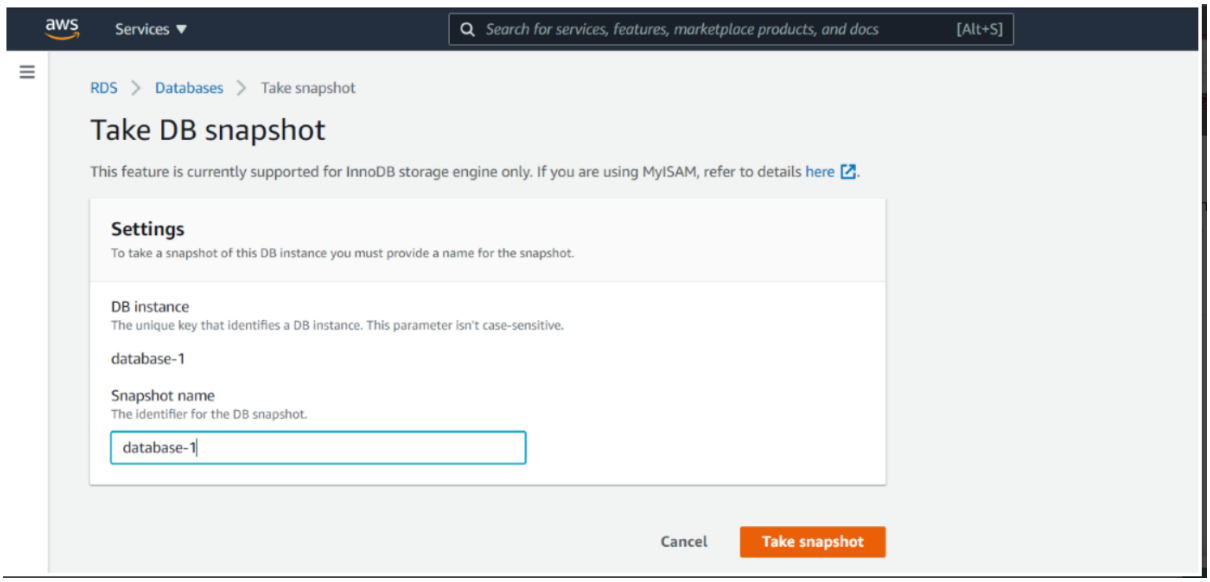
Step 3: After your database is successfully created. Now you are all set for creating a DB Snapshot. But make sure it's successfully created because without successfully creating a database we cannot generate a snapshot for that particular database. It may take a while to create don't bother. The status must say **Available**. Here is the image to know whether your database is successfully created or not, please refer to it.



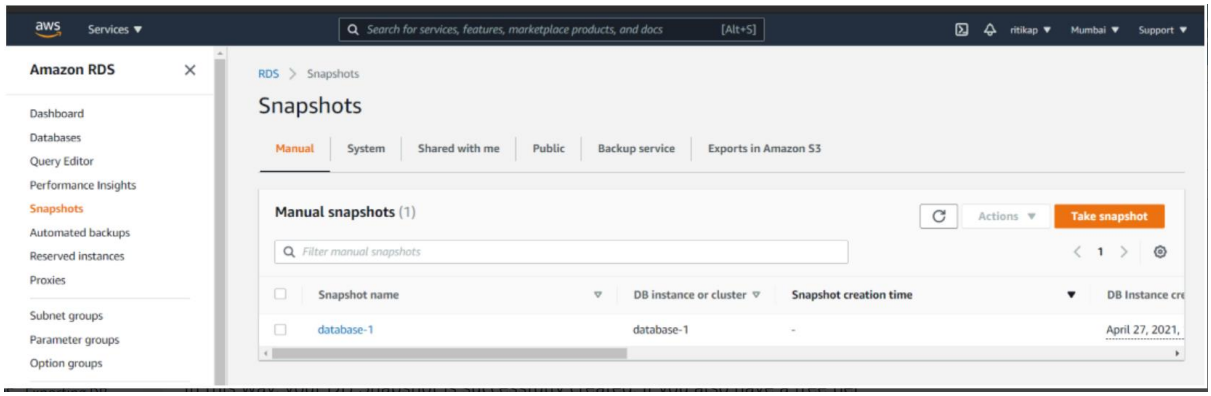
Step 4: Now, select the database for which you wish to create the **snapshot**. Select the database and click on **Actions** and select the option **Take Snapshot** from the list. Please refer to the image for better understanding.



Step 5: Click on **Snapshots.** And write the name of the database for which you are willing to create the snapshot. And finally, click on **Take Snapshot**. Refer to the image attached ahead.

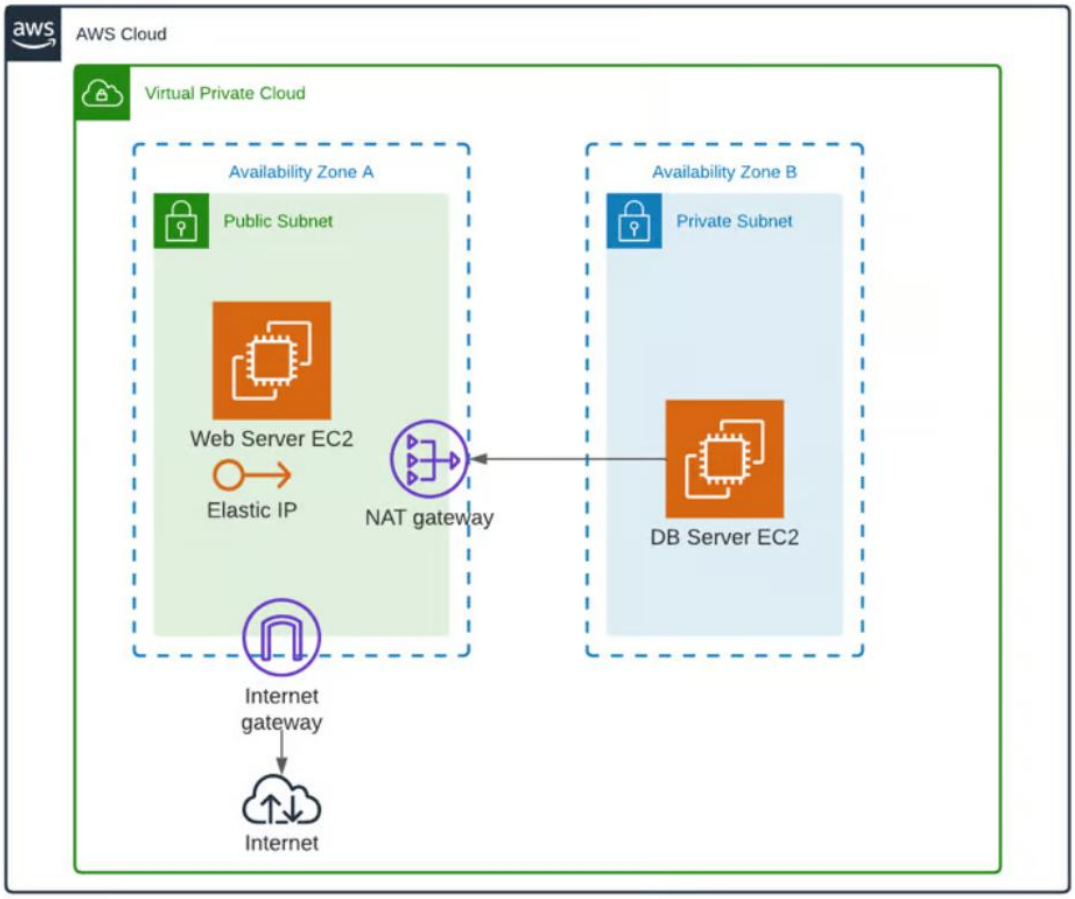


Step 6: Now, from the left drop-down list click on **Snapshots.** And once the fresh screen will be loaded, you'll be able to see the snapshot. The snapshot will have the same name as that of the database. Refer to the image for a much better understanding.



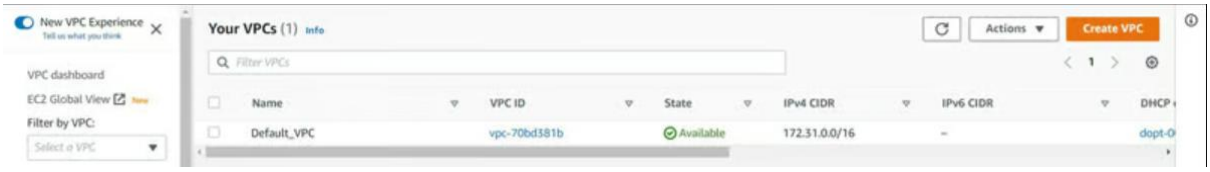
How to create Amazon VPC with screenshots.

AWS VPC Creation - Step by Step

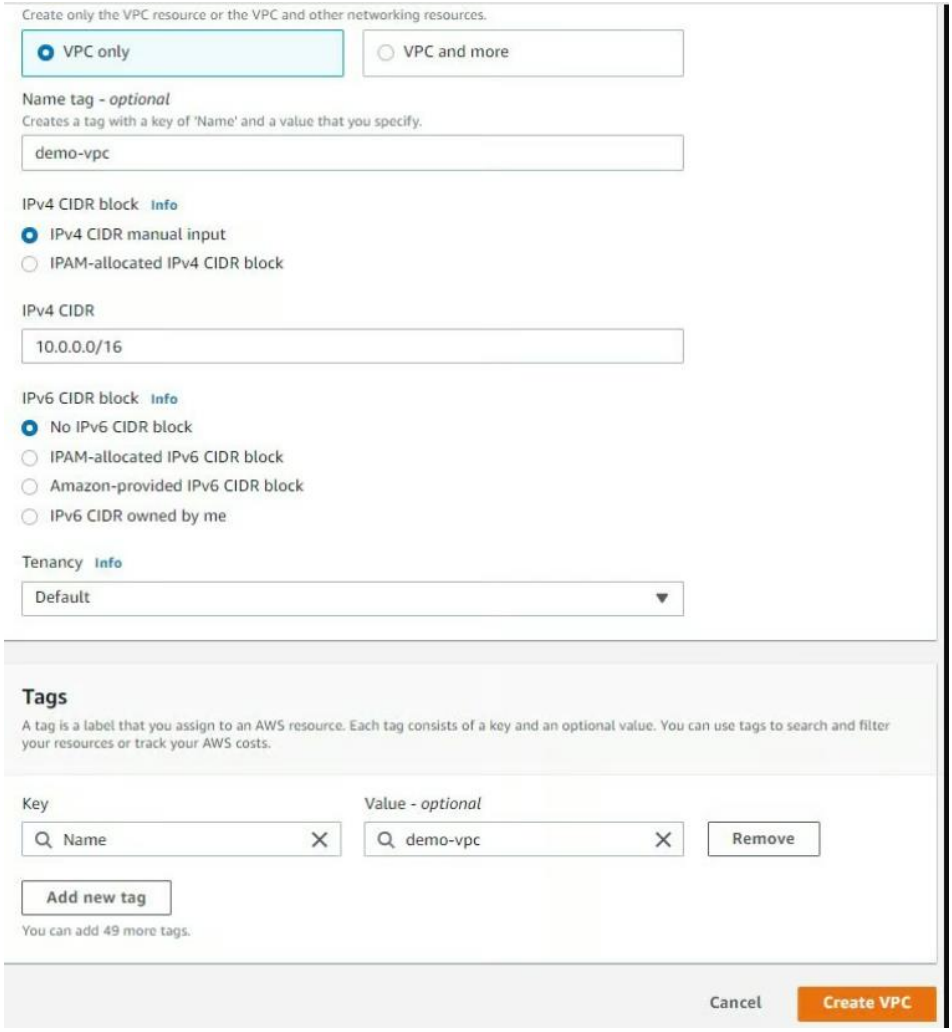


Firstly, go through Architecture Diagram. AWS provides many services for manage the Virtual network. In this guide, you can create AWS VPC step by step.

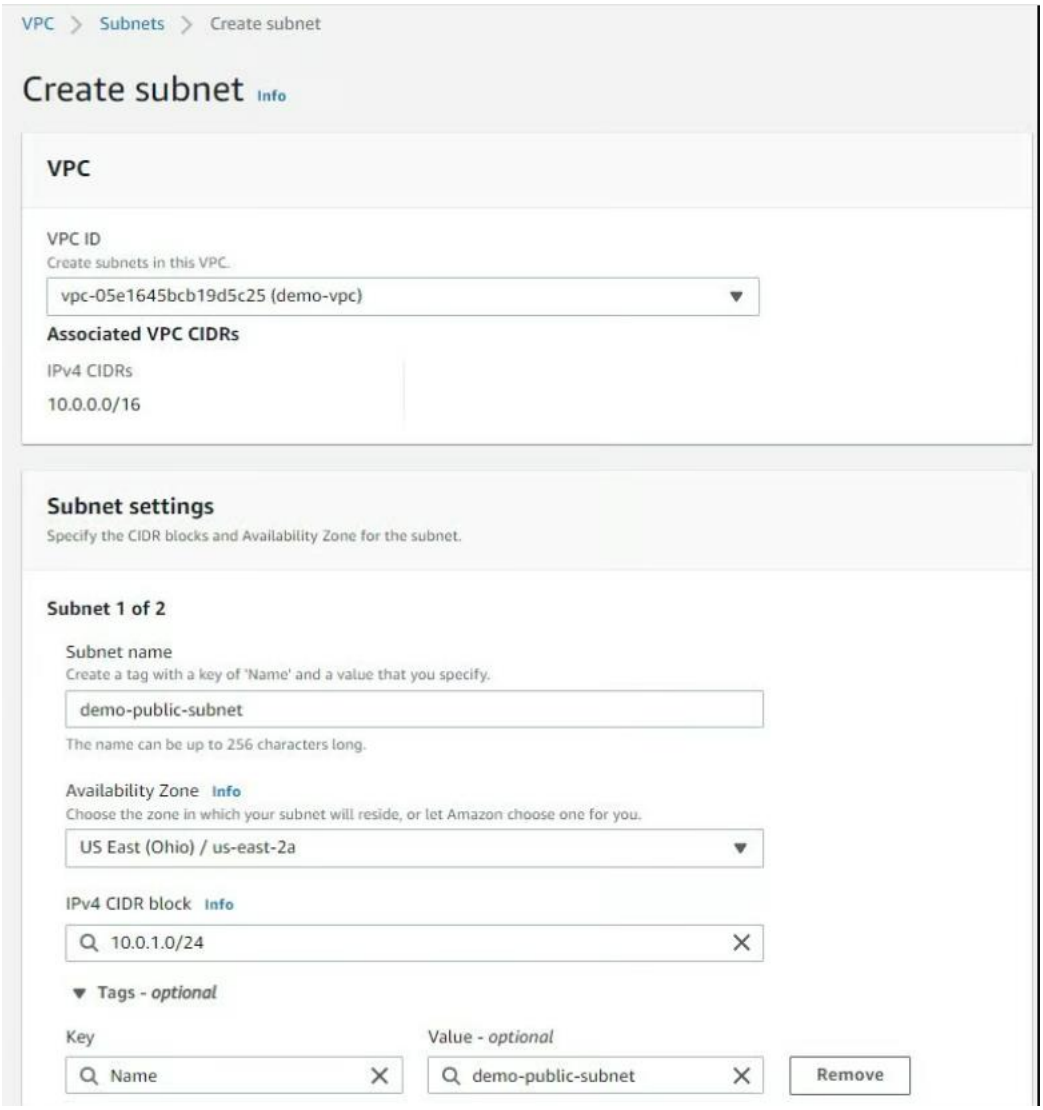
In AWS Management Console search and go VPC services after that click creates VPC button.



Give the name for VPC and add IPv4 CIDR. You can use the 10.0.0.0/16 CIDR range.

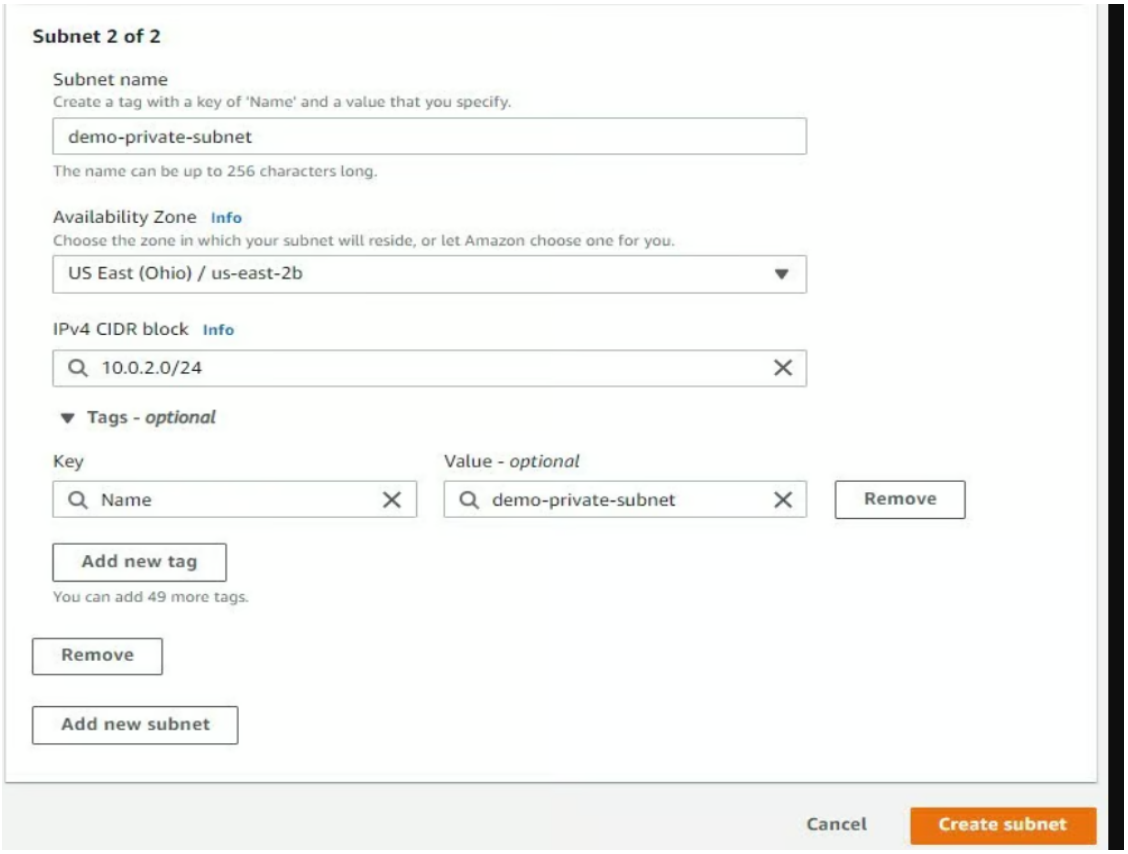


Create Two Subnets. Firstly create a public subnet. You can use the 10.0.1.0/24 CIDR range for the public subnet. Select the availability zone and give the name for the subnet.

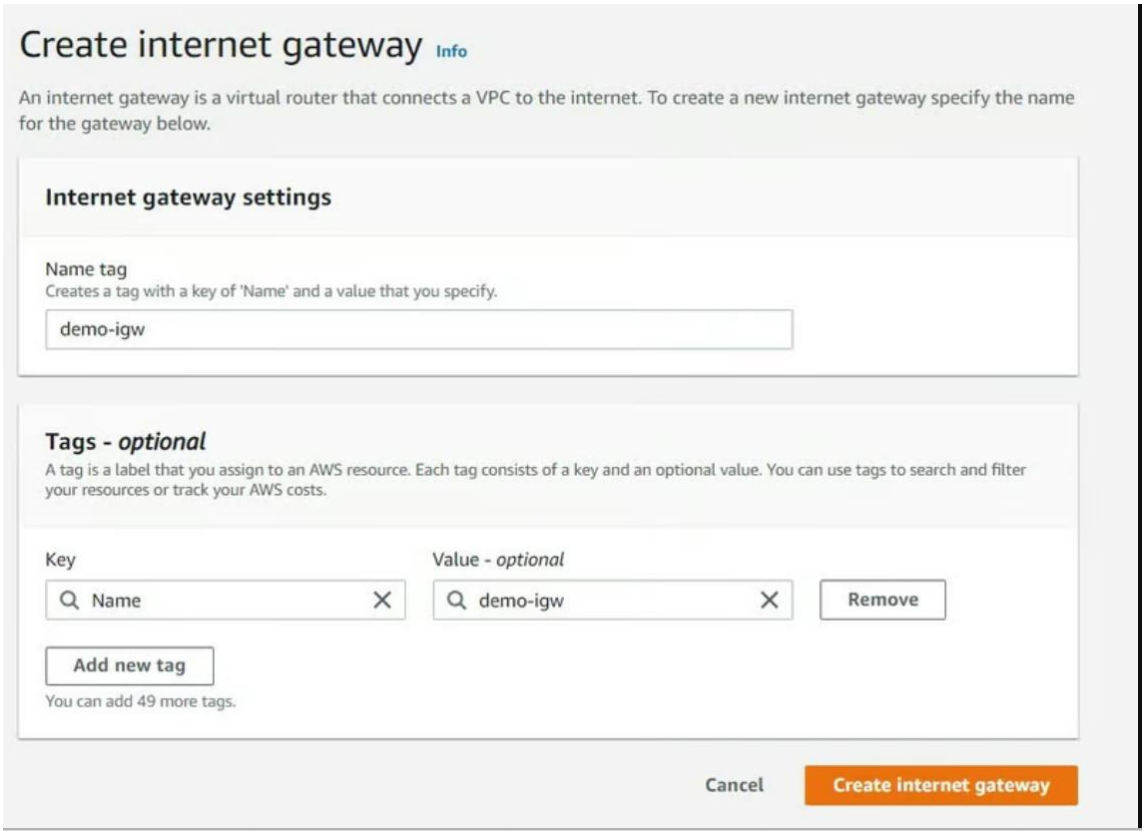


Now the same interface, click add new subnet button for private subnet creation.

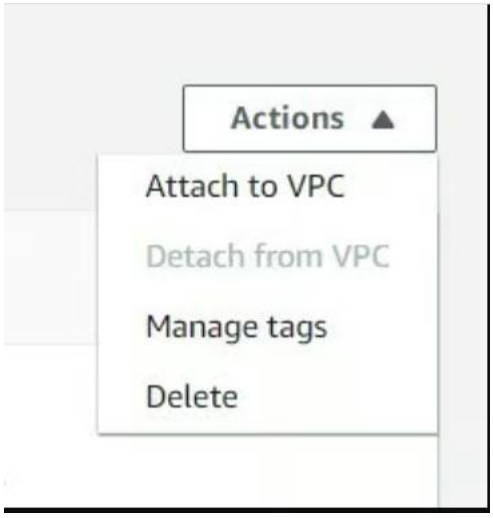
Add IPv4 CIDR as 10.0.2.0/24. After that give the subnet name and select the availability zone. finally, click create subnet button.



Now create an internet gateway for public subnet internet access. In VPC Console select Internet Gateway and create internet gateway. Give the name of Internet Gateway and hit create button.

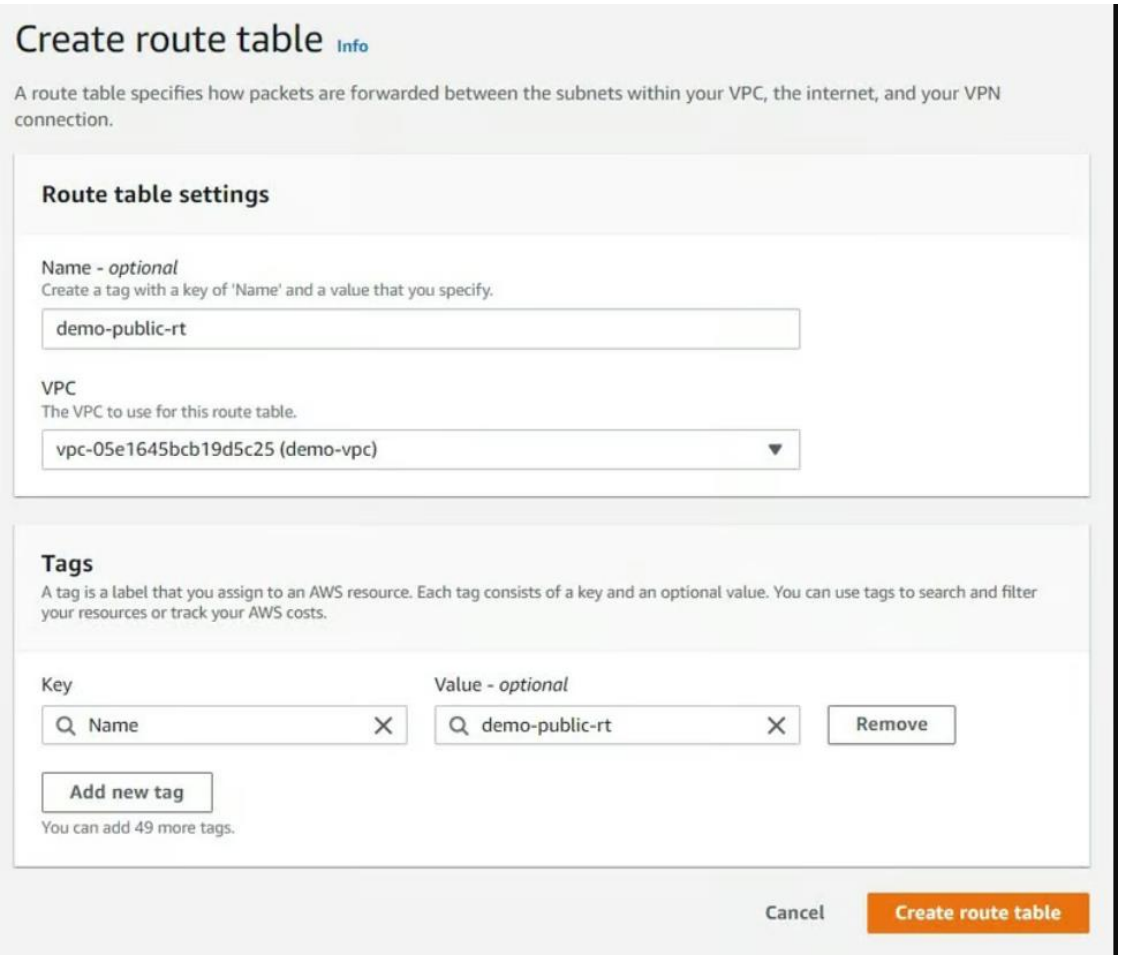


Next, we want to attach this internet gateway for VPC. You can select the internet gateway and click Attach to VPC.

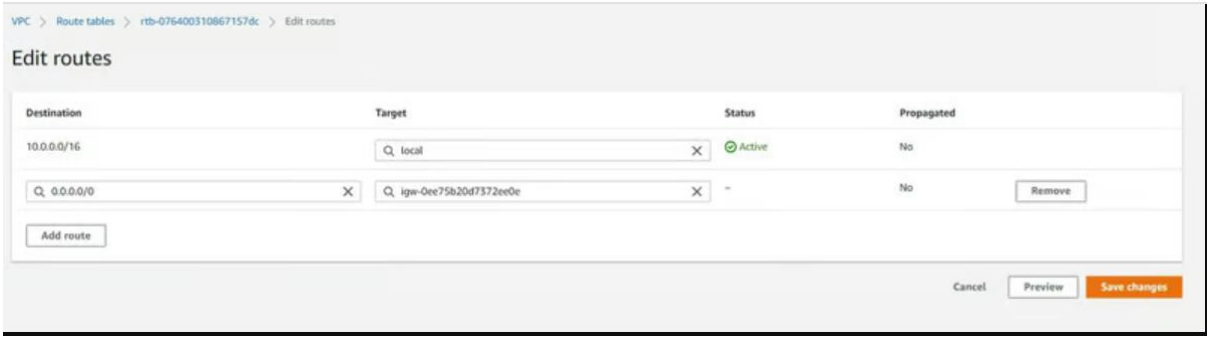


Next, select the previously created VPC and click attach internet gateway button.

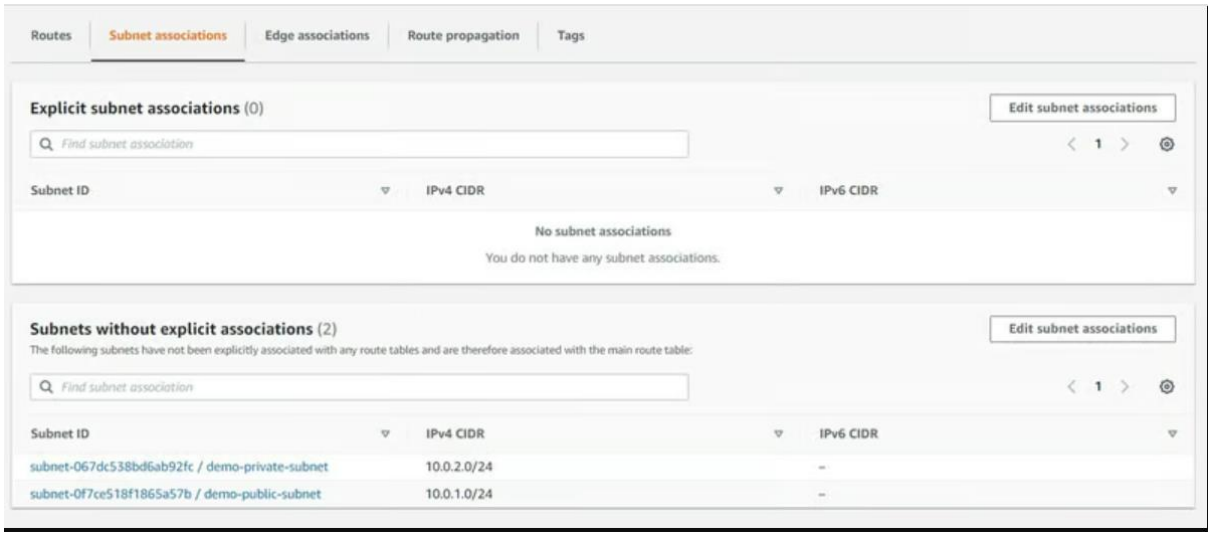
Next, Go to the route table and create a route table for the public subnet. give the name for the route table and select the previously created VPC. The next click creates the route table button.



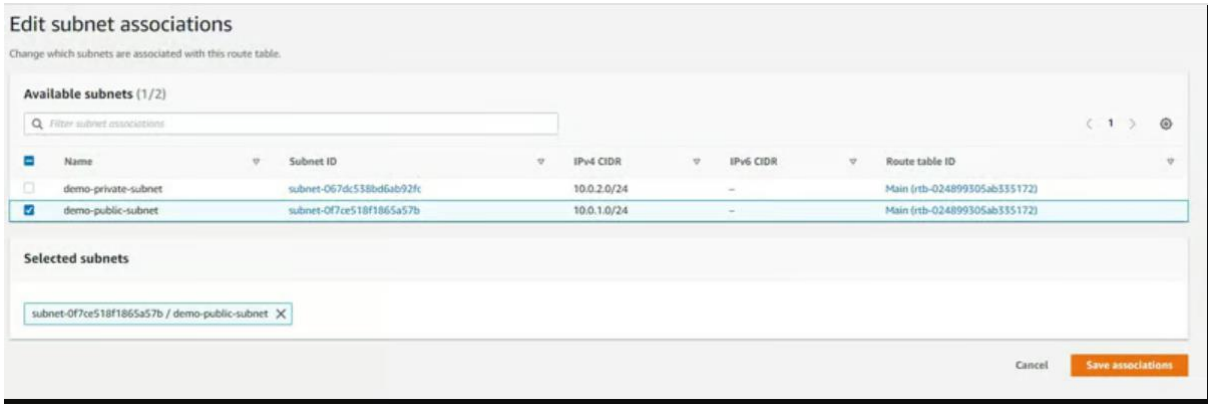
Click the public subnet route table click the Route tab and add route 0.0.0.0/0 and select the previously created Internet Gateway. next hit the save changes button.



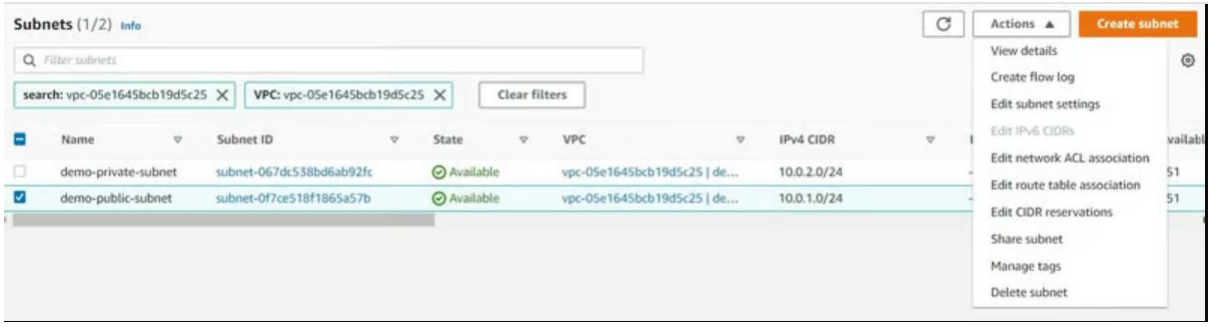
In the public subnet route table click the subnet association section and click edit "Explicit subnet associations" section.



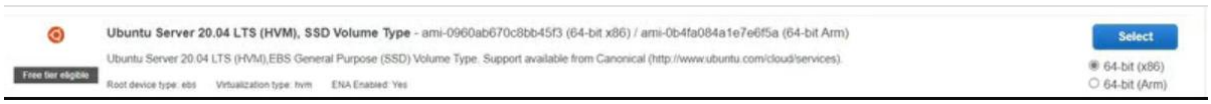
Next, select public subnet and click the save association button.



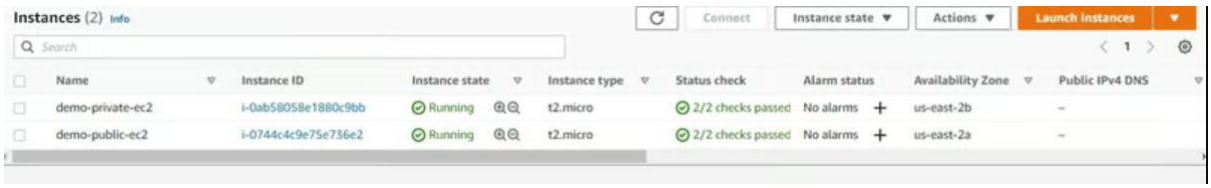
Now we go to public subnet and click subnet settings and enable "Enable auto-assign public IPv4 address" and click save.



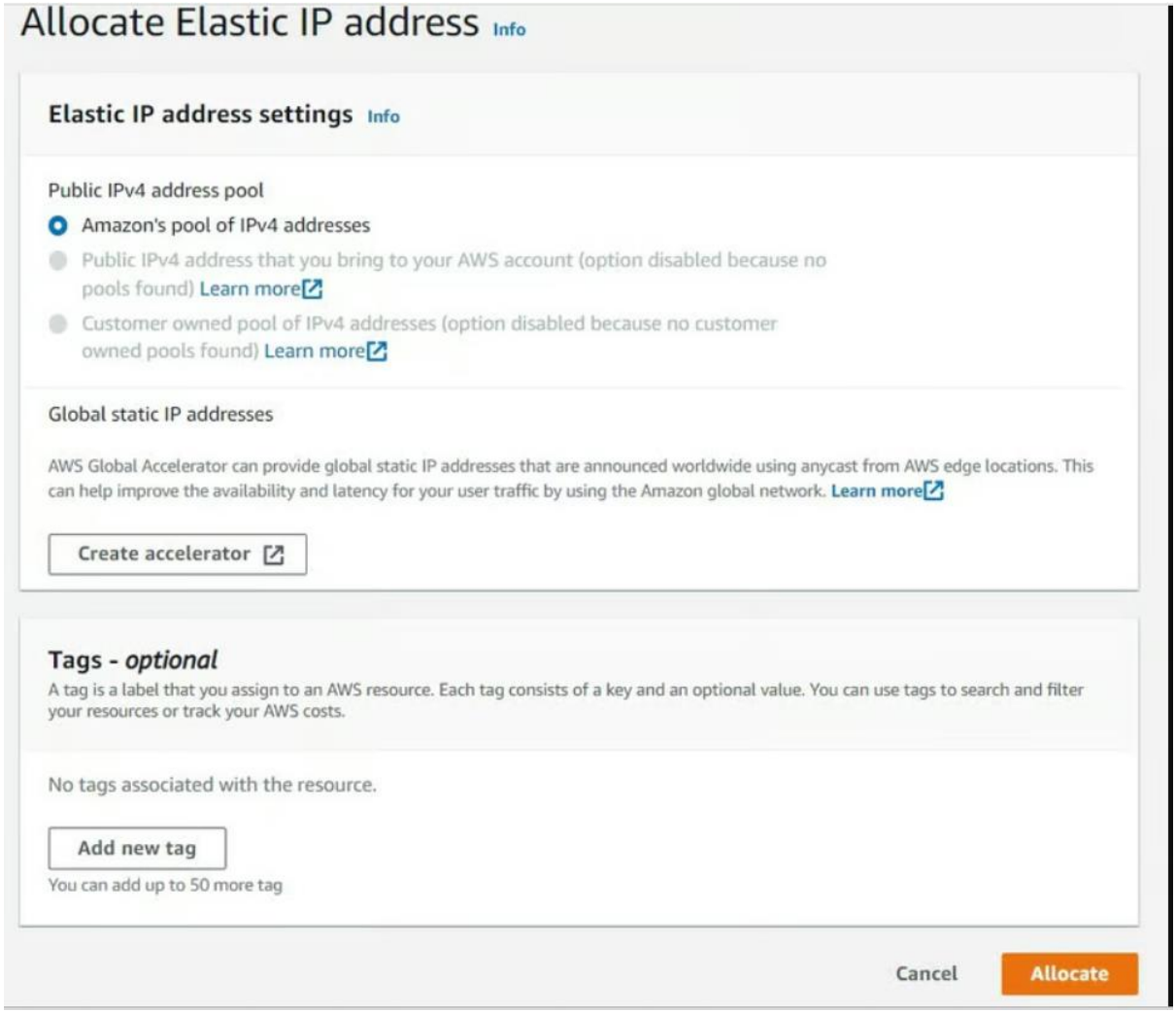
Next, we want to create 2 ec2 instances inside public and private subnets. Use ubuntu server 20.04 LTS AMI.



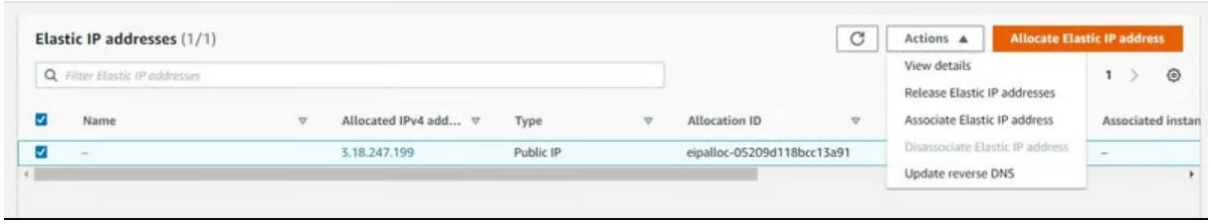
You can see the ec2 instance is up and running.



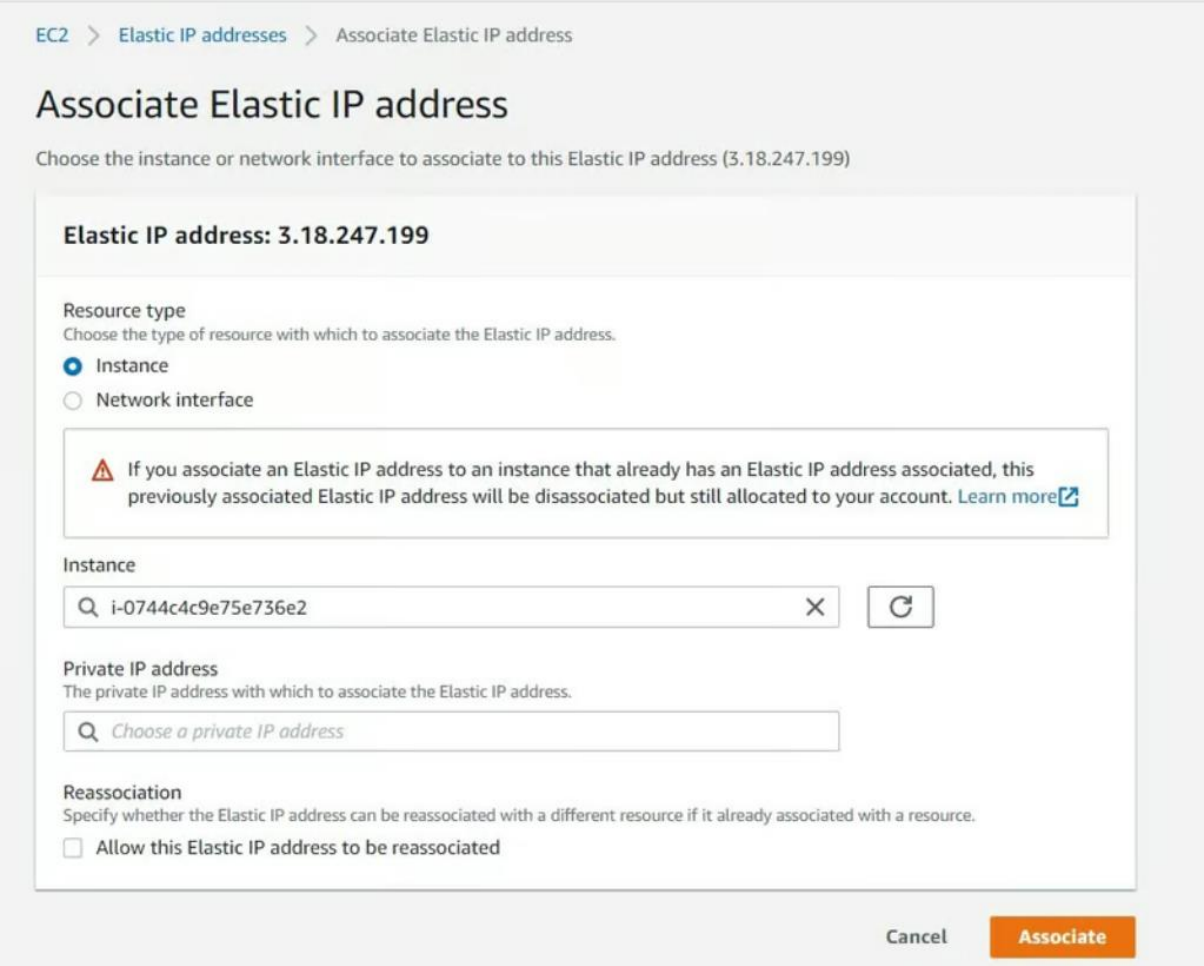
In the EC2 console click Elastic ip and create Elastic ip like following.



Next, select the newly created Elastic ip and click allocate elastic ip button.



Select previously created public ec2 and click the associate button.



Next, connect public EC2 via SSH client or EC2 Instance Connect. Try to update ubuntu. Working it means public ec2 can connect internet.

```
ubuntu@ip-10-0-1-218:~$ sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 c-n-f Metadata [265 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [144 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/multiverse Translation-en [104 kB]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 c-n-f Metadata [9136 B]
Get:11 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1947 kB]
Get:12 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [353 kB]
Get:13 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [15.6 kB]
Get:14 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [1141 kB]
Get:15 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [162 kB]
Get:16 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 c-n-f Metadata [592 B]
Get:17 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [923 kB]
Get:18 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [208 kB]
Get:19 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [20.9 kB]
Get:20 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [24.4 kB]
Get:21 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [7336 B]
Get:22 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 c-n-f Metadata [592 B]
Get:23 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [44.8 kB]
Get:24 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/main Translation-en [11.3 kB]
Get:25 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/main amd64 c-n-f Metadata [976 B]
Get:26 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/restricted amd64 c-n-f Metadata [116 B]
Get:27 http://us-east-2.ec2.archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [23.7 kB]
```

Now, we try to SSH into EC2 in Private Subnet. (It means bastion host). In Linux Terminal You can using the following steps.

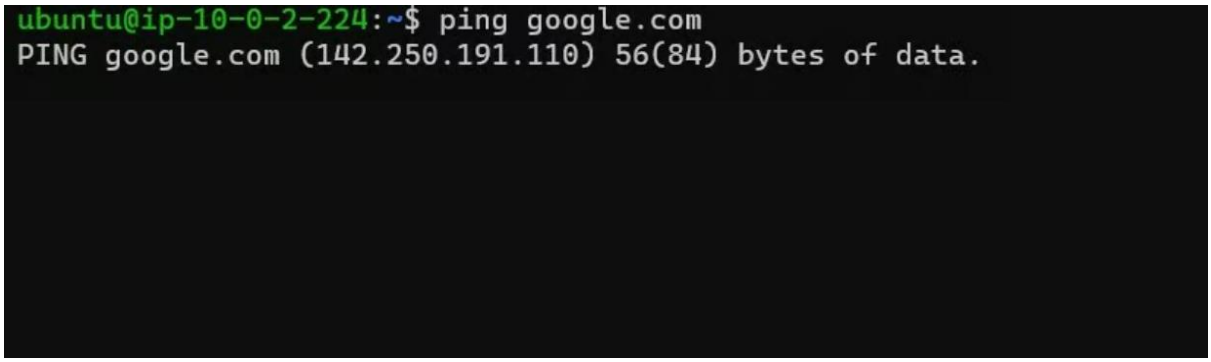
- Configuring the SSH agent using the following command.
  ssh-add -L {{ssh-keyfile-name.pem}}
- Next, connect the bastion host(Public EC2) using this command.
  ssh –A ubuntu@{{Bastion-IP-address or DNS}}
- Connect to the private instances from the bastion host (Agent Forwarding).
  ssh ec2-user@{{InstanceIP or DNS}}

Inside the private ec2 terminal "ping google.com" it's not given any response because it doesn't connect internet.

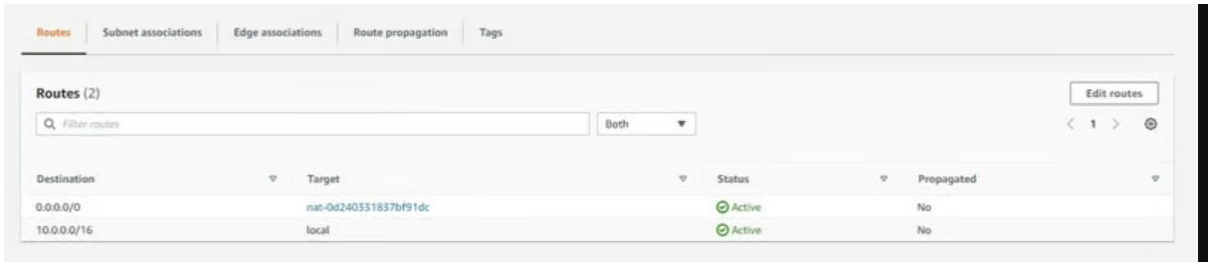
Now, we try to SSH into EC2 in Private Subnet. (It means bastion host). In Linux Terminal You can using the following steps.

- Configuring the SSH agent using the following command.
  ssh-add -L {{ssh-keyfile-name.pem}}
- Next, connect the bastion host(Public EC2) using this command.
  ssh –A ubuntu@{{Bastion-IP-address or DNS}}
- Connect to the private instances from the bastion host (Agent Forwarding).
  ssh ec2-user@{{InstanceIP or DNS}}

Inside the private ec2 terminal "ping google.com" it's not given any response because it doesn't connect internet.

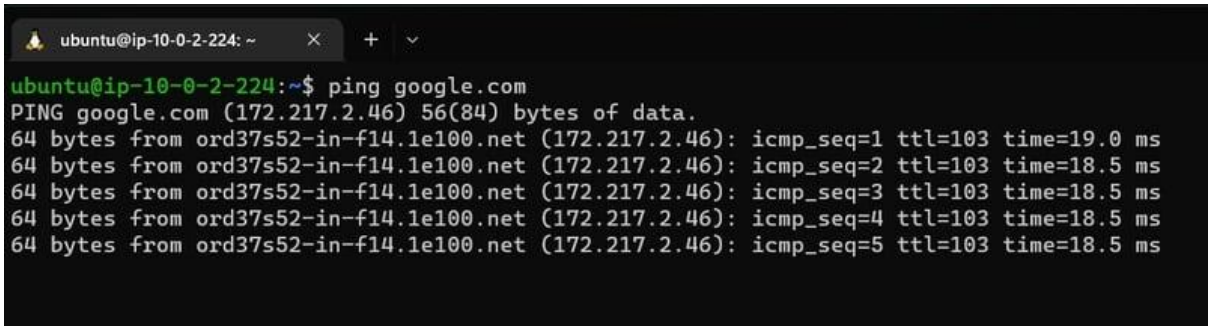


Next, go to the VPC console and create NAT Gateway. In NAT gateway creation select subnet as public subnet and give the name for NAT Gateway. After that click Allocate Elastic IP button and finally click create nat gateway button.



Next, add subnet association. Select private subnet and click save association button.

Finally, try to "ping google.com" inside a private subnet terminal. You can see the following output.



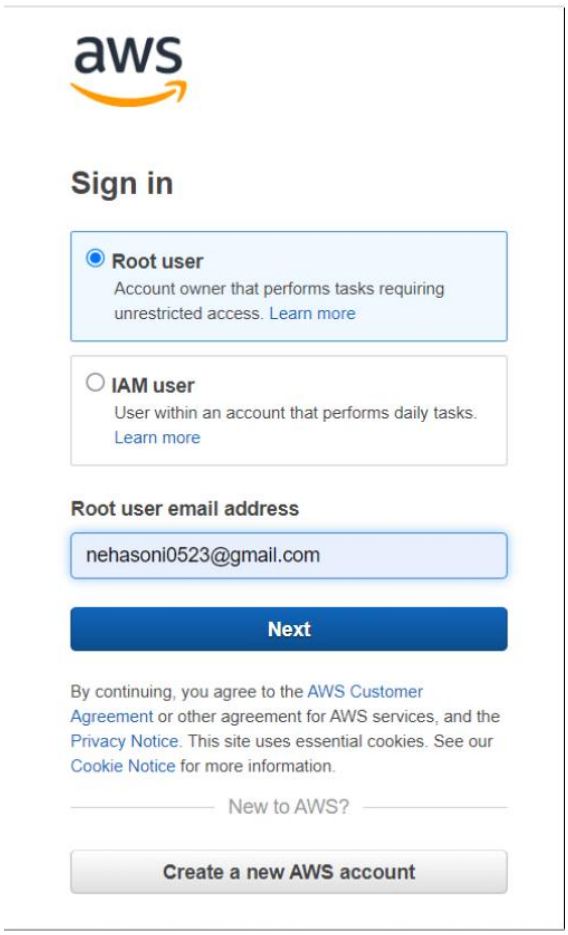
Congratulations, Now you can create AWS VPC.

For clean up

- Terminate instances
- Release Elastic IP
- Delete NAT Gateway
- Delete Internet Gateway
- Delete VPC

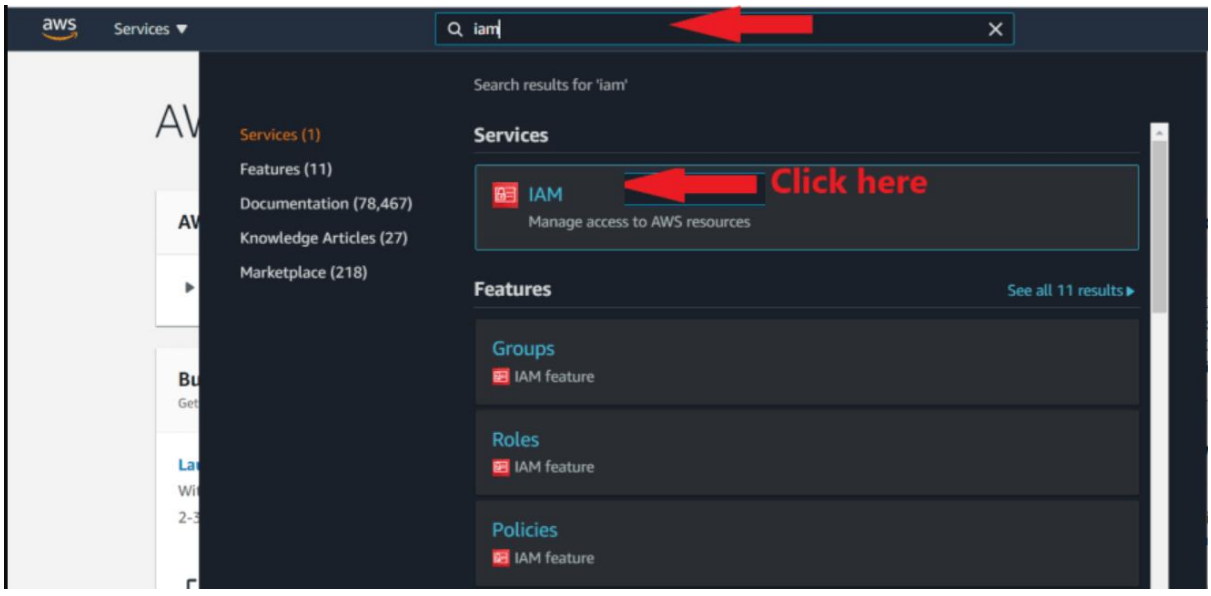
How to create IAM users and roles.

Step 1:- Open your favourite browser and navigate to AWS Login Page

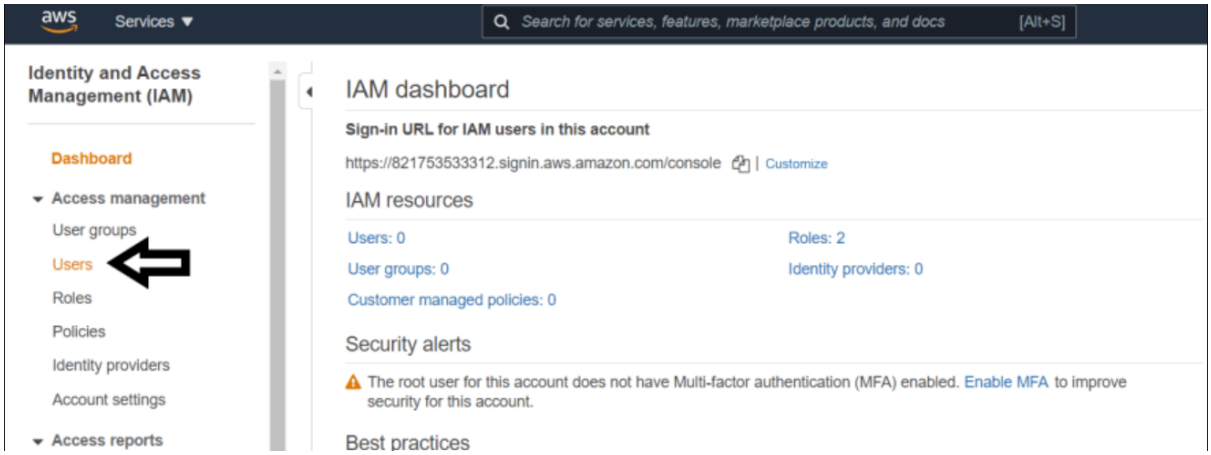


Then enter in your password and click submit. You have now successfully signed in to the AWS Management Console.

Step 2:- In the search bar type IAM and click on IAM(Manage access to AWS resources) to navigate to IAM Dashboard.

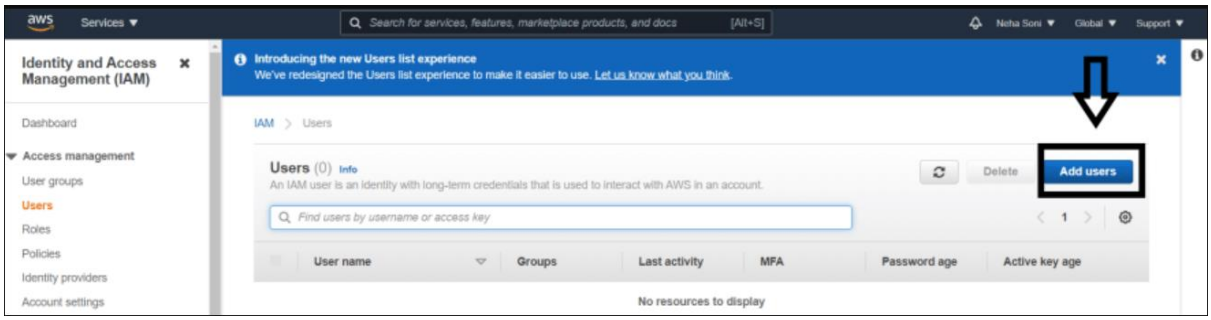


Step 3:- On the left side of the page, you should see an option called **Users**.



Click on that option, and you will be taken to the user's page.

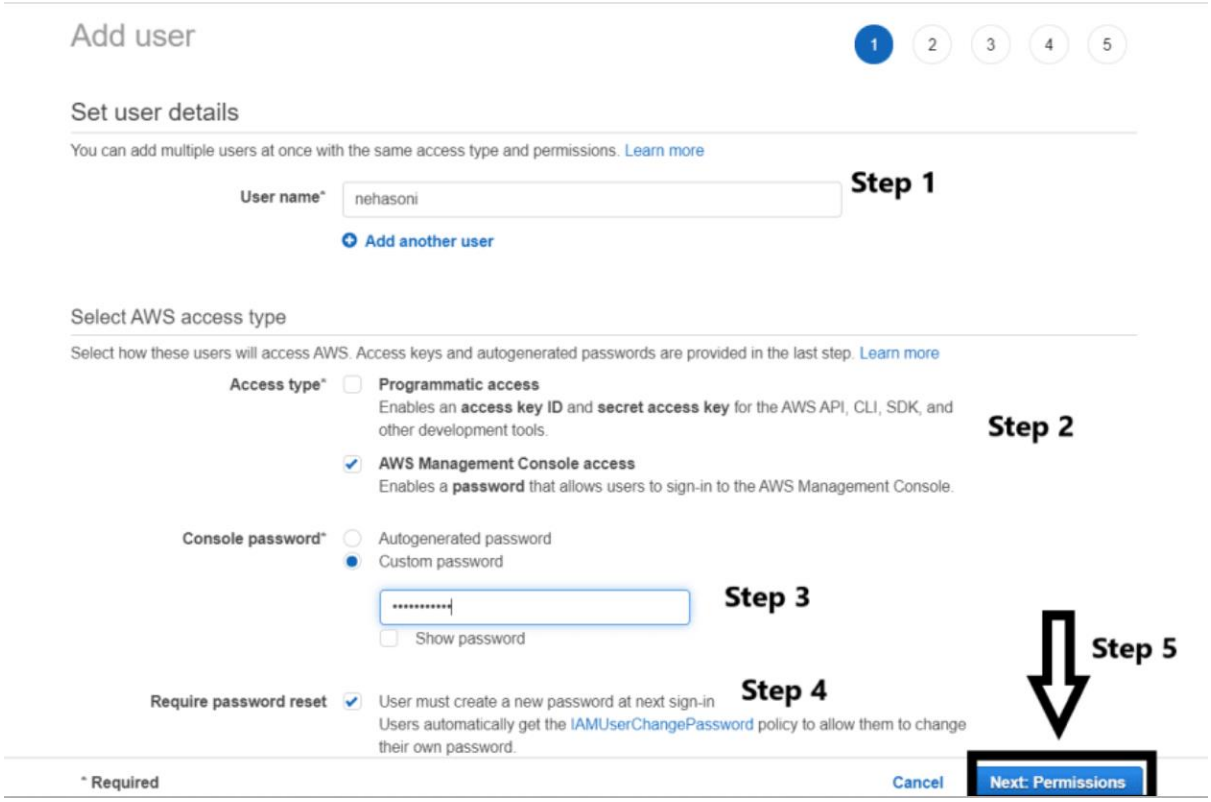
Step 4:- Click on Add user button to create a user.



Step 5:- Enter a username that can be used to log in later.

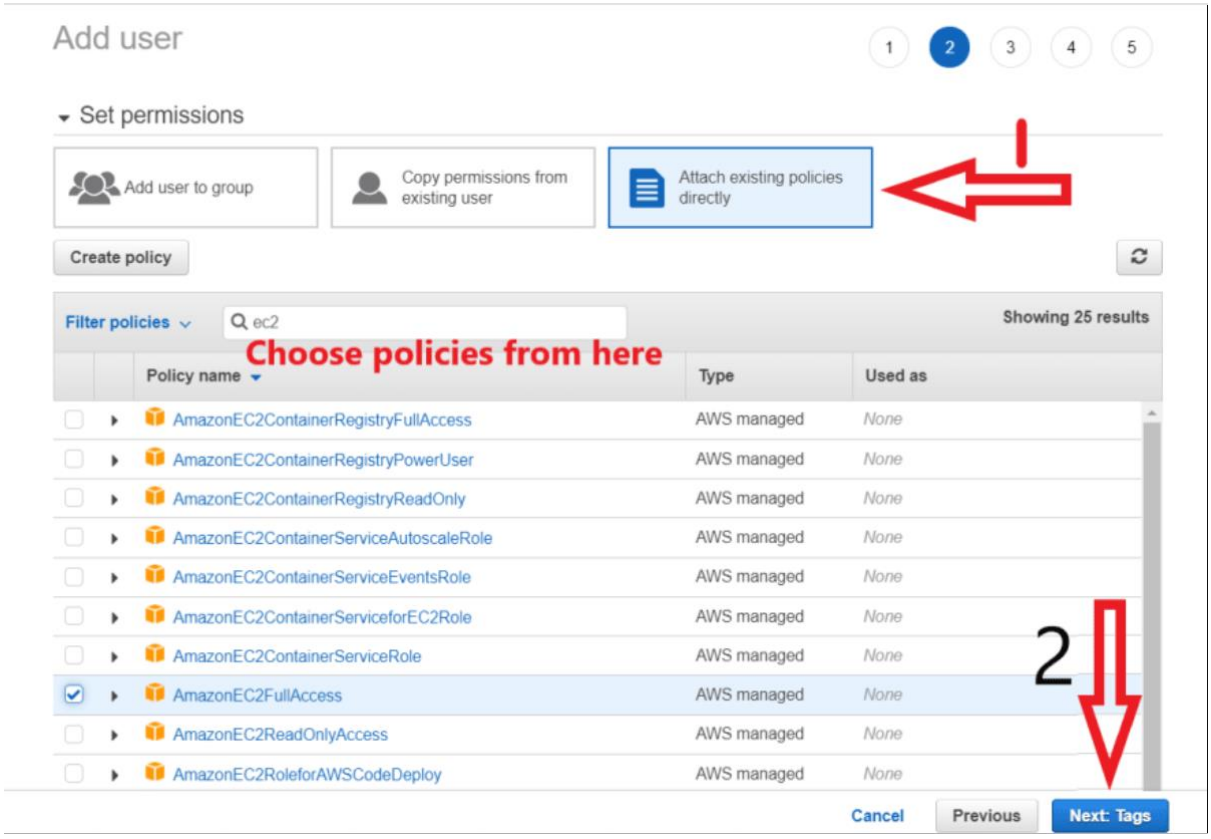
- **Access type**
  When creating a user, you must choose between the following access types:
  a.) Programmatic access: If the IAM user needs to make API calls, use the AWS CLI or the Tools for Windows PowerShell then choose Programmatic access.
  b.) AWS Management Console access: If the user needs to access the AWS Management Console, create a password for the use.
  For now I will go with the second option i.e., AWS Management Console access
- **Console Password**
  For Console password, choose one of the following:
  Autogenerated password:- Each user will get the autogenerated password that meets the account password policy.
  Custom password:- Each user will get the password you type in the textbox. Make sure your password meets the password policy.

Require password reset:- By using this option users are forced to change their password the first time they sign in. It is not mandatory to use this option but recommended one for best practices.



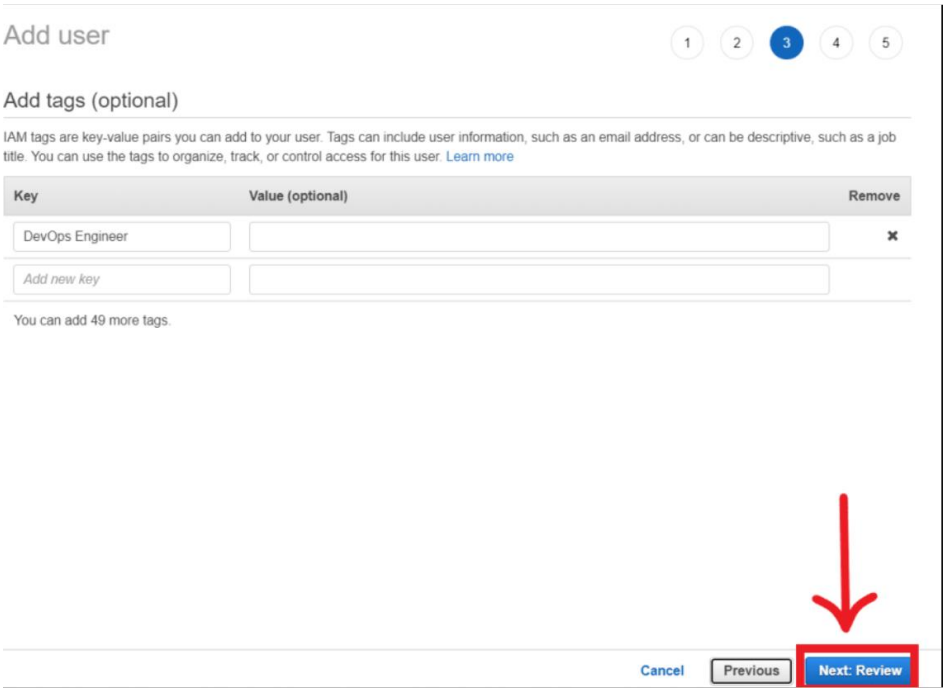
After completing all the required steps click on the **Next: Permissions** button.

Step 6:- Now you need to set permissions for your users. Choose one of the following three options:
a.) Add user to the group:- If you have created any group and you want to add your user to the specific group you can choose this option. You can select one or more existing groups. *If you don't know how to create the group don't worry I will discuss it later.*
b.) Copy permissions from existing user:- Choose this option to copy all the existing permissions boundaries and policies from an existing user to the new user.
c.) Attach existing policies directly:- AWS has a list of a large number of policies. Select the policies that you want to attach to the new users. You can also create your own custom policy.



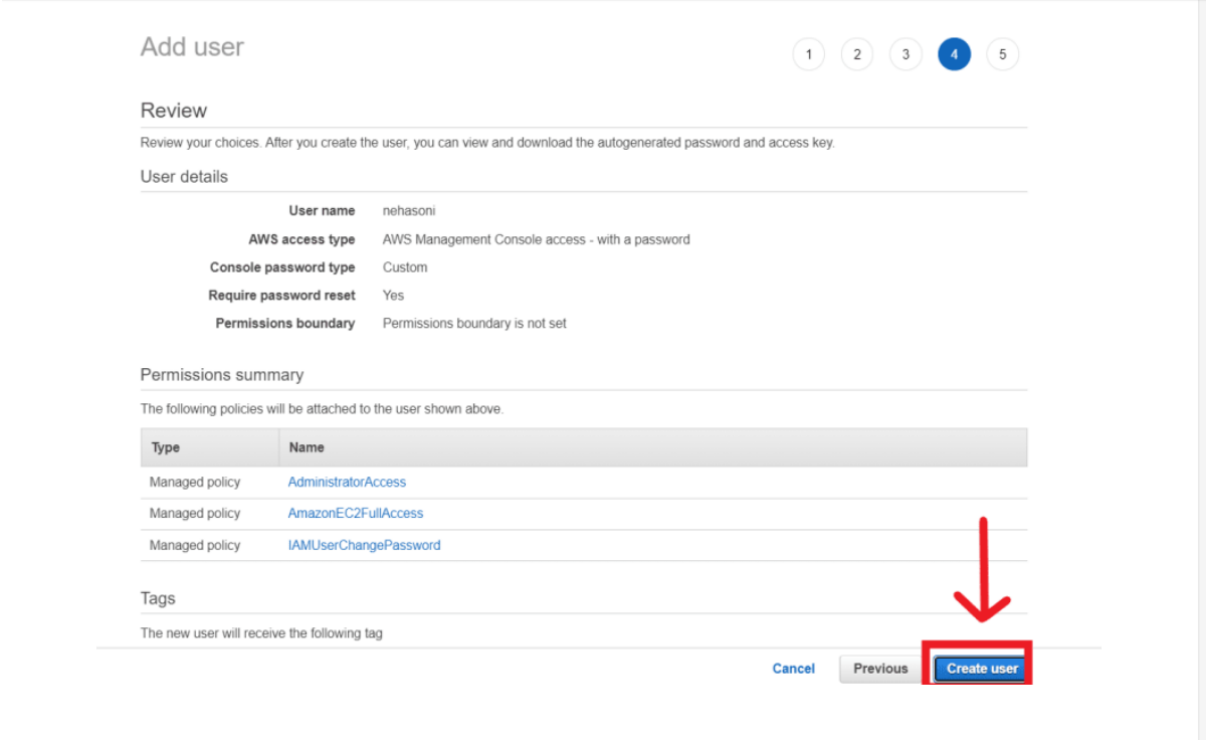
For now, I am going to use the third option i.e Attach existing policies directly. After setting the permissions click on the **Next: Tags** button.

Step 7:- Tags come in handy when we need to find a certain person in a huge group. This is a totally optional step you can skip it if you want.

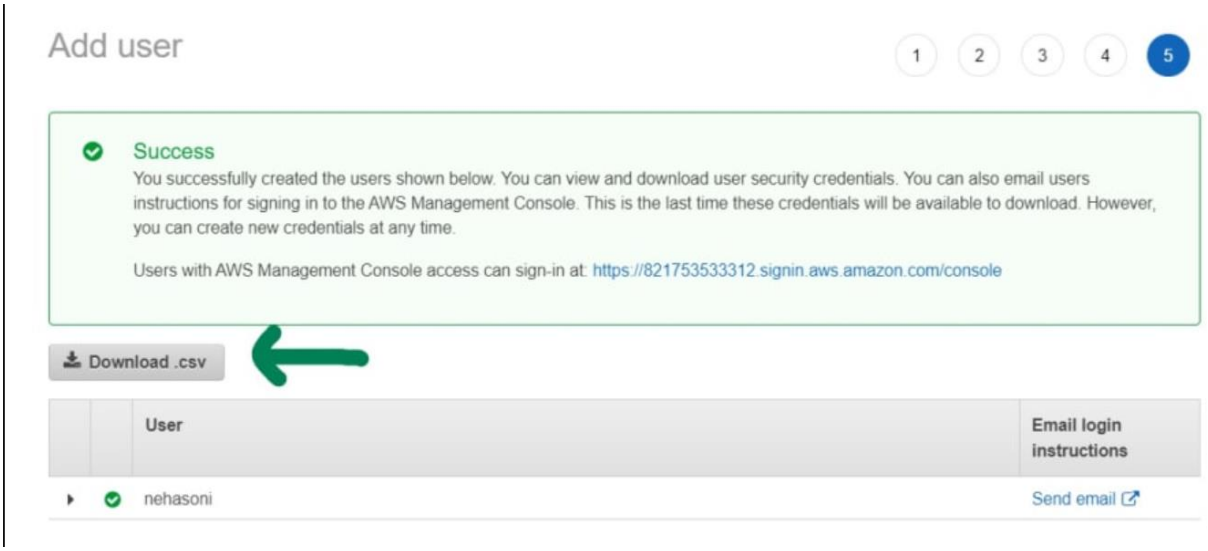


Now, Click on the **Next: Review** button.

Step 8:- On this page you will see all of the choices you made up to this point. After reviewing all the options click on the **Create User** button.

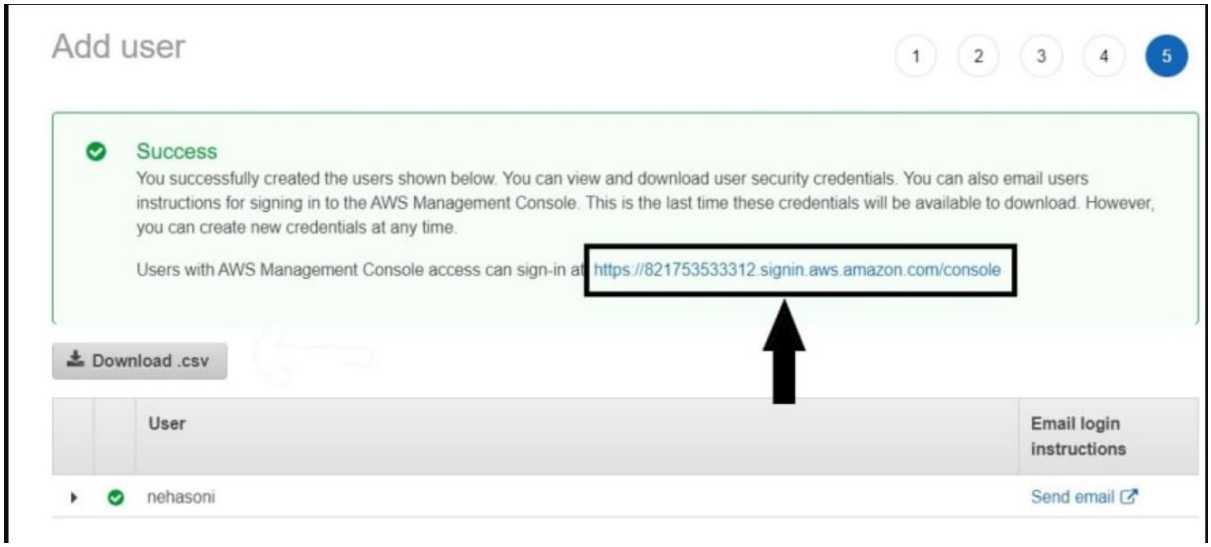


Congratulations you have created an IAM user. To save the access keys, choose Download .csv and then save the file to a safe location.



How to Login as an IAM user?

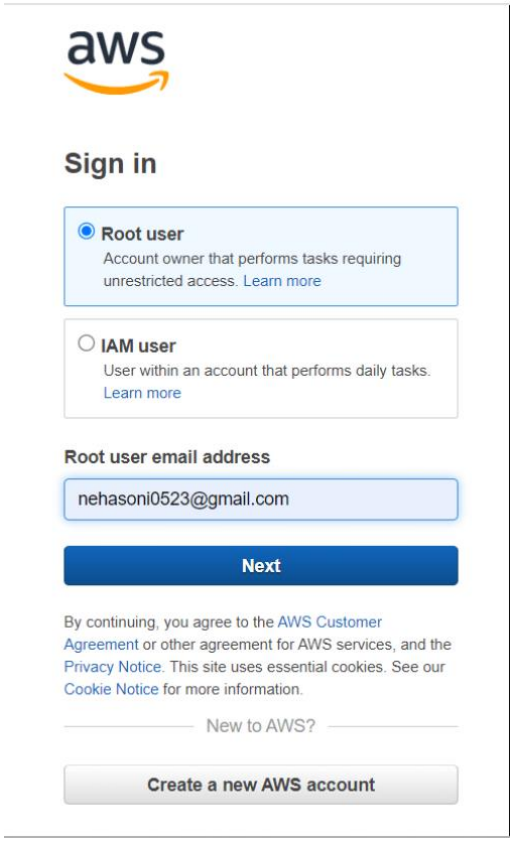
Step 1:- Copy the link and use it to login in an incognito mode as an IAM User.



Step 2:- Enter the username and password and click sign in.
And you can see that you can access EC2 as you've provided the permission while creating that user.

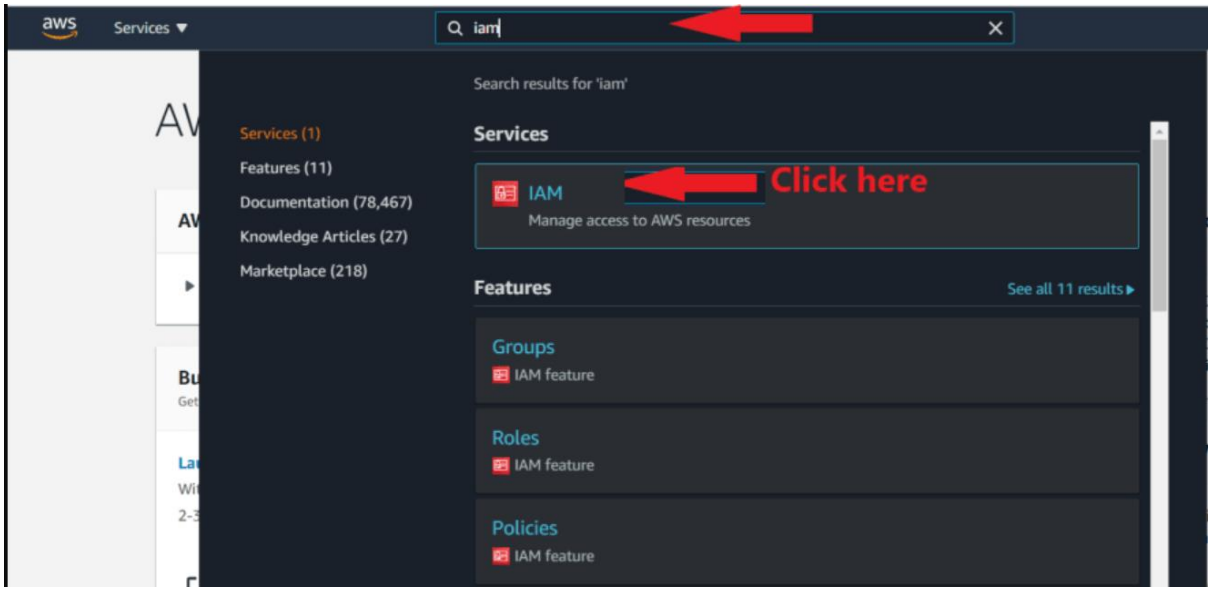
IAM User Groups

We've discussed what is an IAM User and how to create an IAM User. Let's learn how to create IAM user groups and how to add users to groups.
Step 1:- Open your favourite browser and navigate to [AWS Login Page]

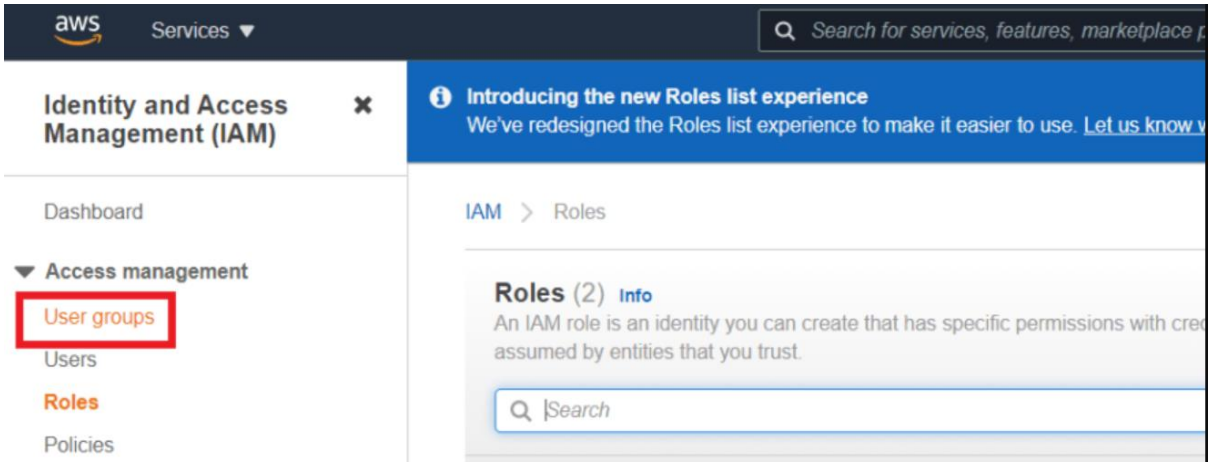


Then enter in your password and click submit. You have now successfully signed in to the AWS Management Console.

Step 2:- In the search bar type IAM and click on IAM(Manage access to AWS resources) to navigate to IAM Dashboard.

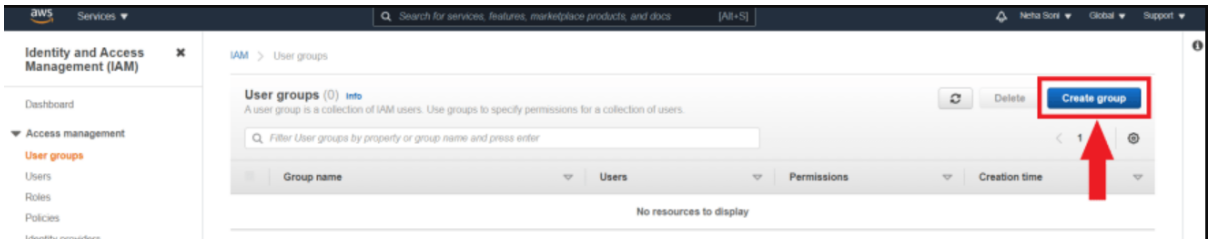


Step 3:- On the left side of the page, you should see an option called **Users groups**.

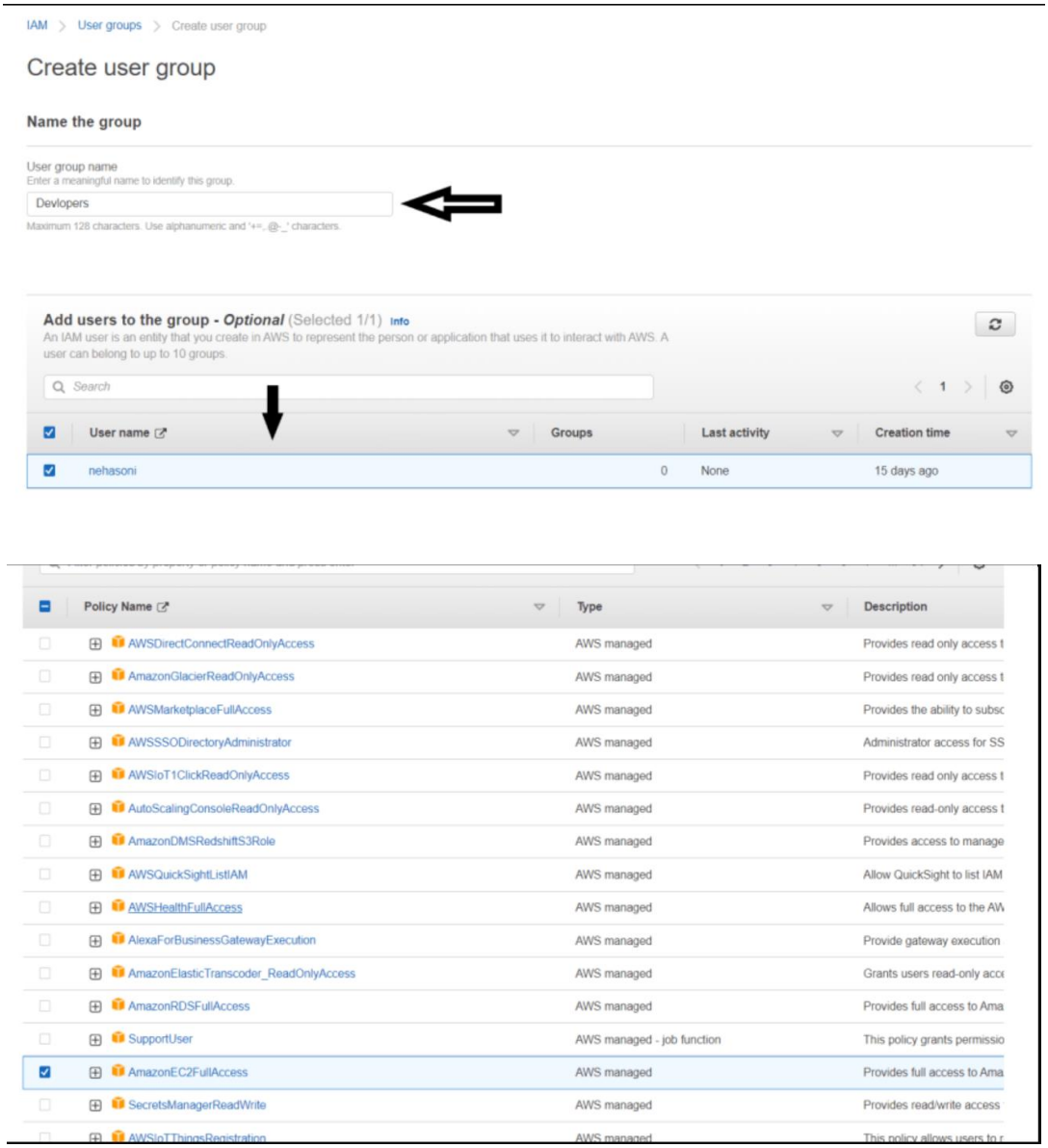


Click on that option, and you will be taken to the Users groups Page

Step 4:- Click on **Create group** button to create a new group.

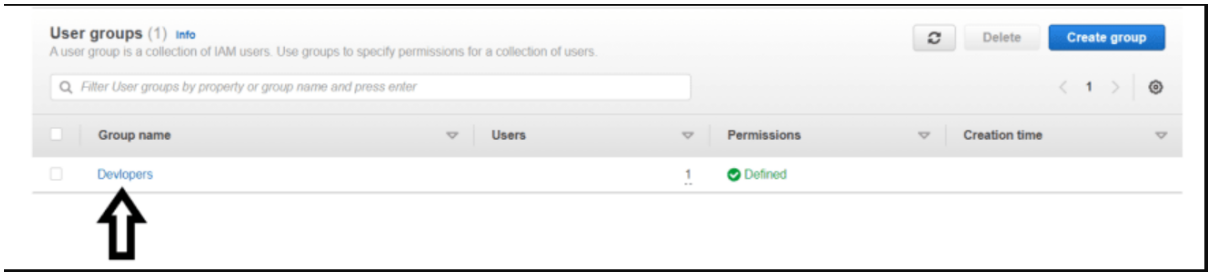


Step 5:-
Provide a **Group name**
Select the **Users** who needed to be a part of the group.
Provide any permissions from existing policies



Scroll down and click on **Create group** button.

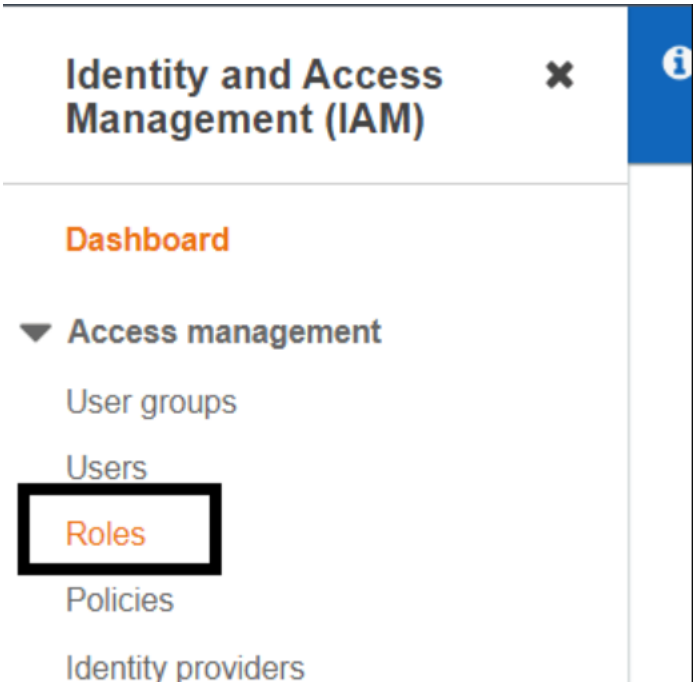
Congratulations you have created an IAM User group



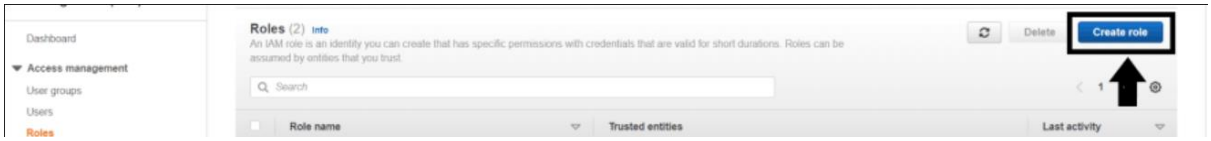
How to create IAM roles for a service:

Step 1:- Sign in to AWS Management Console. In the search bar type IAM and click on IAM(Manage access to AWS resources) to navigate to IAM Dashboard.

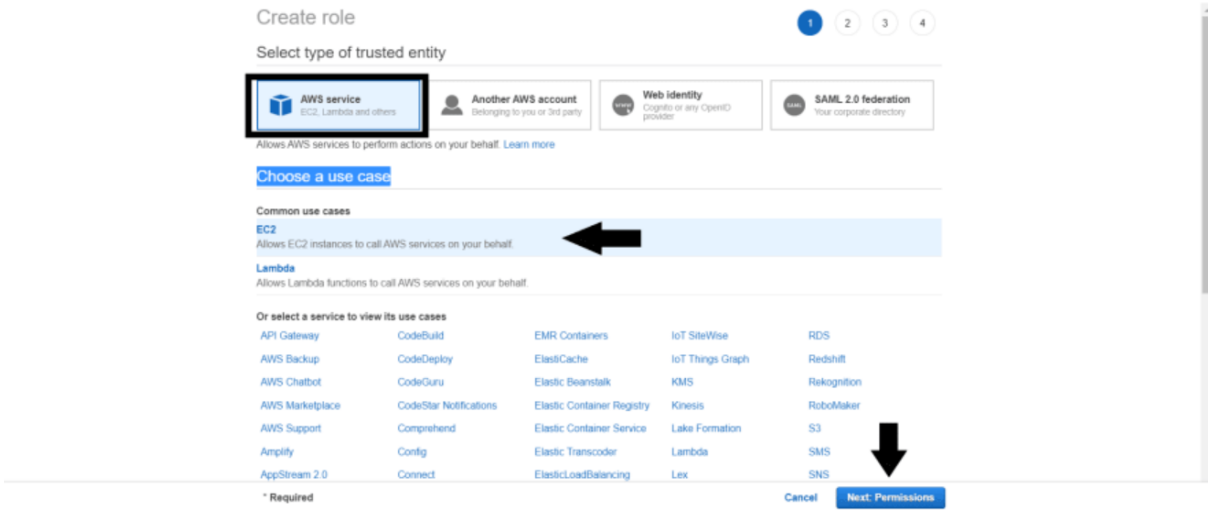
Step 2:- On the left side of the page, you should see an option called **Users groups**.



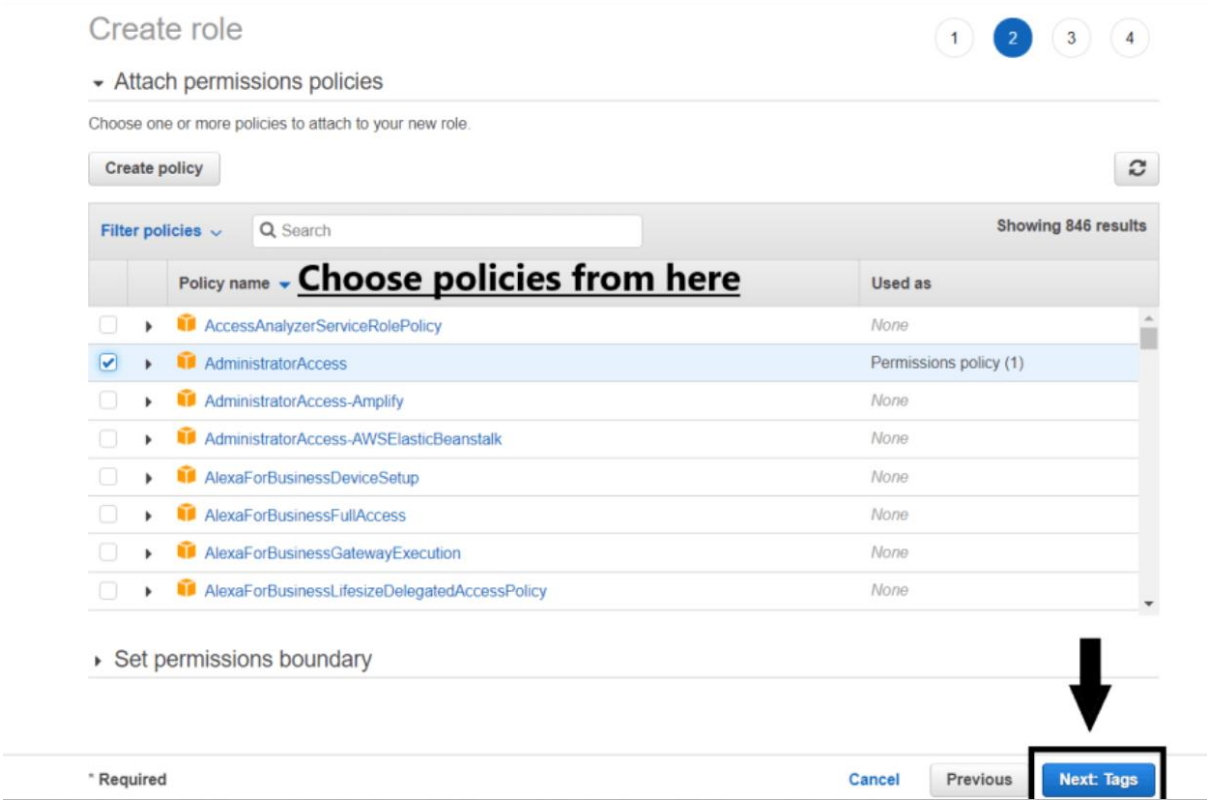
Step 3:-Click on **Create role** button to create a new role.



Step 4:- Choose the AWS service that you want to use with the role.

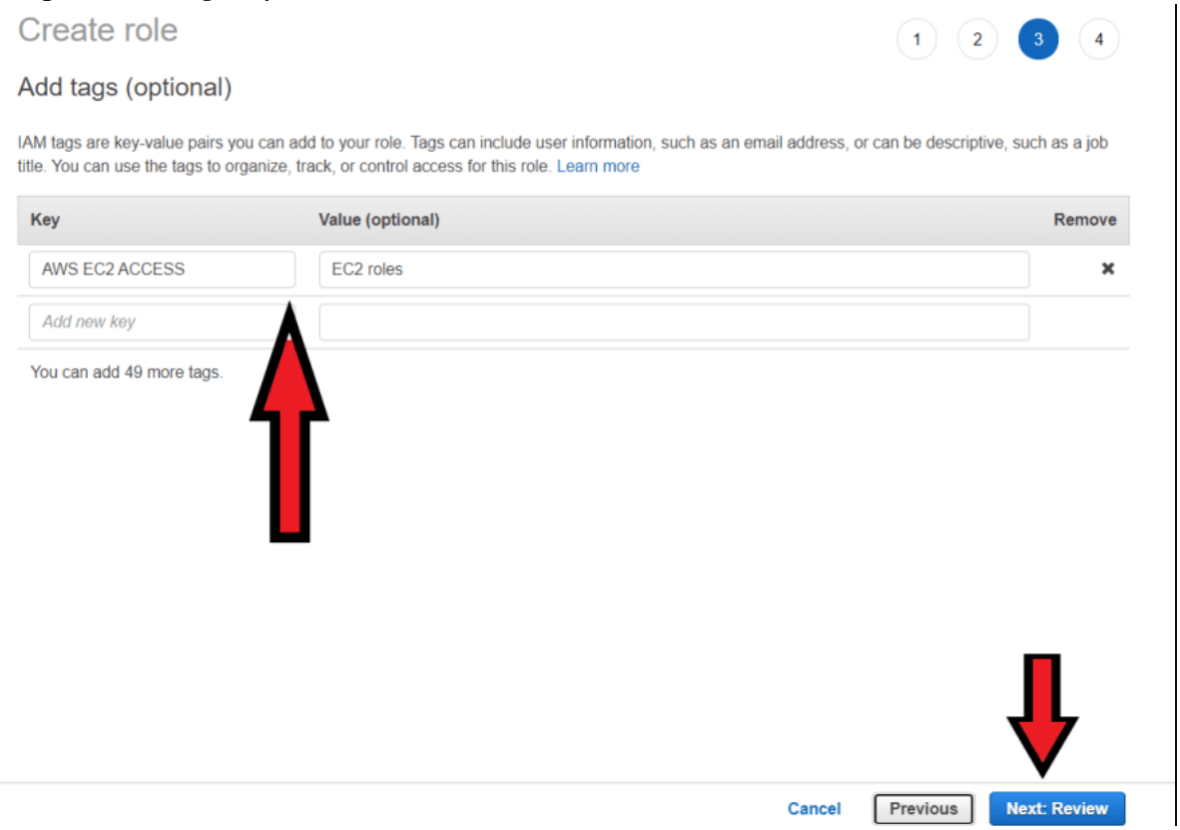


Step 5:- Provide any permissions from existing policies or you can also attach custom policies.

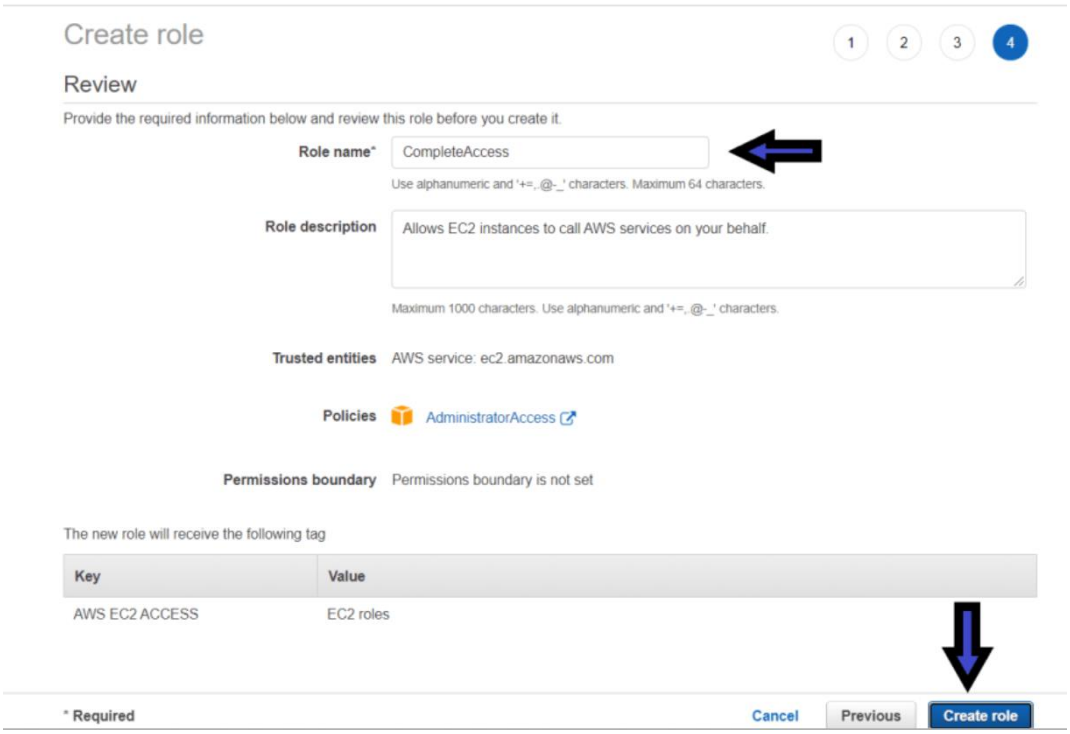


After attaching policies click on the **Next: Tags** button.

Step 6:- Add tags if you want and click on **Next: Review** button.



Step 8:- In a role name box, enter the role name. After completing all the required steps click on **Create role** button.

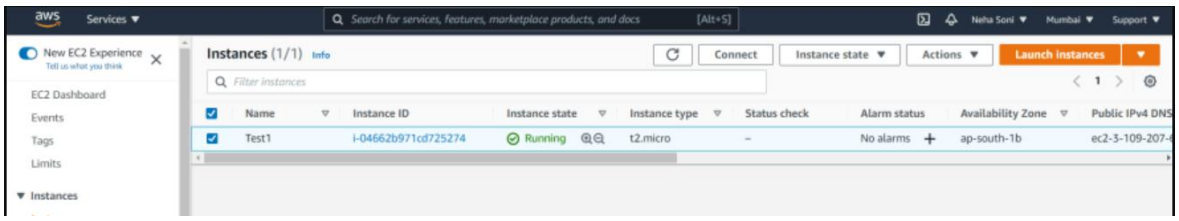


Congratulations you have created an IAM role.

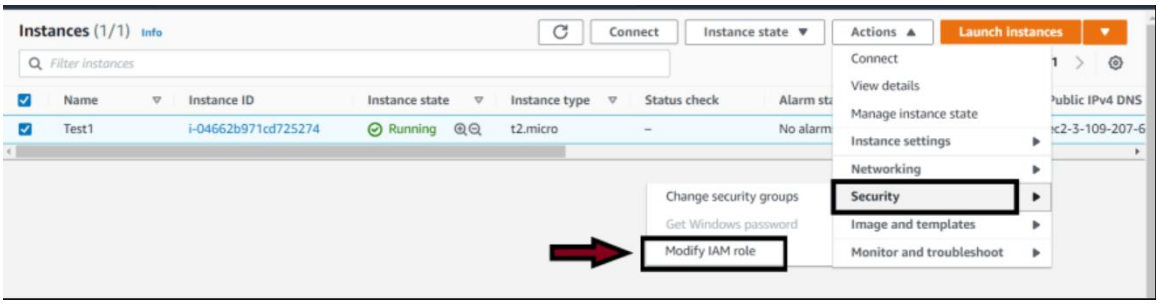
Now how to attach it to any AWS service:

Let's attach it to AWS EC2 instance: -

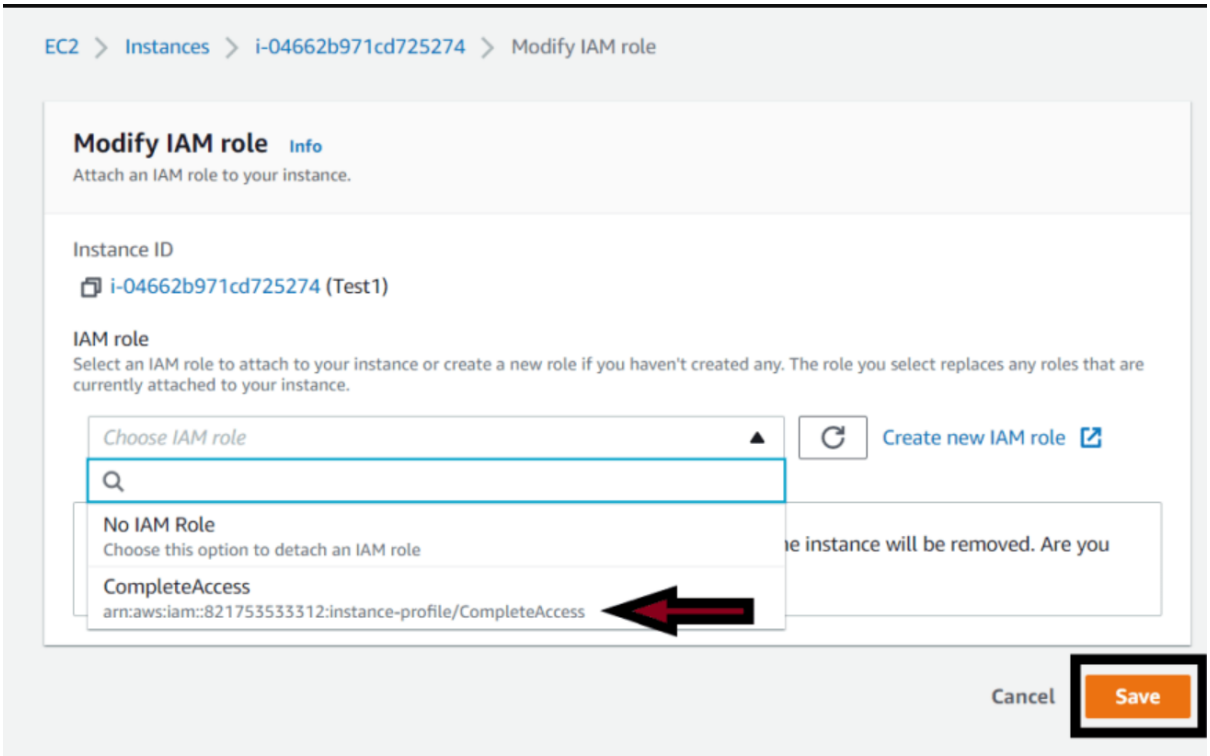
Step 1:- **First create an AWS EC2 instance.**
If you don't know how to create an EC2 instance, I recommend you to checkout this article once -> [How to create an AWS EC2 instance?]



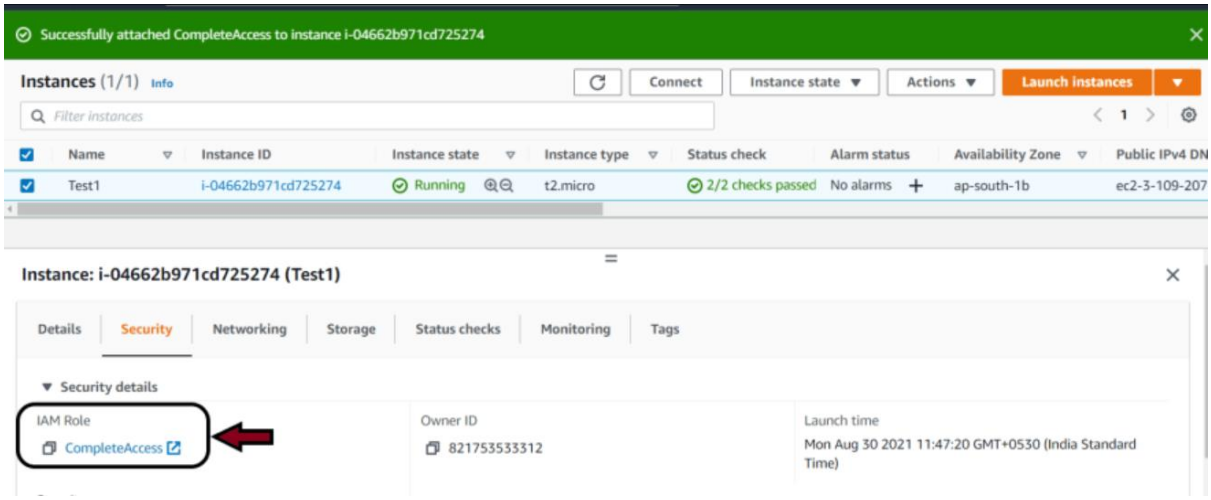
Select the "EC2 Instance" as we did by selecting our "Testing Instance",
Step 2:- **Click on "Actions" button and from the "drop-down menu" go to the** "Security" **again a new menu will be shown, from there look for "Modify IAM Role", and select it.**



Step 3:- **Select the** IAM role **from the drop-down options and then hit** Save **button.**



To confirm the IAM role attachment, select you EC2 instance and check the **security** tab, you will see the **IAM role** has been successfully attached.



Explain the features and use cases of each service, such as Amazon EC2, Amazon S3, Amazon RDS, Amazon VPC, and AWS IAM.

Amazon EC2 (Elastic Compute Cloud):

Features:

Provides resizable compute capacity in the cloud with virtual machines known as instances.

Offers a wide range of instance types with varying CPU, memory, storage, and networking capabilities.

Supports multiple operating systems and allows custom AMIs (Amazon Machine Images) for instance configurations.

Use Cases:

Hosting web applications and websites.

Running development and testing environments.

Handling batch processing workloads.

Running high-performance computing (HPC) applications.

Amazon S3 (Simple Storage Service):

Features:

Provides scalable object storage with high durability and availability.

Supports data encryption at rest and in transit for security.

Allows versioning, lifecycle management, and cross-region replication for data management.

Integrates with AWS CloudFront for content delivery and caching.

Use Cases:

Storing and serving static website content.

Backing up and archiving data.

Hosting media files and large datasets.

Storing logs and analytics data.

Amazon RDS (Relational Database Service):

Features:

Manages relational databases like MySQL, PostgreSQL, Oracle, SQL Server, and MariaDB.

Offers automated backups, software patching, and scaling capabilities.

Supports read replicas for read-heavy workloads and Multi-AZ deployments for high availability.

Provides encryption, monitoring, and performance optimization tools.

Use Cases:

Hosting production databases for applications.

Setting up dev/test environments with managed databases.

Implementing disaster recovery solutions.

Running reporting and analytics workloads.

Amazon VPC (Virtual Private Cloud):

Features:

Creates isolated virtual networks within AWS.

Allows customization of IP address ranges, subnets, routing tables, and network gateways.

Offers security features like network ACLs (Access Control Lists) and security groups.

Supports VPN connections and direct connections for hybrid cloud setups.

Use Cases:

Hosting multi-tier applications with private and public subnets.

Establishing secure communication between AWS resources.

Extending on-premises networks to the cloud.

Creating isolated environments for testing and development.

AWS IAM (Identity and Access Management):

Features:

Manages user identities, permissions, and access to AWS resources.

Provides granular access control with policies, roles, and groups.

Supports multi-factor authentication (MFA) for enhanced security.

Offers integration with AWS services for fine-grained permissions.

Use Cases:

Managing user access to AWS resources based on roles and responsibilities.

Implementing least privilege principles for security.

Enforcing security policies and compliance requirements.

Auditing and monitoring user activity and resource access.

These core AWS services form the foundation of cloud computing on AWS, offering a comprehensive set of tools and capabilities for building scalable, secure, and reliable cloud-based applications and infrastructure.

LAMBDA SERVICE:

Lambda is designed to enable developers to run code without provisioning or managing servers.

LAMBDA FUNCTION:

It is a serverless, event-driven compute service that let us to run code for virtually any type of application or backend services.

We can trigger Lambda from over 200 AWS services and SaaS applications.

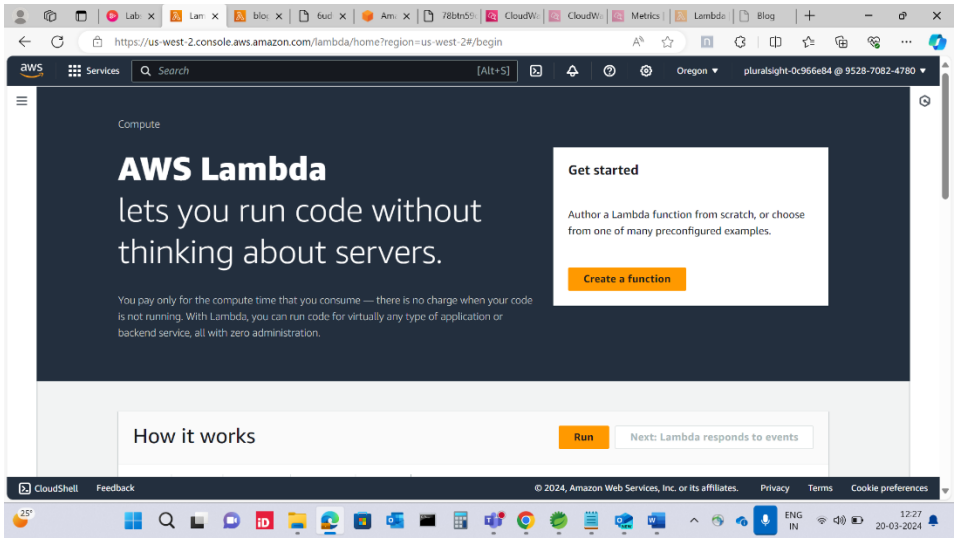
AWS MONITORING:

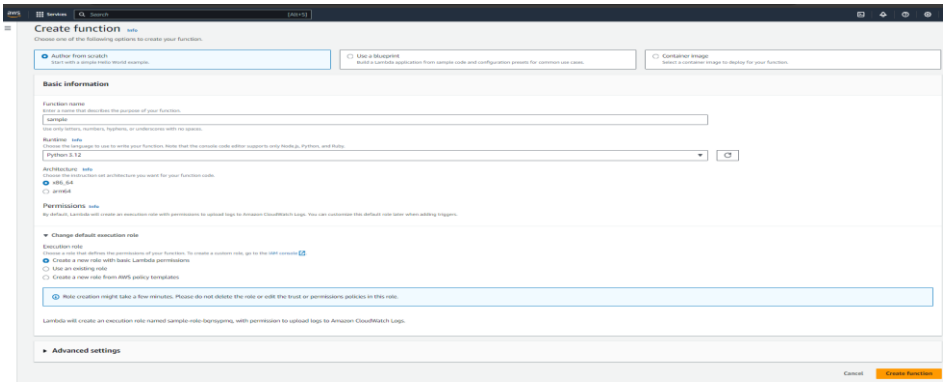
It scans our AWS resources and applications, collecting data to ensure everything is operating smoothly and securely.

Monitoring helps to identify the vulnerabilities and issues, predict performance, and optimize configurations.

STEP 2: Creating Lambda Function

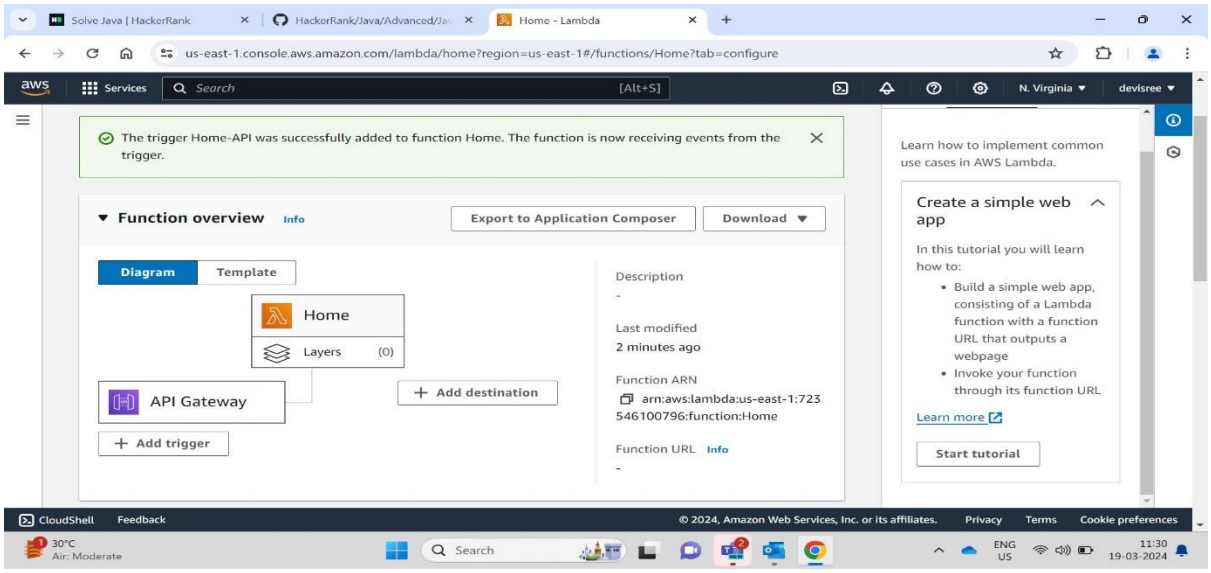
1. We need to choose author from scratch option from the three other options.
2. Click on create function.
3. As per our need, here we're creating 2 Lambda functions.
4. Give name for function, as per our req. we chose (Home Page & Blog) as function name.
5. We need to select the programming language, here we chose "PYTHON" in the runtime tab.
6. Change the default execution role option, choose any one option as per our need.
7. In Advanced option, if necessary, choose options according to the project.
8. Click on create function, the function will create.

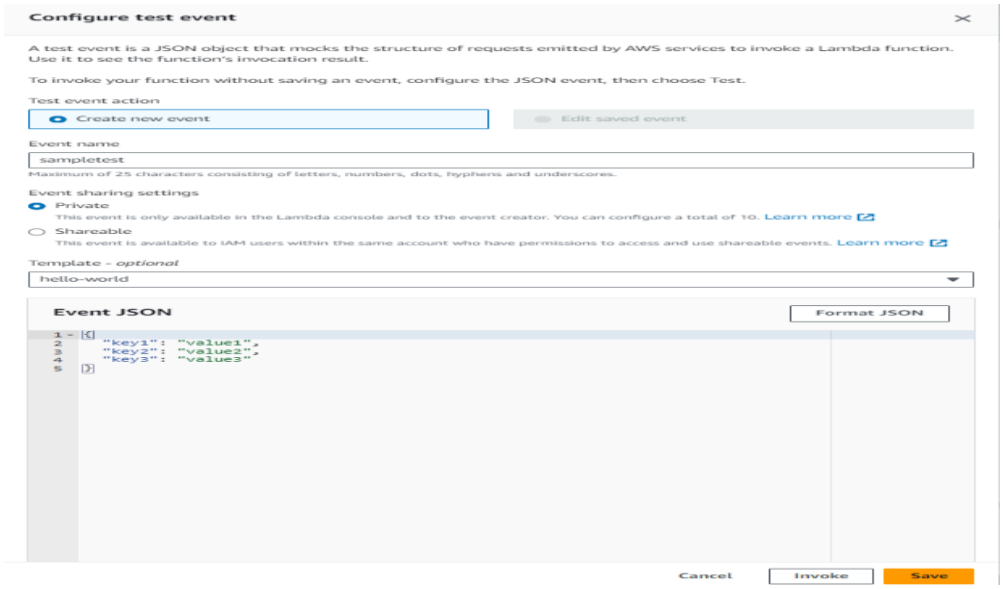




STEP 3: Compiling & Executing Code

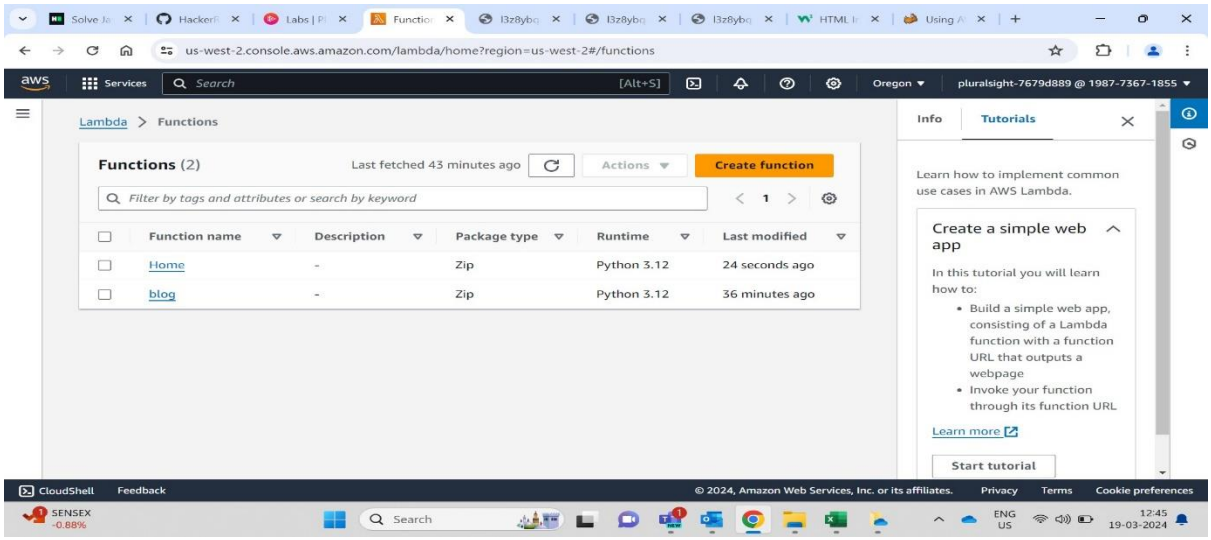
1. Write the code in the code section.
2. After that, click on “Deploy” tab to save the code. (i.e., every time, if we made any changes in code, it is mandatory to deploy the code, to save/update it)
3. After deployed, click “Test” tab to compile the code.
4. Once u clicked the “Test” tab, it asked to create the “Test Event”.
5. Click on new test event and name that event.
6. If u need any changes, made it and save that.
7. After that, click on “Test” to execute the code, it will show the response of the code.
8. If any changes u needs to do in code, after deploying only it will execute the changed code or else it will execute the unchanged code.





STEP 4: Adding Trigger

1. After the execution of the code, we need to add the trigger.
2. Click on add trigger, select the source (as per need, here we chose API GATEWAY) from the drop-down box.
3. Select Intent new / existing one.
4. Select the API type and security.
5. Click on add and it will create the trigger.



STEP 6: Creating and linking the subpages.

1. Follow the above instructions from step 1 – step 5, except in step 4, choose intent as “Existing one” and select the “home page”.
2. After all the steps, copy the link which is created for sub page and paste it in the subpage code (in <a href> tab).
3. Before the configuration tab, “Monitoring” tab is available, we can use that to monitor the created webpages.
4. STEP 5: Opening the website.
5. the creation of the trigger, a link is generated in the "Configuration" tab.
6. This link directs to our lambda function (i.e., Upon homepage) and is securely protected.
7. To access the website page, simply click on the provided link.