

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 centers around the world, so even if one goes down, your data and services remain accessible.

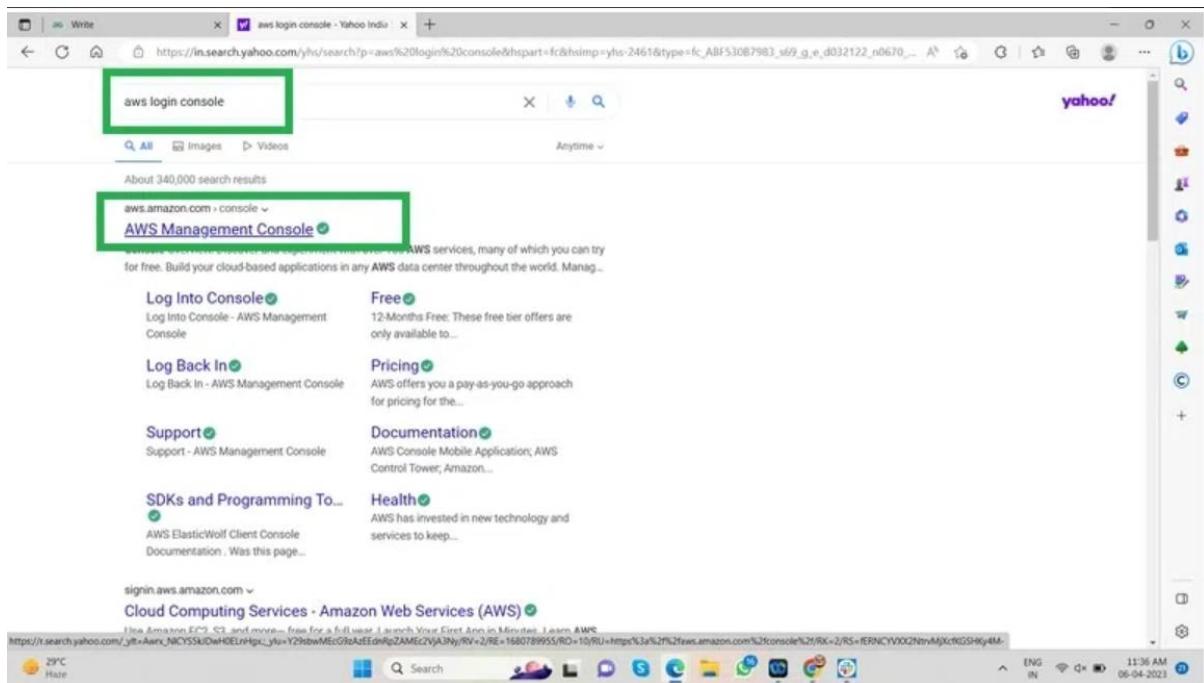
Cost-effectiveness: With AWS, you only pay for what you use. There are no upfront costs, and you can adjust your resources as needed, which helps you save money compared to maintaining your own hardware.

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.

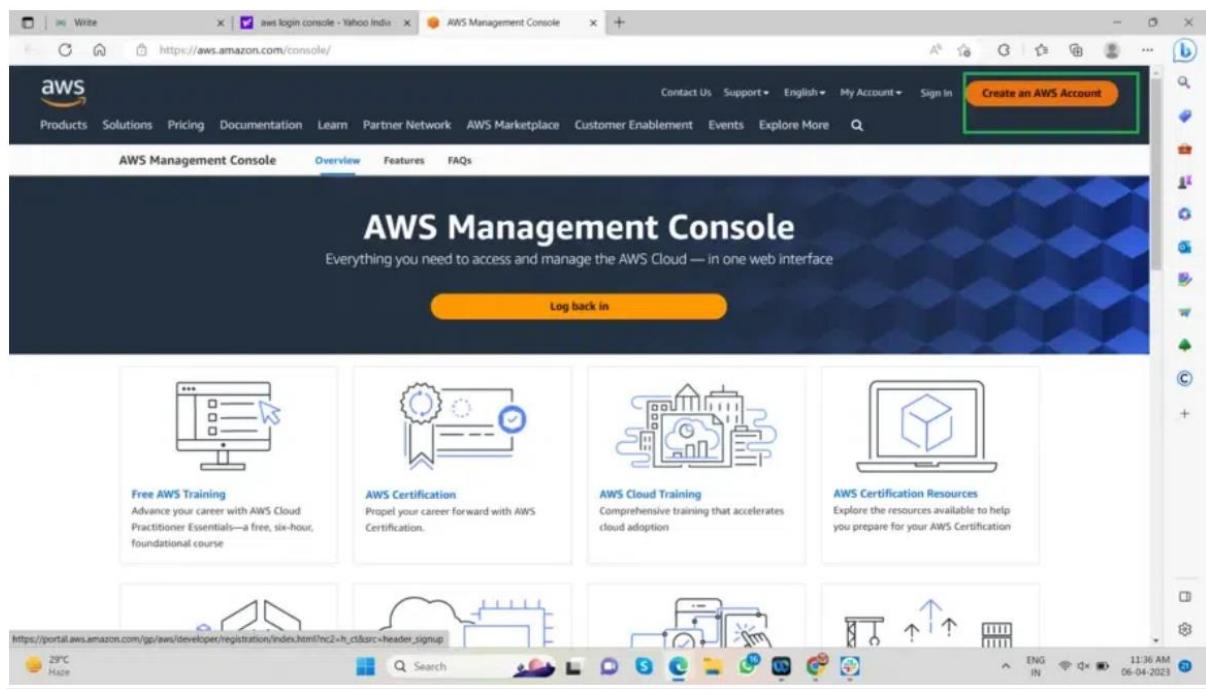
How Creating and Log-in to AWS account.

Creating an AWS Account:

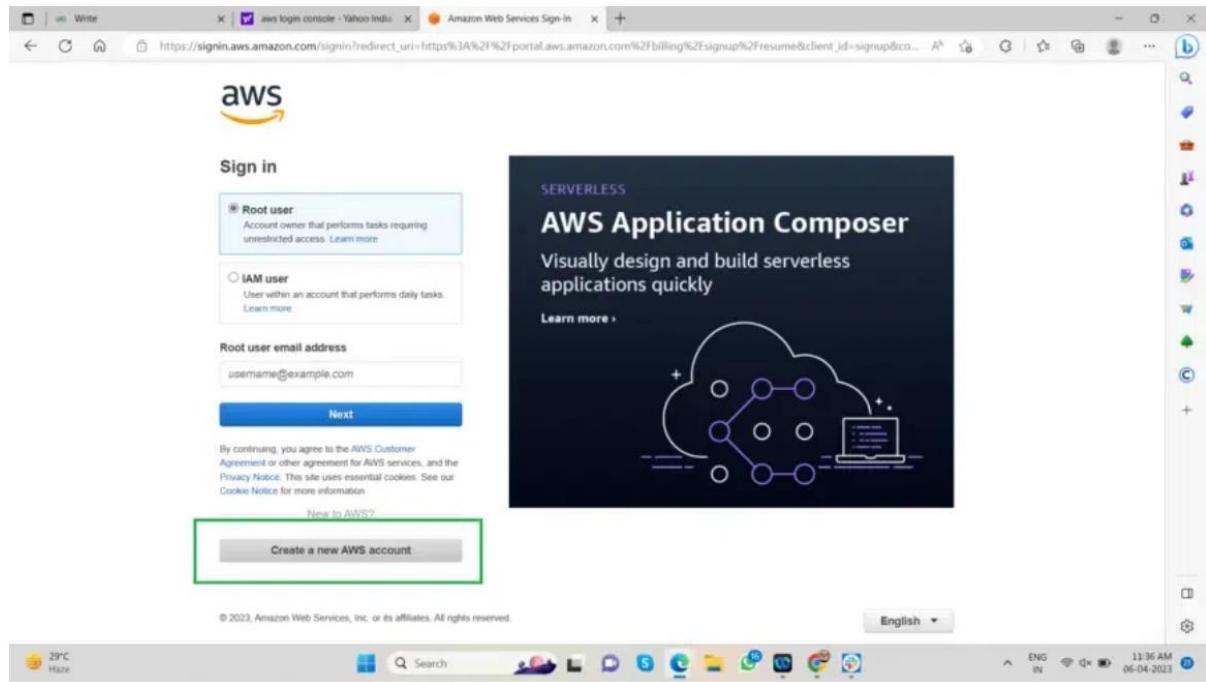
Step 1: First Open your web browser and search for AWS Login Console and click on the first link. As shown in the picture below



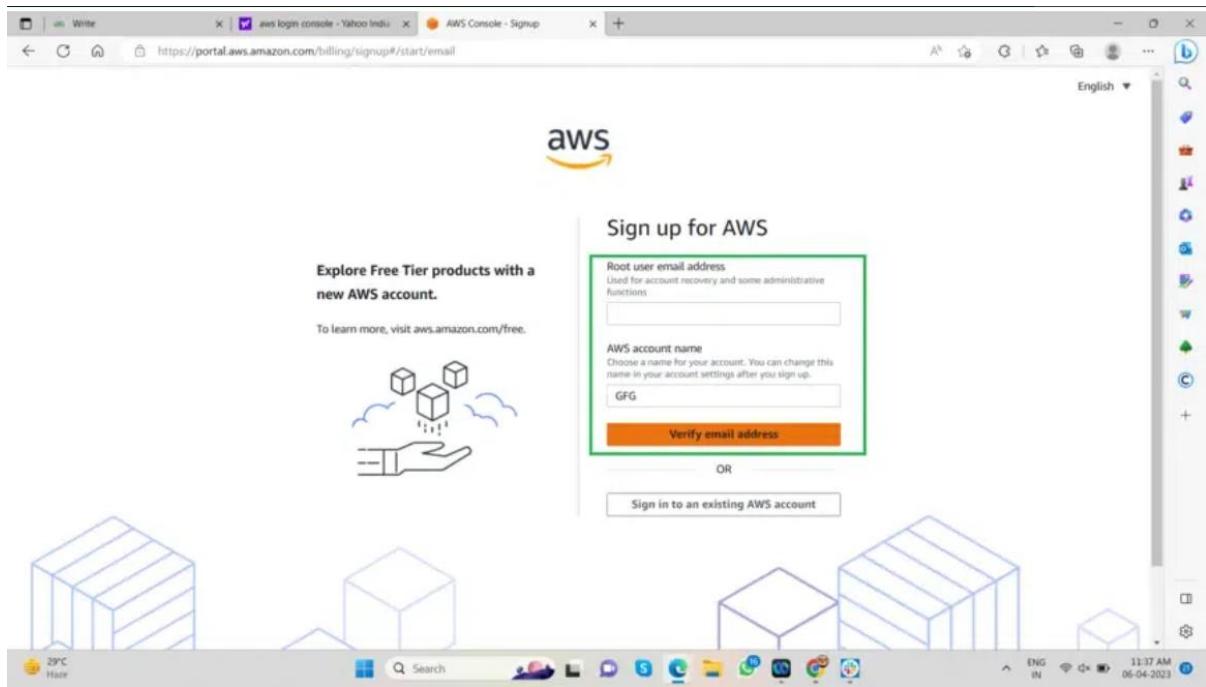
Step 2: An AWS Login Console page will open now click on Create an AWS Account.



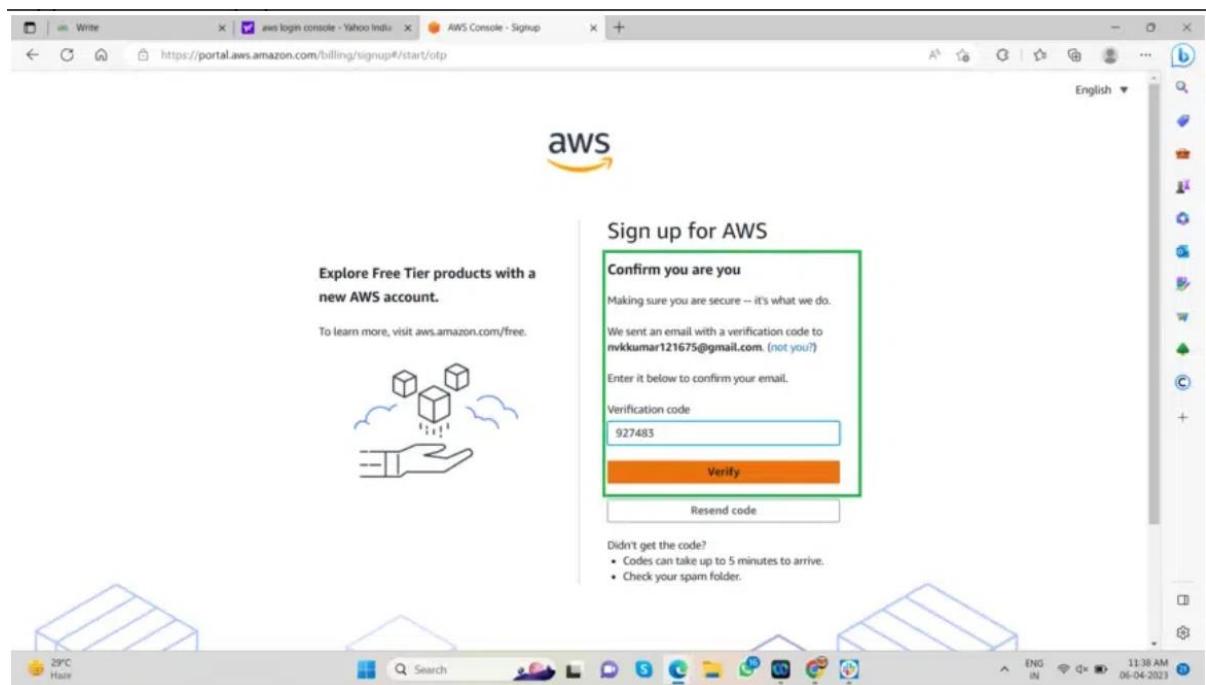
Step 3: A new AWS sign-in page will now open after selecting Create an AWS Account. Choose to Create a new AWS account. As shown in the image below



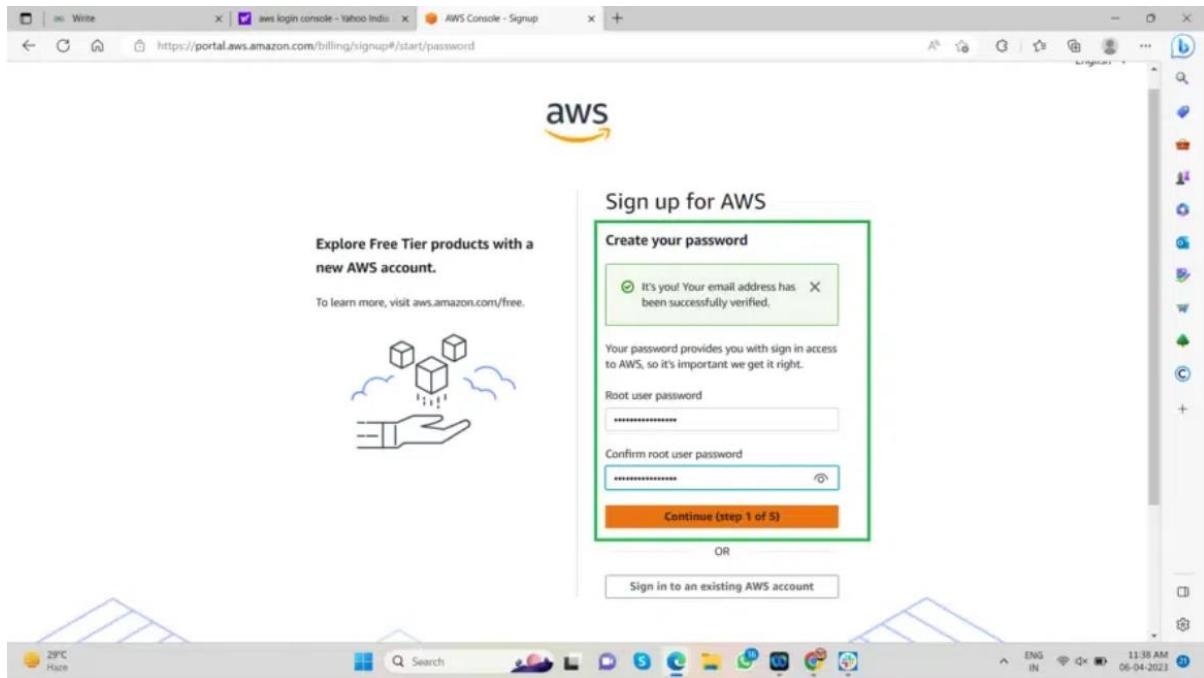
Step 4: In order to use the feature to log into an AWS Free Tire Account, we must validate the email address and have to provide the AWS account name in this stage. After clicking on “Verify Email Address,” you will receive a verification code at the address you provided. Next, you must create a password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



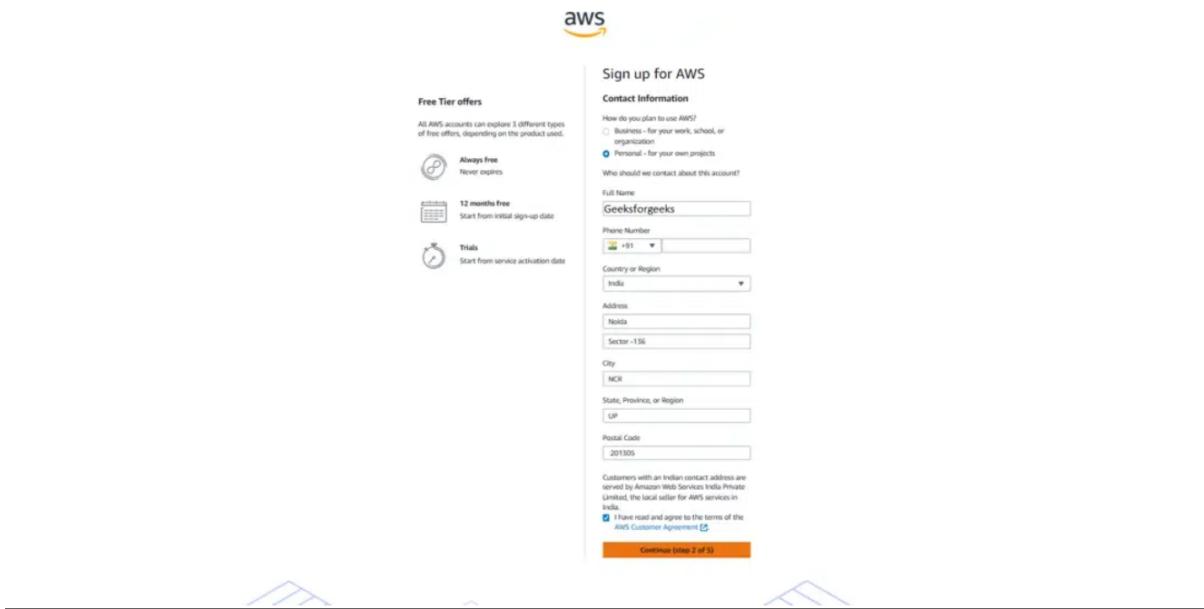
After clicking on “Verify Email Address,” you will receive a verification code at the address you provided.



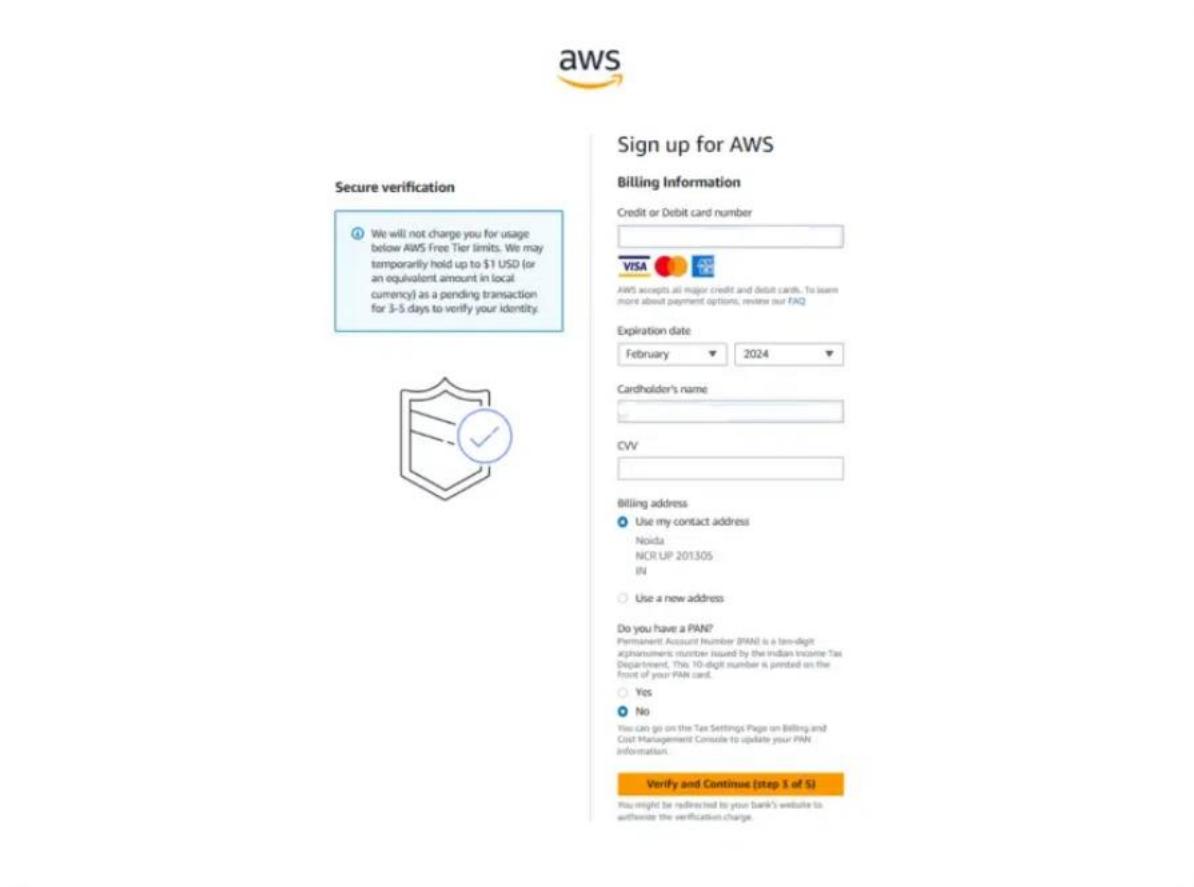
Next, you must create a password for this account. Finally, click “Continue” to move on to the next stage.



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.



The screenshot shows the AWS sign-up process at step 7, specifically the secure verification stage. The top right corner features the AWS logo. Below it, the heading "Sign up for AWS" is displayed. On the left, there's a "Secure verification" section containing a note about temporary hold charges and a shield icon with a checkmark. The main form area is titled "Billing Information". It includes fields for "Credit or Debit card number" (with VISA, MasterCard, and American Express icons), "Expiration date" (February 2024), "Cardholder's name" (empty field), and "CVV" (empty field). Under "Billing address", the "Use my contact address" option is selected, showing details for Noida, NCR UP 201305, IN. There's also an option to "Use a new address". A "Do you have a PAN?" section follows, with the "No" option selected. It provides instructions for updating PAN information via the Tax Settings page. At the bottom right is a yellow "Verify and Continue (step 3 of 5)" button.

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.

Sign up for AWS

Confirm your identity

Before you can use your AWS account, you must verify your phone number. When you continue, the AWS automated system will contact you with a verification code.

How should we send you the verification code?

Text message (SMS)

Voice call

Country or region code

India (+91)

Mobile phone number

Security check

b5 c8x6	n3 roxp
4	5
6	7

Type the characters as shown above

b5c8x6

Send SMS (step 4 of 5)

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.

Sign up for AWS

Confirm your identity

Verify code

1

Continue (step 4 of 5)

Having trouble? Sometimes it takes up to 10 minutes to retrieve a verification code. If it's been longer than that, return to the previous page and try again.

Privacy Policy | Terms of Use | Cookie Preferences | Sign Out

Amazon Web Services, Inc. or its affiliates. All rights reserved.

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

The screenshot shows the 'Sign up for AWS' page. At the top, there's an AWS logo and a heading 'Sign up for AWS'. Below it, a section titled 'Select a support plan' with the sub-instruction 'Choose a support plan for your business or personal account. Compare plans and pricing examples'. A note says 'You can change your plan anytime in the AWS Management Console.' Three support plan options are listed in boxes:

- Basic support - Free** (radio button selected, highlighted with a green border):
 - Recommended for new users just getting started with AWS
 - 24x7 self-service access to AWS resources
 - For account and billing issues only
 - Access to Personal Health Dashboard & Trusted Advisor
- Developer support - From \$29/month** (radio button):
 - Recommended for developers experimenting with AWS
 - Email access to AWS Support during business hours
 - 12 (business)-hour response times
- Business support - From \$100/month** (radio button):
 - Recommended for running production workloads on AWS
 - 24x7 tech support via email, phone, and chat
 - 1-hour response times
 - Full set of Trusted Advisor best-practice recommendations

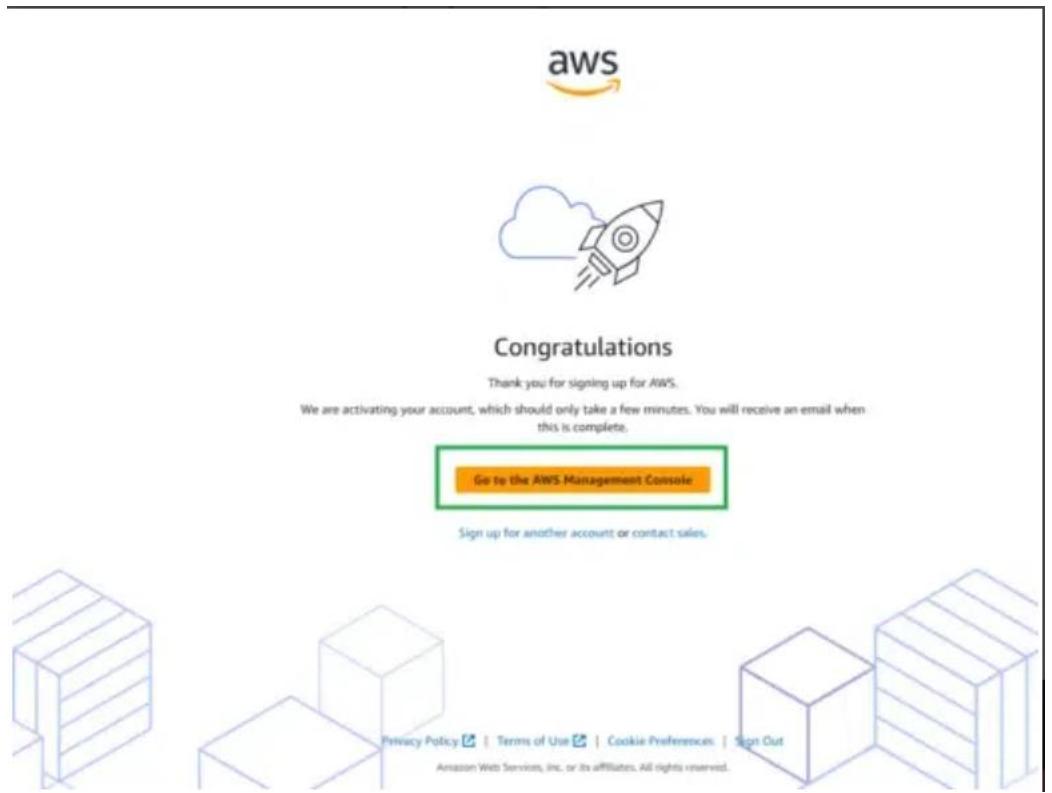
Below the plans, there's a section for 'Enterprise level support':

Need Enterprise level support?
From \$15,000 a month you will receive 15-minute response times and concierge-style experience with an assigned Technical Account Manager. [Learn more](#)

A large orange button at the bottom right is labeled 'Complete sign up'.

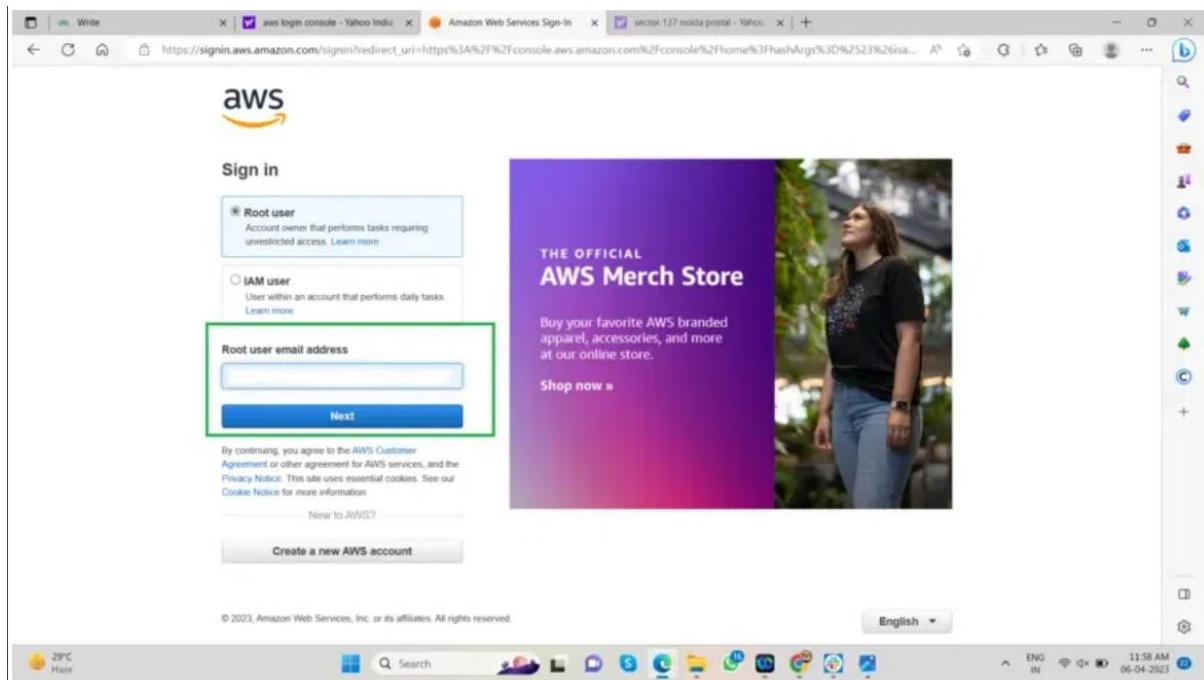
After selecting a plan, click “Complete the sign up” as shown in the image.

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.



A screenshot of the AWS welcome page. The top navigation bar includes links for Contact Us, Support, English, My Account, Sign In, and Create an AWS Account. The main content area features a purple header with the text "Welcome to Amazon Web Services" and a message about account activation. To the right is a red sidebar with a yellow "Sign In to the Console" button, a link to "Check your tax details for accurate invoicing >>>", and a "Contact Sales" button. Below the main content is a section titled "Personalize Your Experience" with a form for selecting a role and interests. At the bottom is a checkbox for newsletter subscription and a "Submit" button.

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.

The screenshot shows the AWS Console Home page. At the top, there are navigation links for 'Services' and a search bar. The main area is titled 'Console Home' with a 'Info' link. It features several sections: 'Recently visited' (listing FSx, Storage Gateway, Simple Notification Service, Elastic Kubernetes Service, AWS Auto Scaling, Support, and VPC), 'Welcome to AWS' (with links to 'Getting started with AWS', 'Training and certification', and 'What's new with AWS?'), 'AWS Health' (showing 0 open issues), and 'Cost and usage' (with a link to 'Info'). The bottom of the page includes links for 'CloudShell', 'Feedback', 'Language', and copyright information: '© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences'.

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

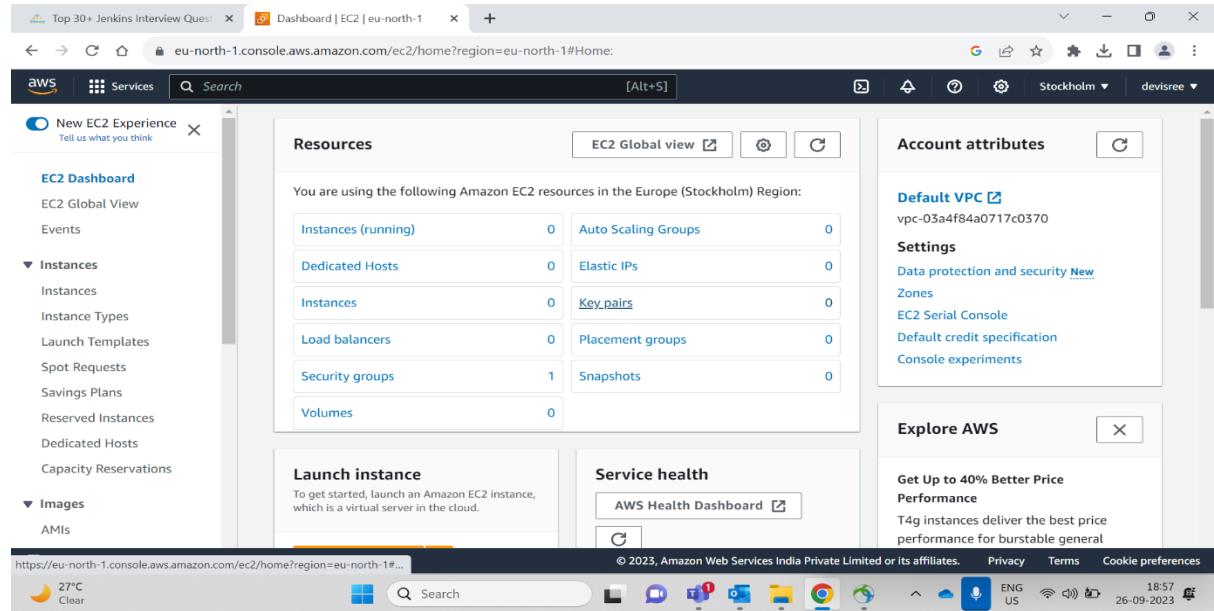
How to create EC2 instance.

Create Instance in AWS

- 1) Create the AWS Account
- 2) Login with your aws account

3) Click on Services

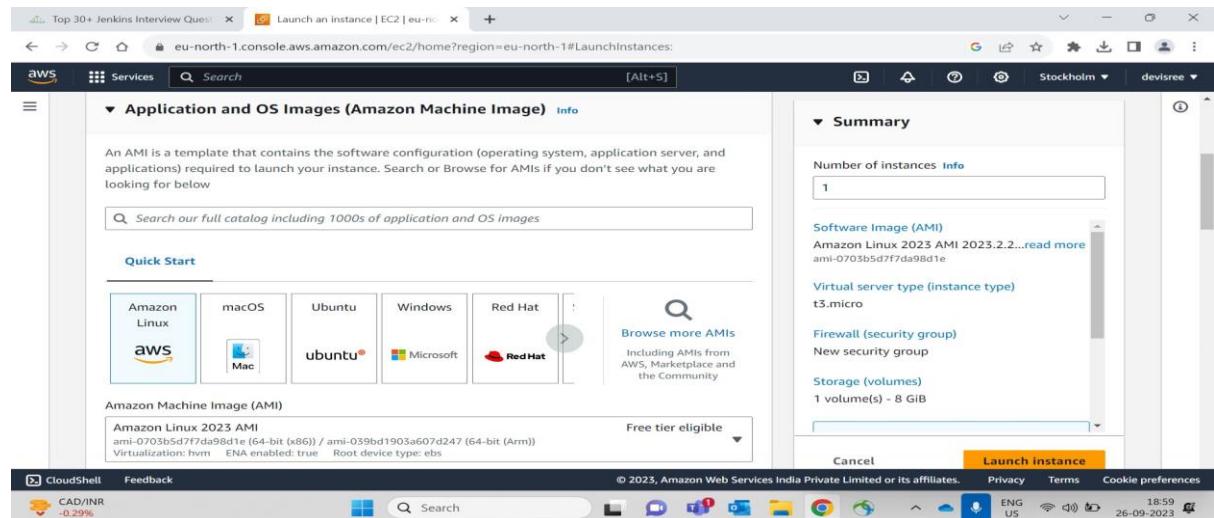
4) Click on EC2



5) Click on Instance

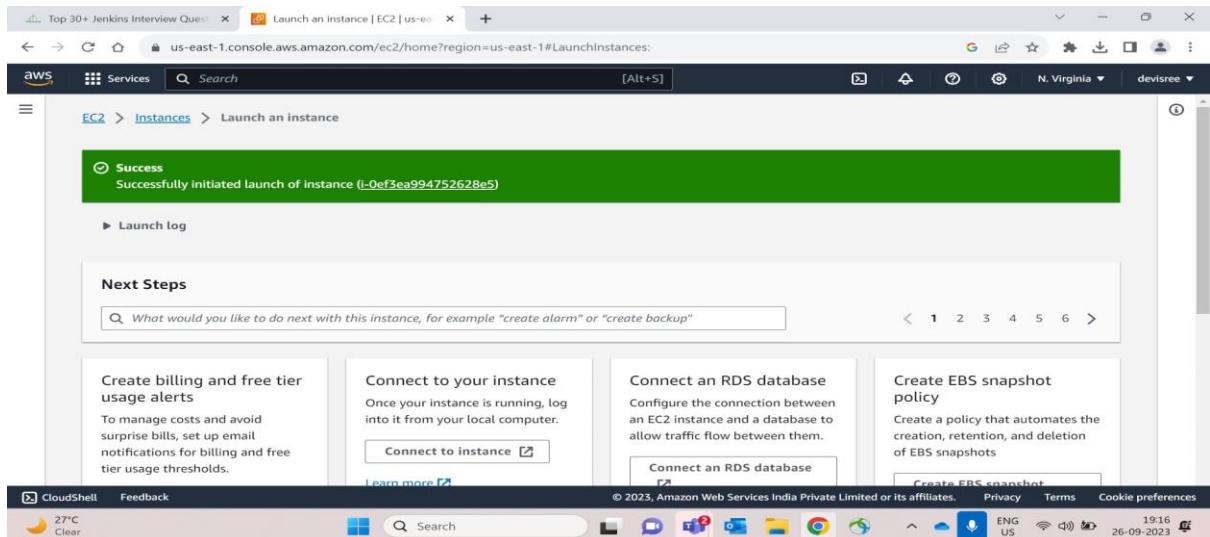
6) Click on launch instance

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 instance was created successful.

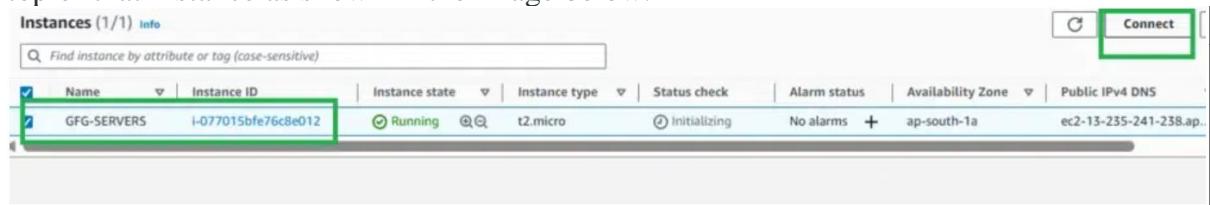
Create a Key-value pair.(A file will be downloaded)



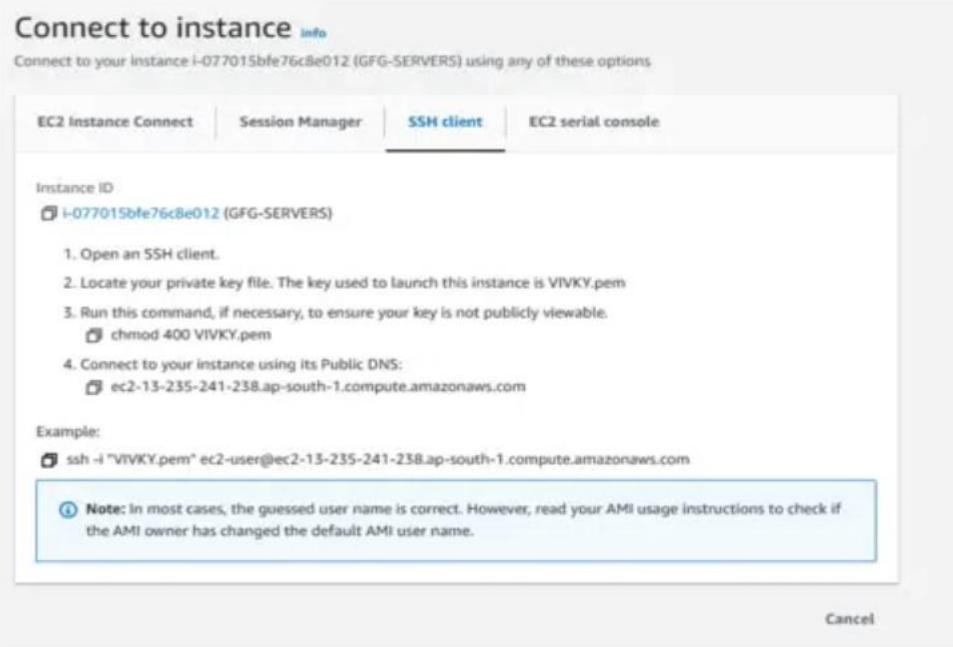
Now launch the instance as EC2 is a block to store data in AWS verify it's working after launching it successfully.

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 act 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](#).

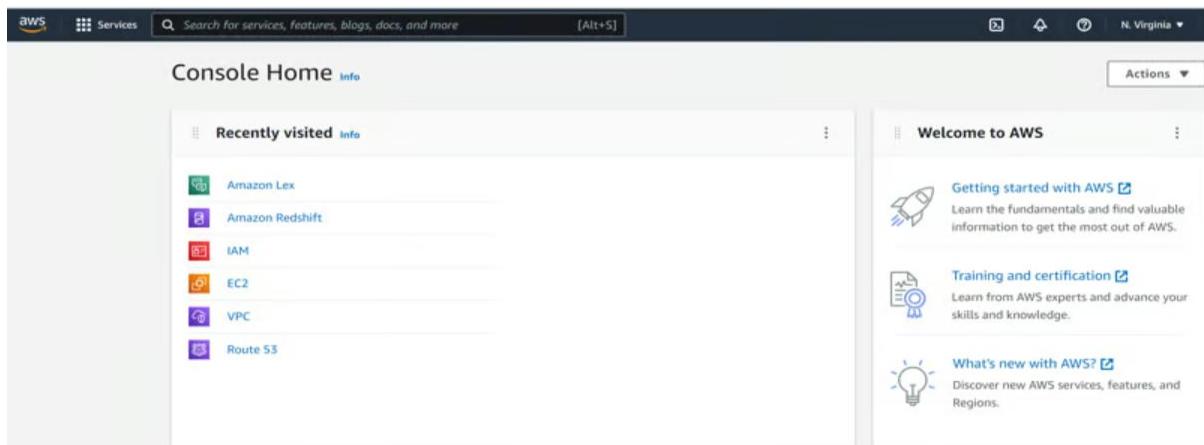
```
PS C:\Users\rknav\Downloads> ssh -i "VTVH.pem" ec2-user@ec2-13-235-241-238.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-13-235-241-238.ap-south-1.compute.amazonaws.com (13.235.241.238)' can't be established.
ED25519 key fingerprint is SHA256:5VxqQUp4UBe9rUMXvZ1uL9UnzRNfpSFK8DjMybXVoyE.
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.

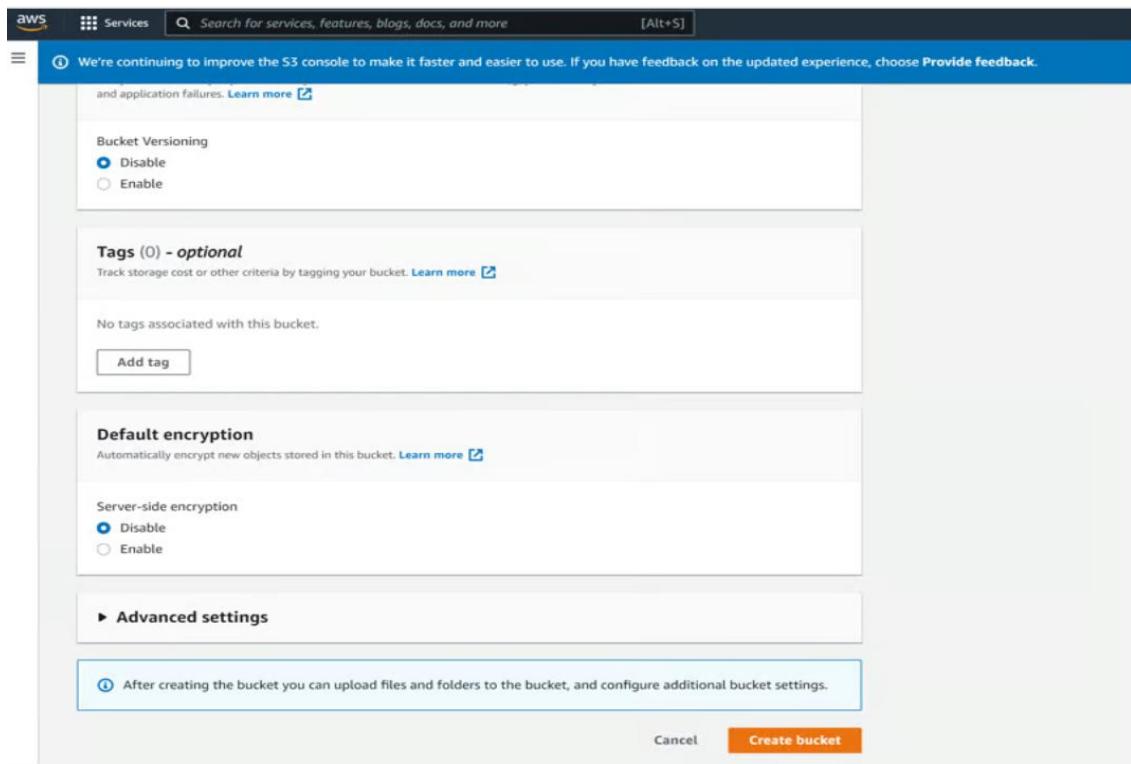


Step 2: Select S3 under the AWS Services navigation search bar On the S3 main screen click 'Create Bucket'

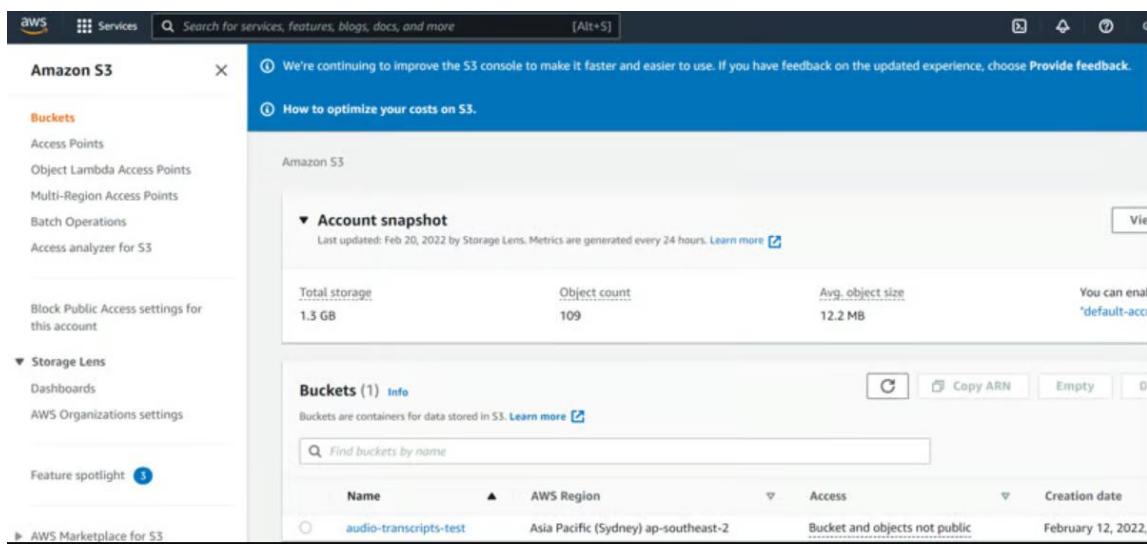
The screenshot shows the AWS S3 console. On the left, a sidebar lists various features like Buckets, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, and Access analyzer for S3. Below this is a section for 'Block Public Access settings for this account'. Under 'Storage Lens', there are links for Dashboards and AWS Organizations settings. A 'Feature spotlight' section is also present. At the bottom of the sidebar, there's a link to 'AWS Marketplace for S3'. The main content area has a blue header bar with a message about improving the console. Below this is a 'Account snapshot' section with a 'View Storage' button. The central part shows a table titled 'Buckets (0) Info' with columns for Name, AWS Region, Access, and Creation date. A message says 'No buckets' and 'You don't have any buckets.' with a 'Create bucket' button.

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

The screenshot shows the 'Create bucket' configuration page. At the top, it says 'Amazon S3 > Create bucket'. Below this is a 'General configuration' section with fields for 'Bucket name' (set to 'audio-transcript') and 'AWS Region' (set to 'Asia Pacific (Sydney) ap-southeast-2'). There's also a 'Copy settings from existing bucket - optional' section with a 'Choose bucket' button. The next section is 'Object Ownership' with a note about controlling object ownership from other accounts. It contains two options: 'ACLs disabled (recommended)' (selected) and 'ACLs enabled'. The 'Object Ownership' field at the bottom is set to 'Bucket owner enforced'.



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

Amazon S3 > audio-transcripts-test

audio-transcripts-test Info

Objects Properties Permissions Metrics Management Access Points

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Actions

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	audio files/	Folder	-	-	-

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

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose [Provide feedback](#).

Drag and drop files and folders you want to upload here, or choose Add files, or Add folders.

Files and folders (0)

All files and folders in this table will be uploaded.

Find by name

<input type="checkbox"/>	Name	Folder	Type	Size
No files or folders You have not chosen any files or folders to upload.				

Destination

Destination
<s3://audio-transcripts-test>

Destination details
Bucket settings that impact new objects stored in the specified destination.

Permissions
Grant public access and access to other AWS accounts.

Properties
Specify storage class, encryption settings, tags, and more.

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.

Name	Folder	Type	Size	Status
4065.mp3	audio files/	audio/mpeg	13.7 MB	Succeeded
4074.mp3	audio files/	audio/mpeg	13.7 MB	Succeeded
4077.mp3	audio files/	audio/mpeg	13.7 MB	Succeeded
4092.mp3	audio files/	audio/mpeg	13.7 MB	Succeeded

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

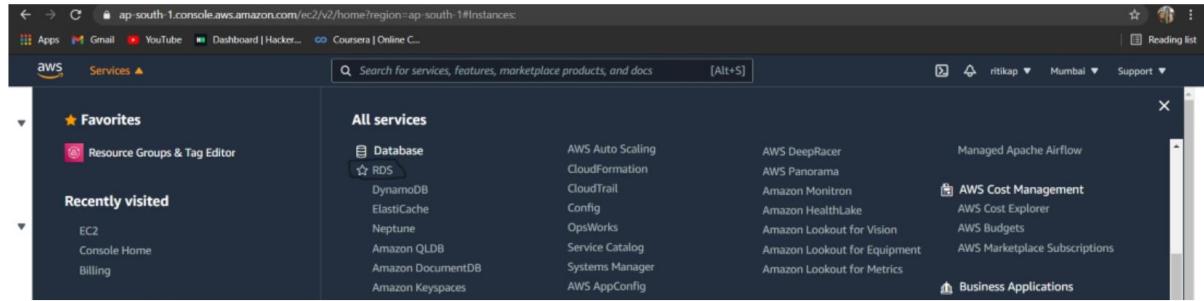
Name	Type	Last modified	Size	Storage class
4065.mp3	mp3	February 12, 2022, 01:11:35 (UTC+11:00)	13.7 MB	Standard
4074.mp3	mp3	February 12, 2022, 01:11:42 (UTC+11:00)	13.7 MB	Standard
4077.mp3	mp3	February 12, 2022, 01:11:49 (UTC+11:00)	13.7 MB	Standard
4092.mp3	mp3	February 12, 2022, 01:11:56 (UTC+11:00)	13.7 MB	Standard
4093.mp3	mp3	February 12, 2022, 01:12:03 (UTC+11:00)	13.7 MB	Standard
4104.mp3	mp3	February 12, 2022, 01:12:11 (UTC+11:00)	13.7 MB	Standard
4112.mp3	mp3	February 12, 2022, 01:12:18 (UTC+11:00)	13.7 MB	Standard
4145.mp3	mp3	February 12, 2022, 01:12:25 (UTC+11:00)	13.7 MB	Standard
4156.mp3	mp3	February 12, 2022, 01:12:32 (UTC+11:00)	8.8 MB	Standard
4157.mp3	mp3	February 12, 2022, 01:12:36 (UTC+11:00)	6.2 MB	Standard
4170.mp3	mp3	February 12, 2022, 01:12:41 (UTC+11:00)	11.1 MB	Standard
4185.mp3	mp3	February 12, 2022, 01:12:47 (UTC+11:00)	11.2 MB	Standard

How to create Amazon RDS.

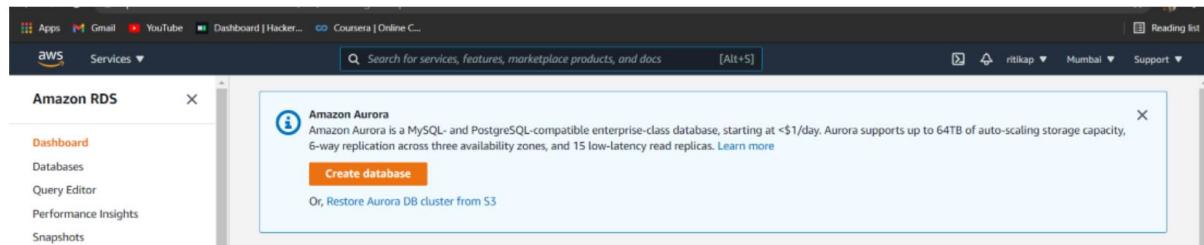
Amazon RDS or Amazon Relational Database Service makes the manipulation of databases much easier and handy for its users. It is very much similar to the traditional

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

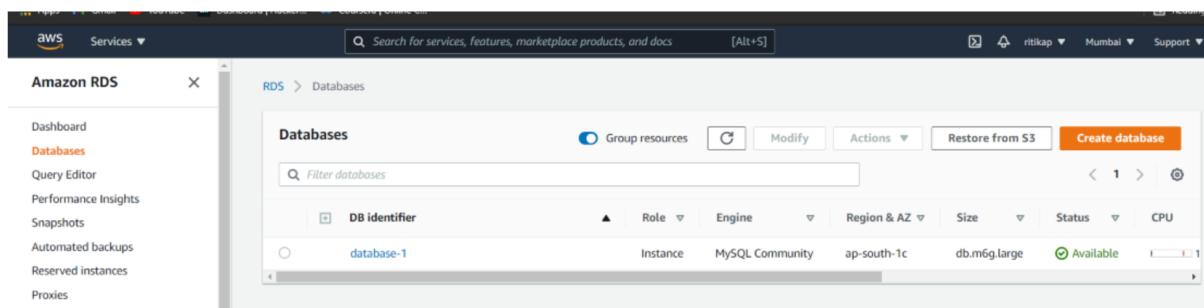
Step 1: First login into your AWS account and once your primary screen is loaded. Click on **services** which is written on the left and from the drop-down menu under **Database** there is an option **RDS**. Click on that and wait for the page to be loaded. Here is the image for better understanding.



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.

The screenshot shows the AWS RDS Databases page. On the left, there's a sidebar with options like Dashboard, Databases (which is selected), Query Editor, Performance Insights, Snapshots, Automated backups, Reserved instances, and Proxies. The main area is titled 'Databases' and shows a table with one row for 'database-1'. The table columns are DB identifier, Instance, and Engine (MySQL Community). To the right of the table are buttons for Actions (Stop, Reboot, Delete, Create read replica, Promote, Take snapshot, Restore to point in time), Restore from S3, and Create database.

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.

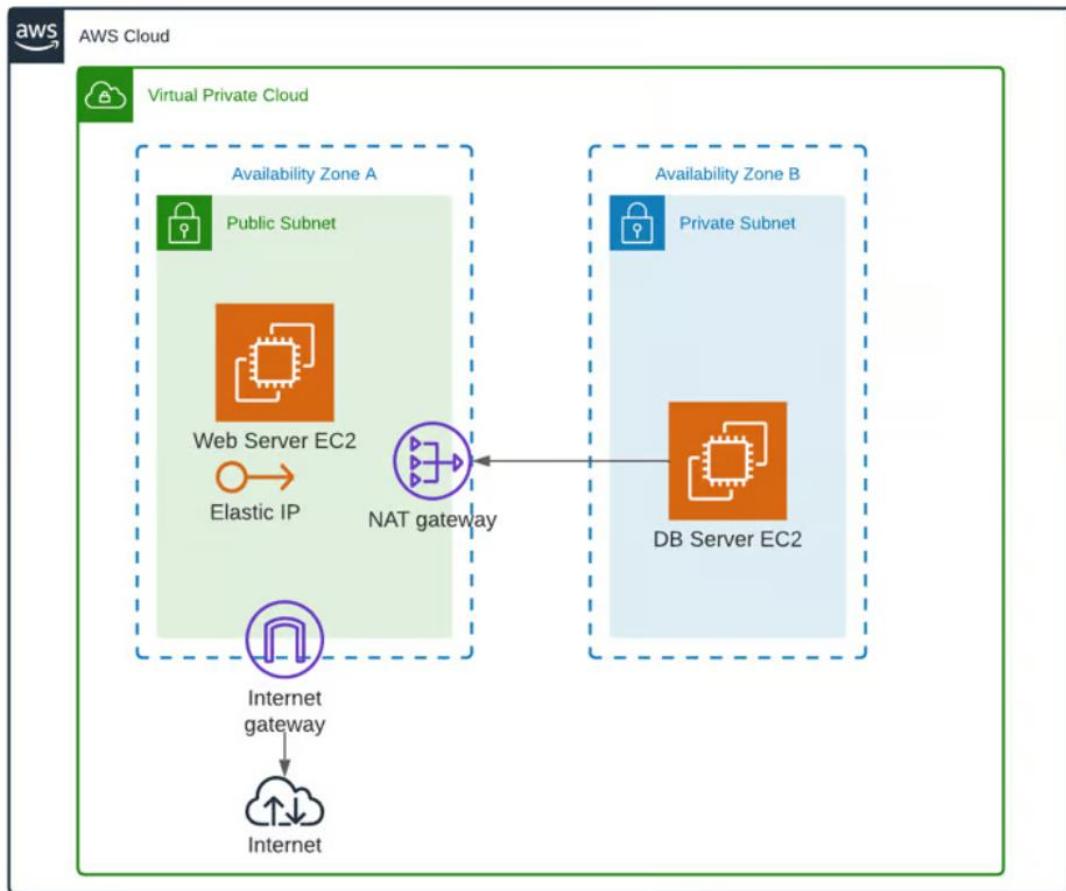
The screenshot shows the 'Take DB snapshot' configuration dialog. It has a 'Settings' section with a note: 'To take a snapshot of this DB instance you must provide a name for the snapshot.' Below this are fields for 'DB instance' (set to 'database-1') and 'Snapshot name' (set to 'database-1'). At the bottom are 'Cancel' and 'Take snapshot' buttons.

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.

The screenshot shows the AWS RDS Snapshots page. The left sidebar includes 'Schemas', 'Tables', 'Functions', 'Procedures', 'Triggers', 'Events', 'Views', 'Materialized views', and 'Constraints'. The main area is titled 'Snapshots' and shows a table for 'Manual snapshots (1)'. The table has columns for Snapshot name (database-1), DB instance or cluster (database-1), Snapshot creation time (April 27, 2021), and DB Instance created (April 27, 2021). There are tabs for Manual, System, Shared with me, Public, Backup service, and Exports in Amazon S3. A 'Actions' button is also present.

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.

The screenshot shows the AWS VPC service dashboard. On the left, there's a sidebar with links like 'VPC dashboard', 'EC2 Global View', and a dropdown for 'Filter by VPC'. The main area is titled 'Your VPCs (1)' and lists a single VPC named 'Default_VPC' with the following details:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP
Default_VPC	vpc-70bd381b	Available	172.31.0.0/16	-	opted-out

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

Create only the VPC resource or the VPC and other networking resources.

VPC only VPC and more

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

IPv4 CIDR block [Info](#)

IPv4 CIDR manual input
 IPAM-allocated IPv4 CIDR block

IPv4 CIDR

IPv6 CIDR block [Info](#)

No IPv6 CIDR block
 IPAM-allocated IPv6 CIDR block
 Amazon-provided IPv6 CIDR block
 IPv6 CIDR owned by me

Tenancy [Info](#)

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - <i>optional</i>
<input type="text" value="Name"/> <input type="button" value="X"/>	<input type="text" value="demo-vpc"/> <input type="button" value="X"/> <input type="button" value="Remove"/>

You can add 49 more tags.

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.

Create subnet Info

VPC

VPC ID

Create subnets in this VPC.

vpc-05e1645bcb19d5c25 (demo-vpc)



Associated VPC CIDRs

IPv4 CIDRs

10.0.0.0/16

Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 2

Subnet name

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

demo-public-subnet

The name can be up to 256 characters long.

Availability Zone Info

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

US East (Ohio) / us-east-2a



IPv4 CIDR block Info

Q 10.0.1.0/24



▼ Tags - optional

Key

Q Name

Value - optional

Q demo-public-subnet



Remove

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

Add new subnet

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.

Subnet 2 of 2

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 CIDR block [Info](#)

Tags - optional

Key	Value - optional	Remove
<input type="text" value="Name"/>	<input type="text" value="demo-private-subnet"/>	<input type="button" value="Remove"/>

You can add 49 more tags.

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.

Create internet gateway [Info](#)

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

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

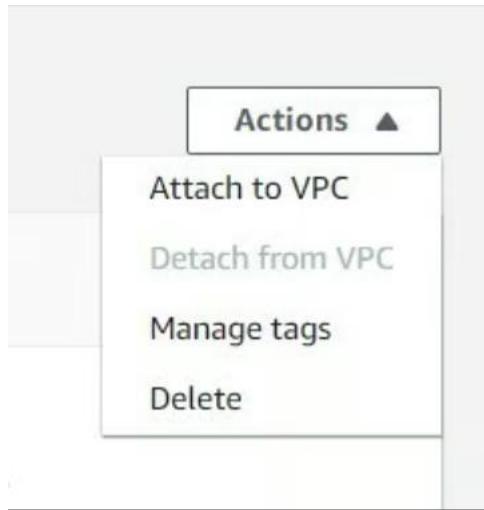
Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional	Remove
<input type="text" value="Name"/>	<input type="text" value="demo-igw"/>	<input type="button" value="Remove"/>

You can add 49 more tags.

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.

Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - optional

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

VPC

The VPC to use for this route table.

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key

Value - optional

You can add 49 more tags.

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.

VPC > Route tables > rtb-076400310867157dc > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	<input type="text" value="Q_ local"/> <input type="button" value="X"/> <input checked="" type="radio"/> Active	No	<input type="button" value="Remove"/>
<input type="text" value="Q_ 0.0.0.0"/> <input type="button" value="X"/>	<input type="text" value="Q_ igw-0ee75b20d7372ee0e"/> <input type="button" value="X"/> -	No	<input type="button" value="Remove"/>

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

The screenshot shows the AWS Route Table Subnet Associations page. It has tabs for Routes, Subnet associations (selected), Edge associations, Route propagation, and Tags. Under Subnet associations, there are two sections: "Explicit subnet associations (0)" and "Subnets without explicit associations (2)".

- Explicit subnet associations (0):** A search bar and a table with columns: Subnet ID, IPv4 CIDR, and IPv6 CIDR. A message says "No subnet associations" and "You do not have any subnet associations." There is a "Edit subnet associations" button.
- Subnets without explicit associations (2):** A search bar and a table with columns: Subnet ID, IPv4 CIDR, and IPv6 CIDR. It lists two subnets: "subnet-067dc538bd6ab92fc / demo-private-subnet" (IPv4: 10.0.2.0/24) and "subnet-0f7ce518f1865a57b / demo-public-subnet" (IPv4: 10.0.1.0/24). There is a "Edit subnet associations" button.

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

The screenshot shows the "Edit subnet associations" dialog box. It has a title "Edit subnet associations" and a subtitle "Change which subnets are associated with this route table." It contains two sections:

- Available subnets (1/2):** A table with columns: Name, Subnet ID, IPv4 CIDR, IPv6 CIDR, and Route table ID. It lists "demo-private-subnet" (subnet-067dc538bd6ab92fc, 10.0.2.0/24, Main rtb-024899305ab335172) and "demo-public-subnet" (subnet-0f7ce518f1865a57b, 10.0.1.0/24, Main rtb-024899305ab335172).
- Selected subnets:** A list containing "subnet-0f7ce518f1865a57b / demo-public-subnet".

At the bottom are "Cancel" and "Save associations" buttons.

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

The screenshot shows the AWS Subnets list page. It has a title "Subnets (1/2) Info" and a search bar. It lists two subnets:

Name	Subnet ID	State	VPC	IPv4 CIDR
demo-private-subnet	subnet-067dc538bd6ab92fc	Available	vpc-05e1645bcb19d5c25	10.0.2.0/24
demo-public-subnet	subnet-0f7ce518f1865a57b	Available	vpc-05e1645bcb19d5c25	10.0.1.0/24

A context menu is open over the "demo-public-subnet" row, listing actions: View details, Create flow log, Edit subnet settings, Edit IPv6 CIDRs, Edit network ACL association, Edit route table association, Edit CIDR reservations, Share subnet, Manage tags, and Delete subnet.

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

The screenshot shows the AWS EC2 instance creation wizard. It displays the following details:

- Ubuntu Server 20.04 LTS (HVM), SSD Volume Type**: ami-0960ab670c8bb45f3 (64-bit x86) / ami-0b4fa084a1e7e6f5a (64-bit Arm)
- Root device type**: ebs
- Virtualization type**: hvm
- ENAv Enabled**: Yes
- Select** button
- 64-bit (x86)** and **64-bit (Arm)** radio buttons

You can see the ec2 instance is up and running.

Instances (2) Info										Launch Instances					
<input type="checkbox"/> Name		Instance ID		Instance state		Instance type		Status check		Alarm status		Availability Zone		Public IPv4 DNS	
<input type="checkbox"/>	demo-private-ec2	i-0ab58058e1880c9bb		Running	View details	t2.micro		2/2 checks passed	No alarms	+	us-east-2b		-		
<input type="checkbox"/>	demo-public-ec2	i-0744c4c9e75e736e2		Running	View details	t2.micro		2/2 checks passed	No alarms	+	us-east-2a		-		

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

Allocate Elastic IP address [Info](#)

Elastic IP address settings [Info](#)

Public IPv4 address pool

Amazon's pool of IPv4 addresses

Public IPv4 address that you bring to your AWS account (option disabled because no pools found) [Learn more](#)

Customer owned pool of IPv4 addresses (option disabled because no customer owned pools found) [Learn more](#)

Global static IP addresses

AWS Global Accelerator can provide global static IP addresses that are announced worldwide using anycast from AWS edge locations. This can help improve the availability and latency for your user traffic by using the Amazon global network. [Learn more](#)

[Create accelerator](#)

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tag

[Cancel](#) [Allocate](#)

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

Elastic IP addresses (1/1)						Actions		Allocate Elastic IP address	
<input type="checkbox"/> Filter Elastic IP addresses									
<input checked="" type="checkbox"/>	Name	Allocated IPv4 add...	Type	Allocation ID					
<input checked="" type="checkbox"/>	-	5.18.247.199	Public IP	eipalloc-05209d118bcc13a91		View details		Release Elastic IP address	1 > View details

Select previously created public ec2 and click the associate button.

Associate Elastic IP address

Choose the instance or network interface to associate to this Elastic IP address (3.18.247.199)

Elastic IP address: 3.18.247.199

Resource type

Choose the type of resource with which to associate the Elastic IP address.

- Instance
- Network interface

⚠️ If you associate an Elastic IP address to an instance that already has an Elastic IP address associated, this previously associated Elastic IP address will be disassociated but still allocated to your account. [Learn more](#)

Instance



Private IP address

The private IP address with which to associate the Elastic IP address.

Reassociation

Specify whether the Elastic IP address can be reassigned to a different resource if it already associated with a resource.

- Allow this Elastic IP address to be reassigned

[Cancel](#)
[Associate](#)

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.

```
ubuntu@ip-10-0-2-224:~$ ping google.com
PING google.com (142.250.191.110) 56(84) bytes of data.
```

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.

Destination	Target	Status	Propagated
0.0.0.0	nat-0d240331837bf91dc	Active	No
10.0.0.0/16	local	Active	No

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

The screenshot shows the AWS VPC console interface. The top navigation bar has tabs for 'Routes', 'Subnet associations' (which is the active tab), 'Edge associations', 'Route propagation', and 'Tags'. Below the tabs, there's a section titled 'Explicit subnet associations (0)' with a search bar labeled 'Find subnet association'. To the right of the search bar is a button 'Edit subnet associations'. Below the search bar are two dropdown menus: 'Subnet ID' and 'IPv4 CIDR'. A message 'No subnet associations' is displayed, followed by the text 'You do not have any subnet associations.'

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

```
ubuntu@ip-10-0-2-224:~$ ping google.com
PING google.com (172.217.2.46) 56(84) bytes of data.
64 bytes from ord37s52-in-f14.1e100.net (172.217.2.46): icmp_seq=1 ttl=103 time=19.0 ms
64 bytes from ord37s52-in-f14.1e100.net (172.217.2.46): icmp_seq=2 ttl=103 time=18.5 ms
64 bytes from ord37s52-in-f14.1e100.net (172.217.2.46): icmp_seq=3 ttl=103 time=18.5 ms
64 bytes from ord37s52-in-f14.1e100.net (172.217.2.46): icmp_seq=4 ttl=103 time=18.5 ms
64 bytes from ord37s52-in-f14.1e100.net (172.217.2.46): icmp_seq=5 ttl=103 time=18.5 ms
```

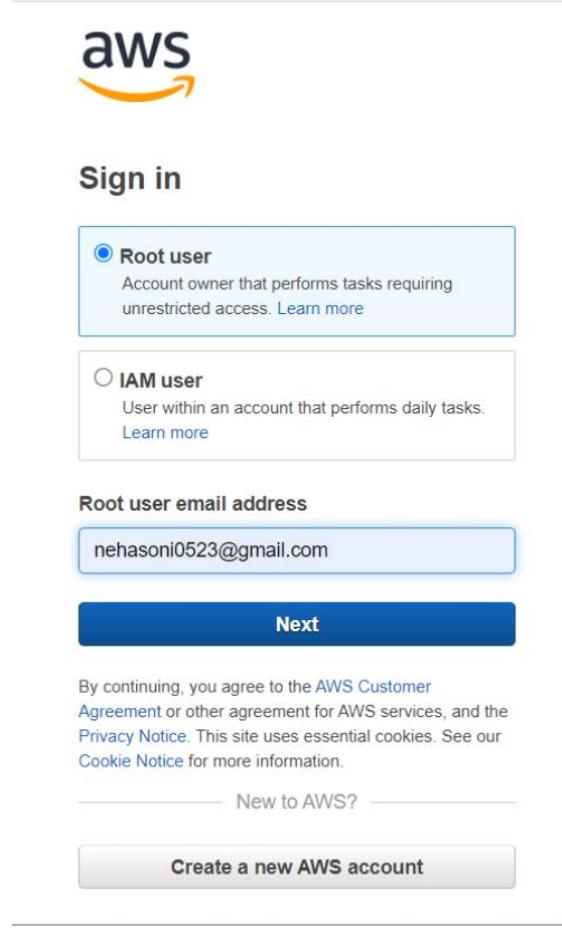
Congratulations, Now you can create AWS VPC.

For clean up

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

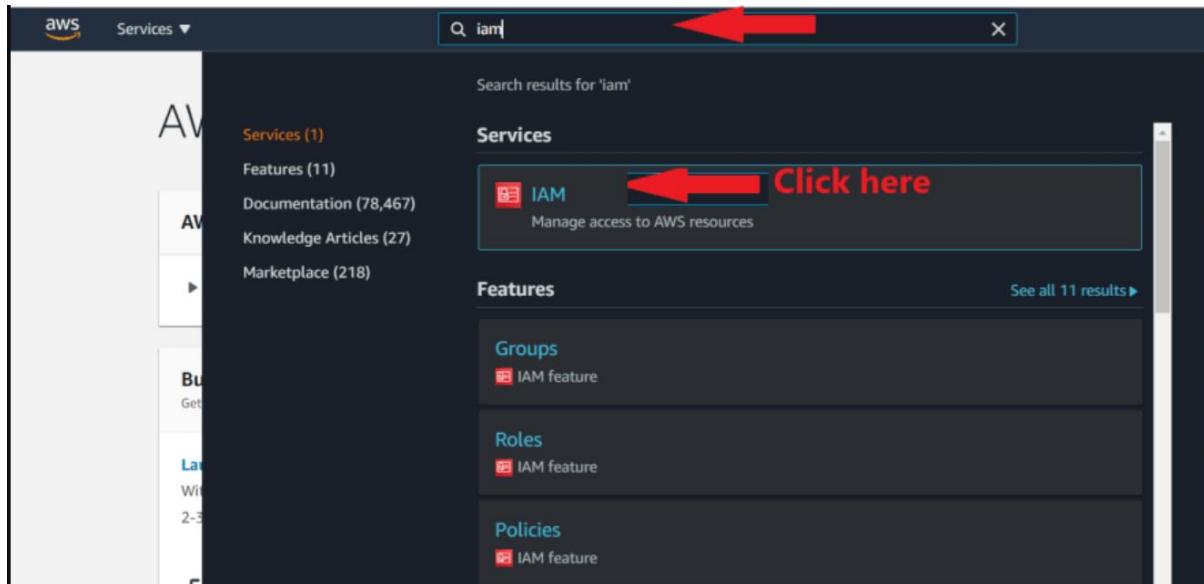
How to create IAM users and roles.

Step 1:- Open your favourite browser and navigate to [AWS Login Page](#)

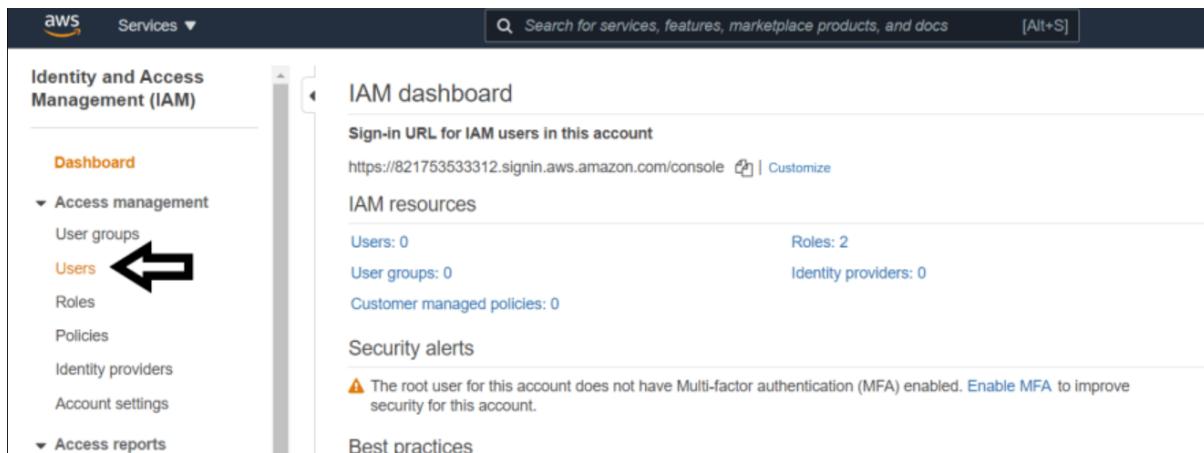


Then enter in your password and click submit. You have now successfully signed in to the AWS Management Console.

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

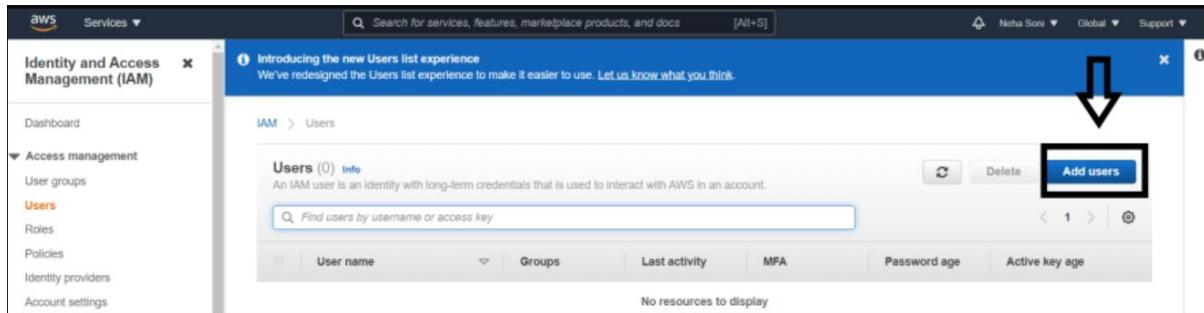


Step 3:- On the left side of the page, you should see an option called **Users**.



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 user.

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.

Add user

1 2 3 4 5

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

[Add another user](#)

Step 1

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type* Programmatic access

Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

AWS Management Console access

Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password* Autogenerated password
 Custom password

Show password

Step 3

Require password reset User must create a new password at next sign-in

Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

Step 4

Step 2

Step 5

[Cancel](#)

Next: Permissions

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.

Add user

1 2 3 4 5

Set permissions

Add user to group Copy permissions from existing user Attach existing policies directly

Create policy

Filter policies v ec2 Choose policies from here Showing 25 results

Policy name	Type	Used as
AmazonEC2ContainerRegistryFullAccess	AWS managed	None
AmazonEC2ContainerRegistryPowerUser	AWS managed	None
AmazonEC2ContainerRegistryReadOnly	AWS managed	None
AmazonEC2ContainerServiceAutoscaleRole	AWS managed	None
AmazonEC2ContainerServiceEventsRole	AWS managed	None
AmazonEC2ContainerServiceforEC2Role	AWS managed	None
AmazonEC2ContainerServiceRole	AWS managed	None
<input checked="" type="checkbox"/> AmazonEC2FullAccess	AWS managed	None
AmazonEC2ReadOnlyAccess	AWS managed	None
AmazonEC2RoleforAWSCodeDeploy	AWS managed	None

Cancel Previous Next: Tags

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.

Add user

1 2 3 4 5

Add tags (optional)

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. [Learn more](#)

Key	Value (optional)	Remove
DevOps Engineer		x
Add new key		

You can add 49 more tags.

Cancel Previous Next: Review

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.

Add user

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	nehasoni
AWS access type	AWS Management Console access - with a password
Console password type	Custom
Require password reset	Yes
Permissions boundary	Permissions boundary is not set

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	AdministratorAccess
Managed policy	AmazonEC2FullAccess
Managed policy	IAMUserChangePassword

Tags

The new user will receive the following tag

Cancel Previous **Create user**



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

Add user

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://821753533312.signin.aws.amazon.com/console>

Download .csv 

User	Email login instructions
nehasoni	Send email

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.

Add user

1 2 3 4 5

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at <https://821753533312.signin.aws.amazon.com/console>

[Download .csv](#)

User	Email login instructions
nehasoni	Send email



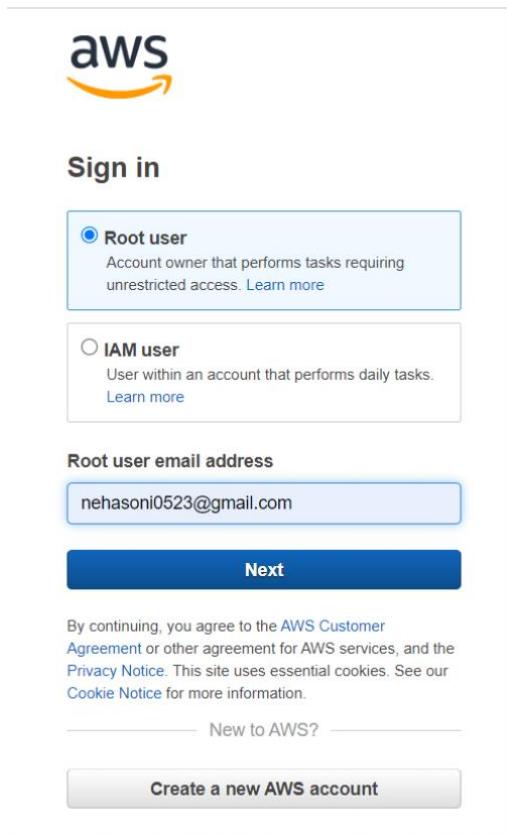
Step 2:- Enter the username and password and click sign in.

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.

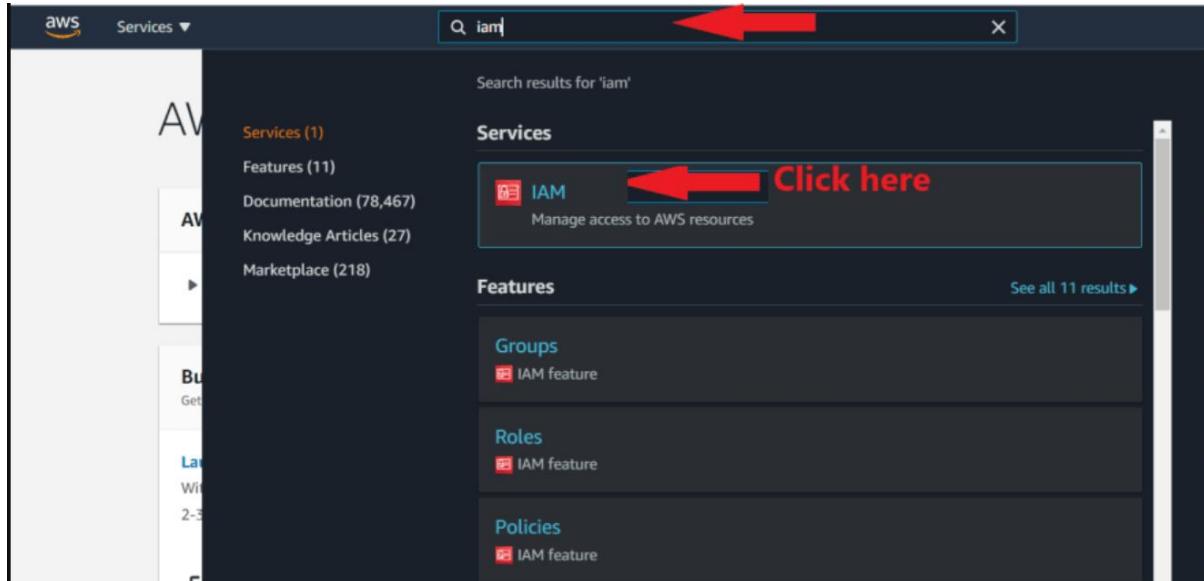
Step 1:- Open your favourite browser and navigate to [AWS Login Page](#)



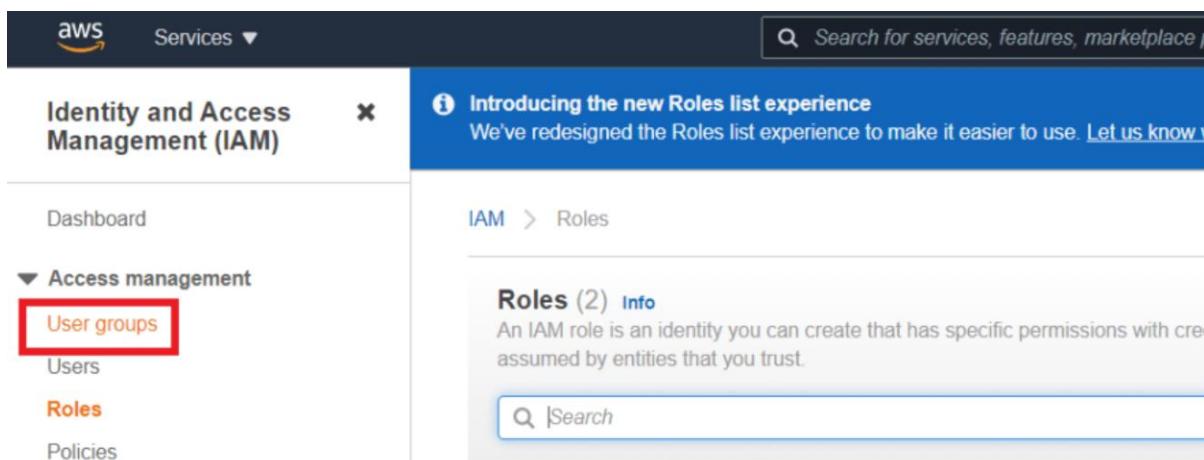
The screenshot shows the AWS Sign In page. At the top is the AWS logo. Below it is the "Sign in" header. There are two radio button options: "Root user" (selected) and "IAM user". The "Root user" option is described as "Account owner that performs tasks requiring unrestricted access." Below the radio buttons is a field labeled "Root user email address" containing "nehasoni0523@gmail.com". A large blue "Next" button is at the bottom of this section. At the very bottom of the page, there is a note about agreeing to the AWS Customer Agreement and Privacy Notice, followed by links for "New to AWS?" and "Create a new AWS account".

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.

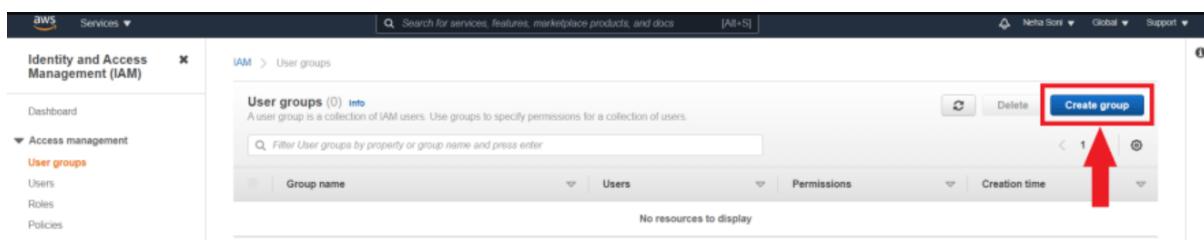


Step 3:- On the left side of the page, you should see an option called **Users groups**.



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

The screenshot shows the 'Create user group' page in the AWS IAM console. At the top, there's a breadcrumb navigation: IAM > User groups > Create user group. Below it, a section titled 'Name the group' has a text input field containing 'Developers'. A large black arrow points to this input field. Below this, another section titled 'Add users to the group - Optional (Selected 1/1)' shows a table with one user selected: 'nehasoni'. A large black arrow points down to this table. At the bottom of the page is a table of AWS managed policies, with 'AmazonEC2FullAccess' checked.

Policy Name	Type	Description
AWSDirectConnectReadOnlyAccess	AWS managed	Provides read only access to Direct Connect interface
AmazonGlacierReadOnlyAccess	AWS managed	Provides read only access to Glacier
AWSMarketplaceFullAccess	AWS managed	Provides the ability to subscribe to Marketplace products
AWSSSDirectoryAdministrator	AWS managed	Administrator access for SSSD
AWSIoT1ClickReadOnlyAccess	AWS managed	Provides read only access to IoT 1-Click
AutoScalingConsoleReadOnlyAccess	AWS managed	Provides read-only access to Auto Scaling console
AmazonDMSRedshiftS3Role	AWS managed	Provides access to manage Amazon DMS Redshift S3 role
AWSQuickSightListIAM	AWS managed	Allow QuickSight to list IAM users
AWSHHealthFullAccess	AWS managed	Allows full access to the AW HHealth service
AlexaForBusinessGatewayExecution	AWS managed	Provide gateway execution
AmazonElasticTranscoder_ReadOnlyAccess	AWS managed	Grants users read-only access to Elastic Transcoder
AmazonRDSFullAccess	AWS managed	Provides full access to Amazon RDS
SupportUser	AWS managed - job function	This policy grants permission to support users
<input checked="" type="checkbox"/> AmazonEC2FullAccess	AWS managed	Provides full access to Amazon EC2
SecretsManagerReadWrite	AWS managed	Provides read/write access to Secrets Manager
<input type="checkbox"/> AWSIoTThingsRegistration	AWS managed	This policy allows users to register their devices with AWS IoT

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

Congratulations you have created an IAM User group

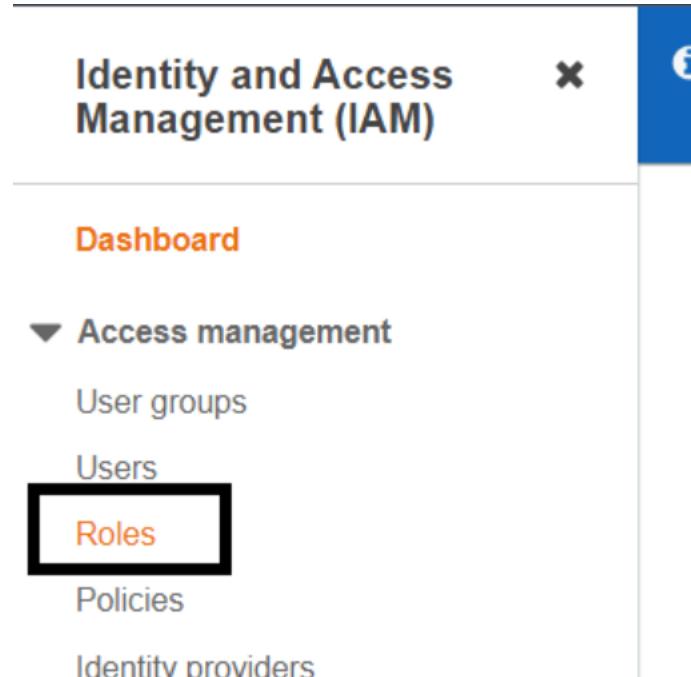
The screenshot shows the 'User groups' page in the AWS IAM console. It displays a table with one group listed: 'Developers'. A large black arrow points up to the 'Create group' button at the top right of the page.

Group name	Users	Permissions	Creation time
Developers	1	Defined	15 days ago

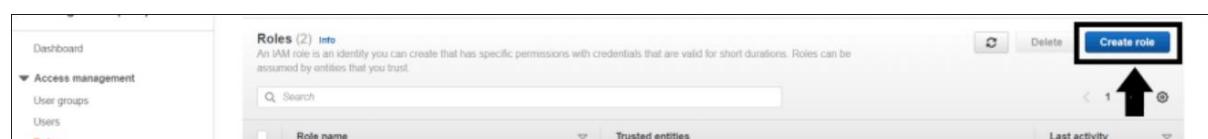
How to create IAM roles for a service:

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.

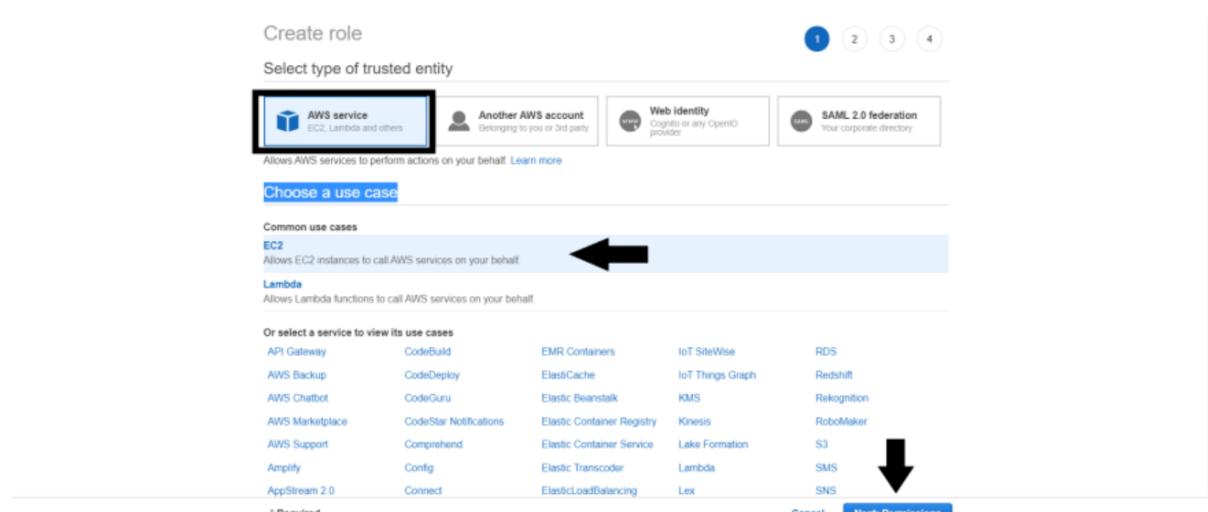
Step 2:- On the left side of the page, you should see an option called **Users groups**.



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.

Create role

1 2 3 4

▼ Attach permissions policies

Choose one or more policies to attach to your new role.

[Create policy](#)

[Filter policies](#) Search Showing 846 results

Policy name	Choose policies from here	Used as
<input type="checkbox"/> AccessAnalyzerServiceRolePolicy		None
<input checked="" type="checkbox"/> AdministratorAccess		Permissions policy (1)
<input type="checkbox"/> AdministratorAccess-Amplify		None
<input type="checkbox"/> AdministratorAccess-AWSElasticBeanstalk		None
<input type="checkbox"/> AlexaForBusinessDeviceSetup		None
<input type="checkbox"/> AlexaForBusinessFullAccess		None
<input type="checkbox"/> AlexaForBusinessGatewayExecution		None
<input type="checkbox"/> AlexaForBusinessLifesizeDelegatedAccessPolicy		None

▶ Set permissions boundary

* Required Cancel Previous **Next: Tags**

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

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

Create role

1 2 3 4

Add tags (optional)

IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. [Learn more](#)

Key	Value (optional)	Remove
AWS EC2 ACCESS	EC2 roles	x
Add new key		

You can add 49 more tags.

Red Up Arrow: Points to the 'Add new key' input field.

Red Down Arrow: Points to the **Next: Review** button.

Cancel Previous **Next: Review**

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

Create role

Review

Provide the required information below and review this role before you create it.

Role name*	CompleteAccess	←
Use alphanumeric and '+-=,@-_.' characters. Maximum 64 characters.		
Role description	Allows EC2 instances to call AWS services on your behalf.	
Maximum 1000 characters. Use alphanumeric and '+-=,@-_.' characters.		
Trusted entities AWS service: ec2.amazonaws.com		
Policies	AdministratorAccess	
Permissions boundary Permissions boundary is not set		
The new role will receive the following tag		
Key	Value	
AWS EC2 ACCESS	EC2 roles	
* Required		Cancel Previous Create role

Congratulations you have created an IAM role.

Now how to attach it to any AWS service:

Let's attach it to AWS EC2 instance: -

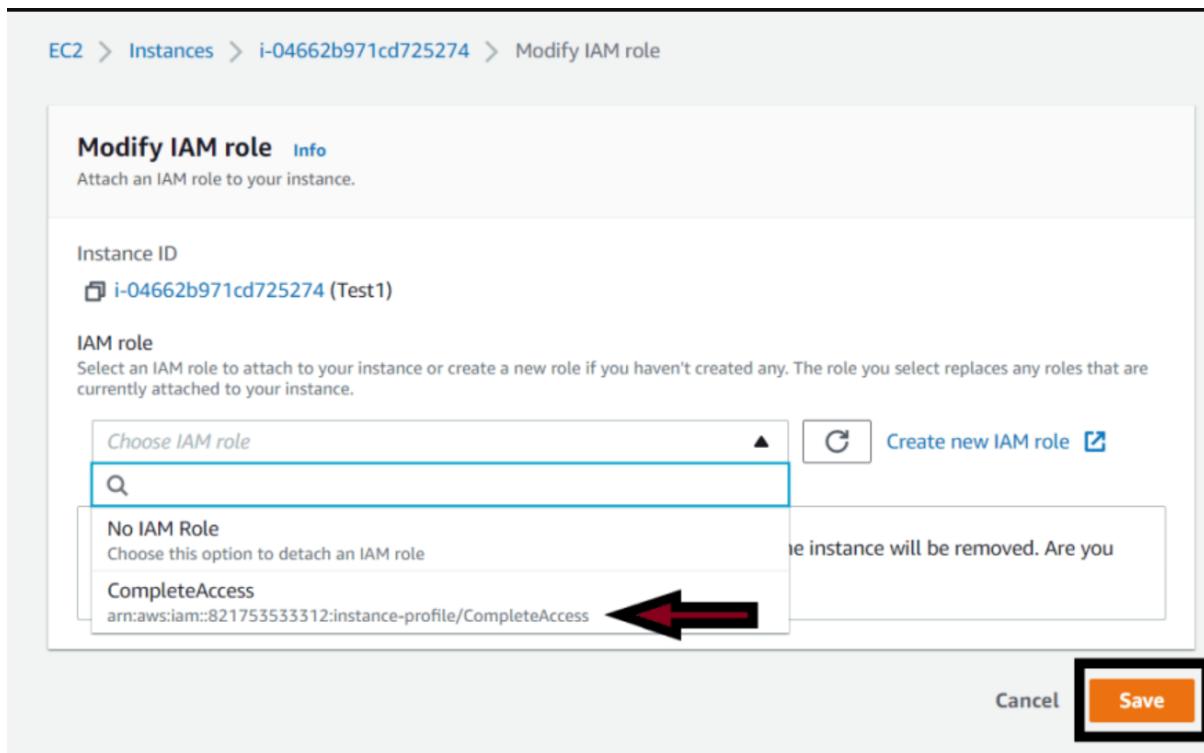
Step 1:- First create an AWS EC2 instance.

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?](#)

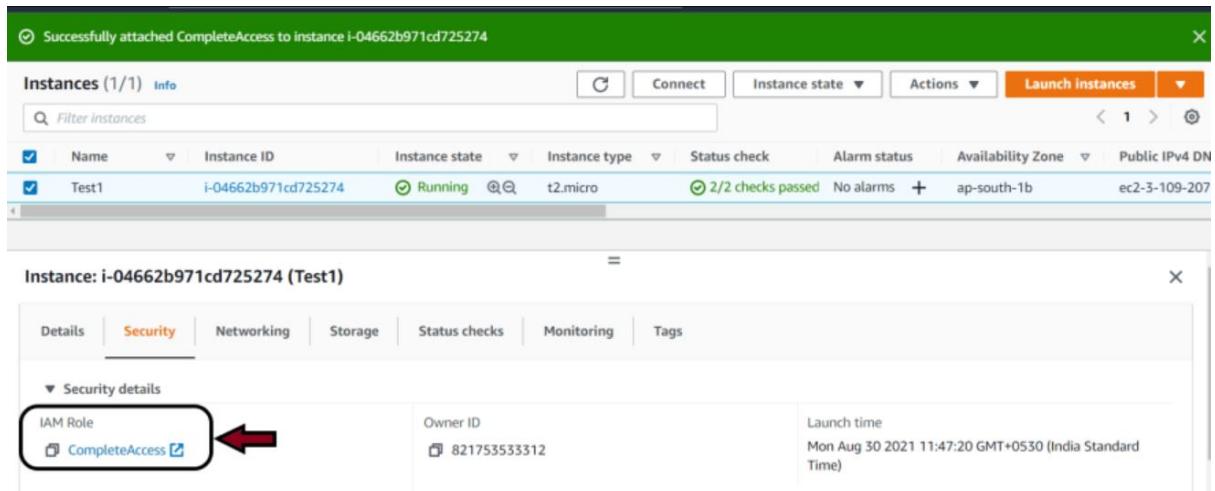
Select the “EC2 Instance” as we did by selecting our “Testing Instance”,

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 your 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.

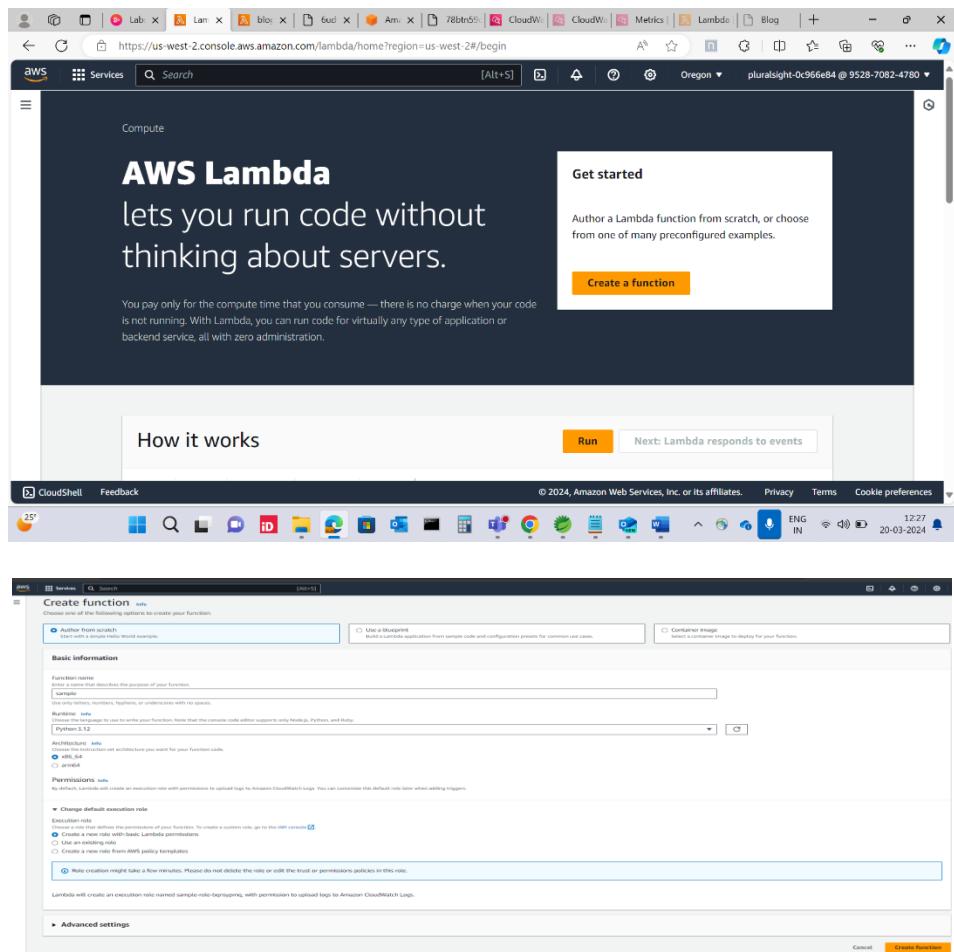
AWS MONITORING:

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.

The screenshot shows the AWS Lambda console. A success message at the top states: "The trigger Home-API was successfully added to function Home. The function is now receiving events from the trigger." Below this, the "Function overview" section is displayed. It includes a "Diagram" tab showing a connection between an "API Gateway" and a "Home" Lambda function, and a "Template" tab. To the right, there's a "Description" panel with fields for "Last modified" (2 minutes ago), "Function ARN" (arn:aws:lambda:us-east-1:723546100796:function:Home), and "Function URL". On the far right, a sidebar titled "Create a simple web app" provides a tutorial on building a simple web application using Lambda.

This screenshot shows the "Configure test event" dialog. It includes fields for "Event name" (set to "sampletest"), "Event sharing settings" (set to "Private"), and a "Template - optional" dropdown (set to "hello-world"). Below these, the "Event JSON" field contains the following JSON code:

```

1: [{}]
2:   "key1": "value1",
3:   "key2": "value2",
4:   "key3": "value3"
5: []

```

At the bottom of the dialog are "Cancel", "Invoke", and "Save" buttons.

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.

The screenshot shows the AWS Lambda console interface. At the top, there are several tabs and a search bar. Below the header, the 'Lambda > Functions' section is displayed. A table lists two functions:

Function name	Description	Package type	Runtime	Last modified
Home	-	Zip	Python 3.12	24 seconds ago
blog	-	Zip	Python 3.12	36 minutes ago

To the right of the table, there is a sidebar titled 'Create a simple web app' with a 'Start tutorial' button. The status bar at the bottom shows 'CloudShell Feedback' and various system icons.

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.

The screenshot shows a web browser window with the title 'My Home page'. The content of the page is:

Hi! My name is sunitha. I like Bengaluru.
ISKCON Temple, Bannerghatta Biological Park, Bull Temple!

The browser's address bar shows the URL: 7k4c33ec1e.execute-api.eu-north-1.amazonaws.com/default/home12. The status bar at the bottom shows weather information (34°C Sunny), system icons, and the date/time (19-03-2024 16:29).

