

The background features a gradient from dark purple at the top to dark blue at the bottom. Overlaid on this are several faint, light-colored circular patterns and a large, semi-circular scale on the left side. The scale has markings from 140 to 260 in increments of 10. The text 'AWS SERVICES' is centered in a white, stylized, serif font.

# AWS SERVICES

**NAME: L.A.VAISHNAVI**  
**ROLL NO.: 20A31A05E5**

# BUILDING A VPC AND LAUNCHING A WEB SERVER

Step 1: First start the lab, when the lab status gets ready, it redirects to the AWS management console.

Step 2: Go to AWS services, search, and choose VPC to open the VPC console.

Step 3: Verify that your regions remain in the N-Virginia(us-east-1). Choose to create VPC in the VPC dashboard

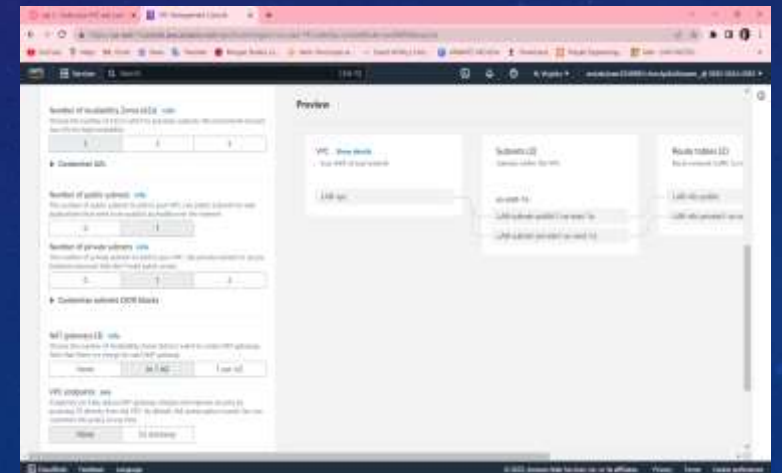
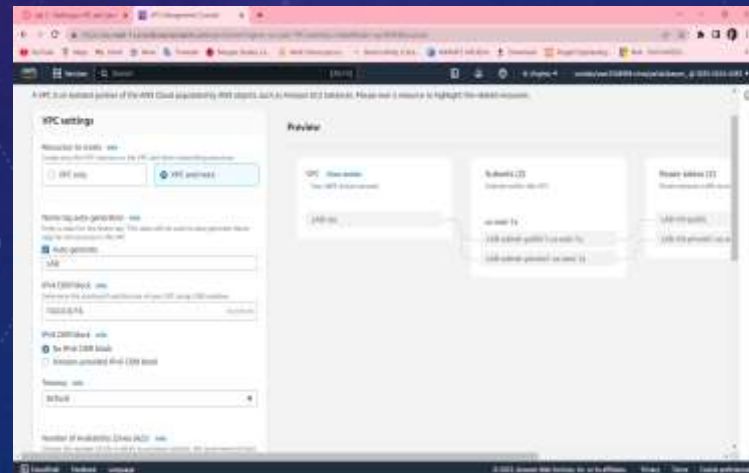
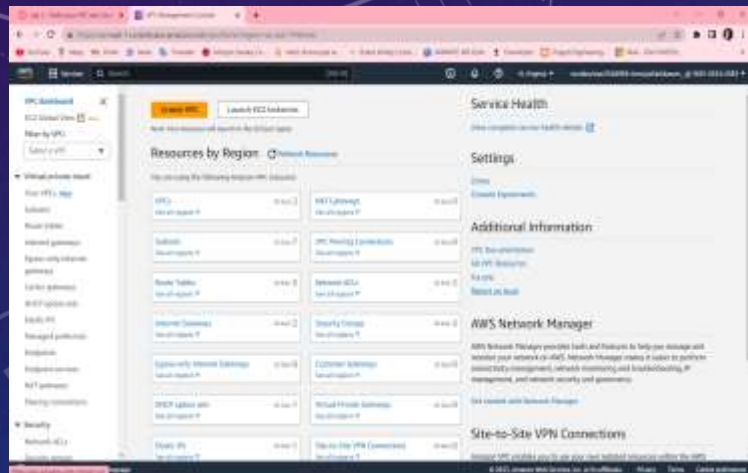
Step 4: Choose VPC and more, in name tag auto-generation, keep auto-generate select and change it to a lab.

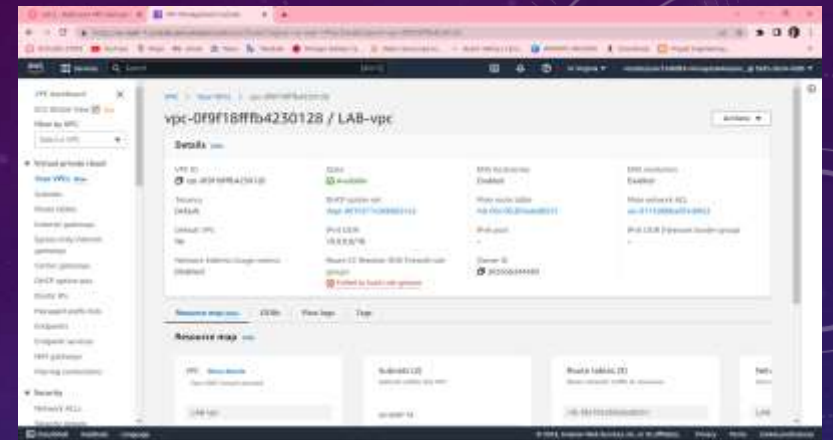
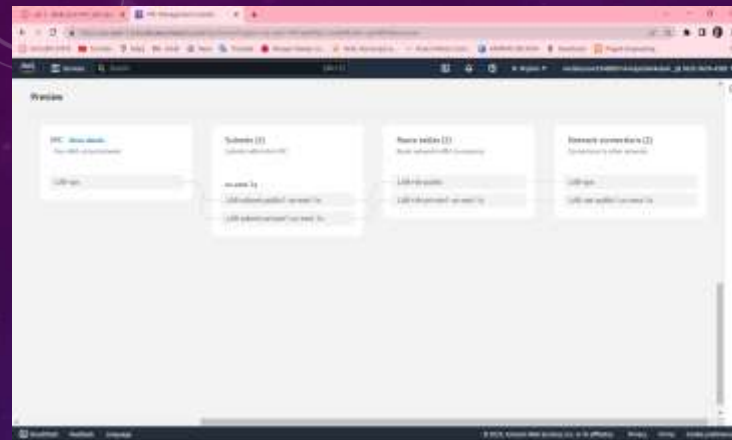
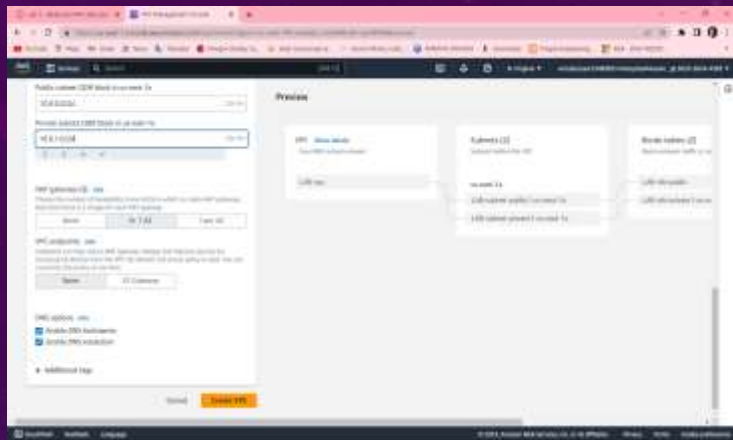
Step 5: Keep the IPv4 CIDR block to 10.10.0.0/16 and next for a number of availability zones select 1, number of public subnets select 1 , number of private subnets select 1

Step 6: Now customize subnets CIDR blocks, change public subnet(us-east-1a) to 10.10.0.0/24, and change private subnet (us-east-1a) to 10.0.1.0/24

Step 7: Set the NAT gateways to 1AZ and set VPC endpoints to none and keep both DNS hostnames and DNS resolutions enabled

Step 8: Check on the preview panel on the right side whether they match with the given regions or not





## CREATING ADDITIONAL SUBNETS :

Step 1: Choose subnets in the left panel, choose to CREATE SUBNET

Step 2: in this, choose VPC ID: LAB-vpc and choose subnet name to lab-subnet-public2, choose availability zone to us-east-1b and choose IPv4 block to 10.0.2.0/24

Step 3: choose to create a subnet and again create the same subnet with lab-subnet-private2, change the IPv4 block to 10.0.2.0/24, and then create a subnet

Step 4: Choose the routing table in the left panel, select lab-rtb-private1-us-east-1a, in the lower panel, choose routes note destination, choose the subnet association tab, in the explicit subnet associations panel choose EDIT SUBNET ASSOCIATIONS

Step 5 : Leave lab-subnet-private1-us-east-1a and also select lab-subnet-private2

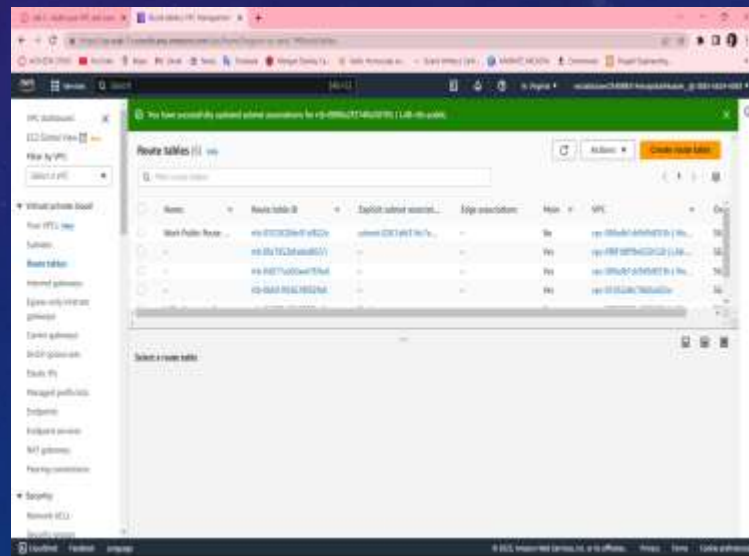
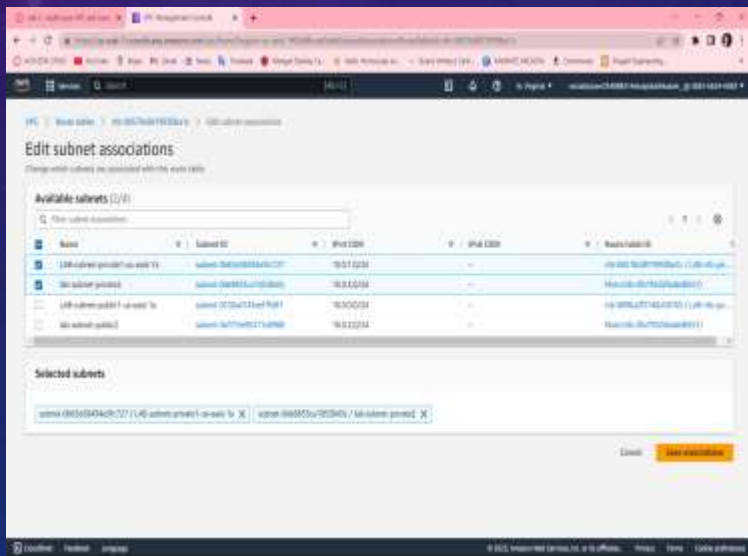
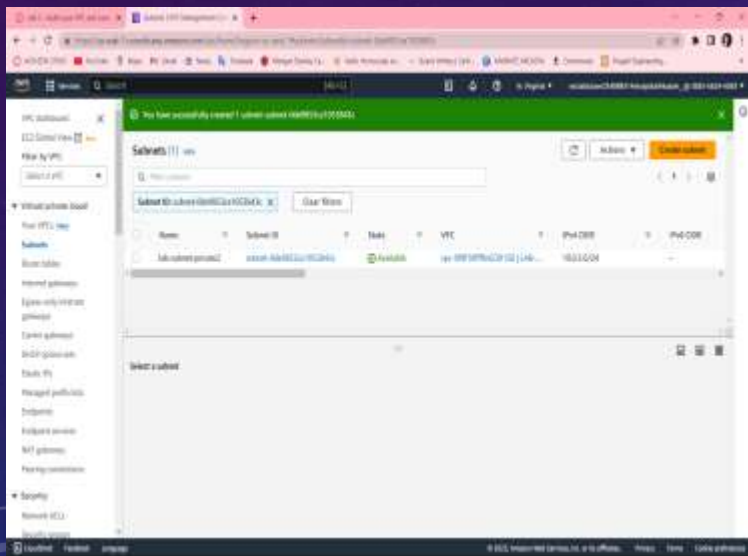
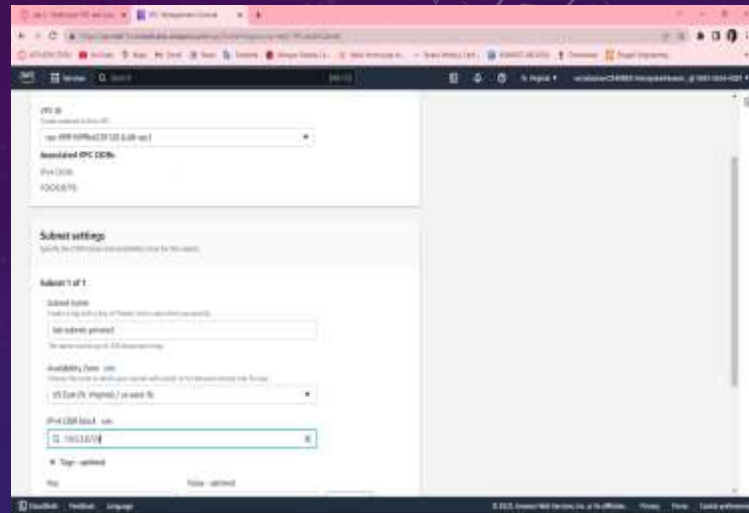
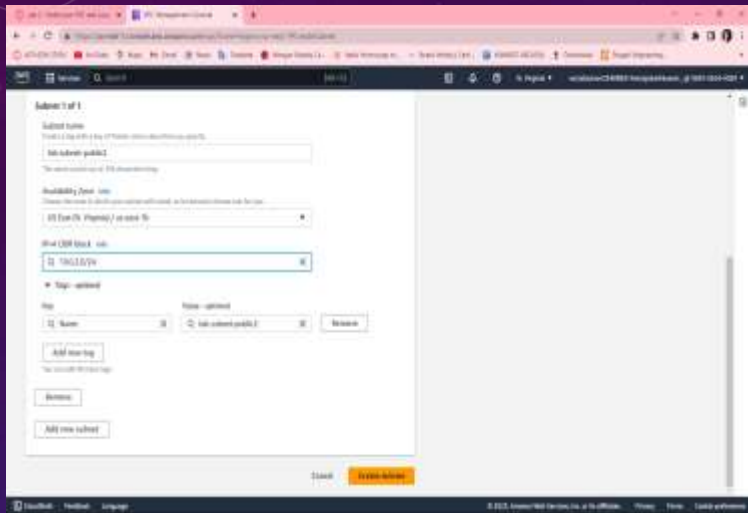
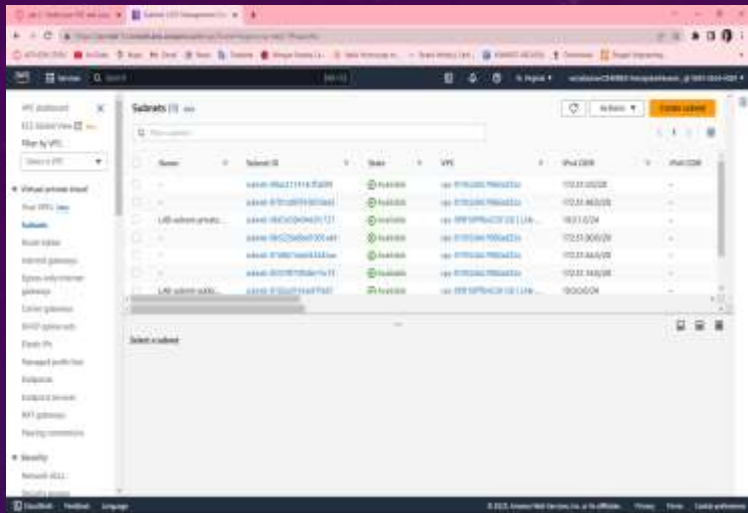
Step 6: Choose SAVE ASSOCIATIONS

Step 7: Select lab-rtb-public route table and deselect any other subnet

Step 8: In the lower panel, again repeat from step 4 but here we select lab-subnet-public2 also

Step 9: Choose SAVE ASSOCIATIONS

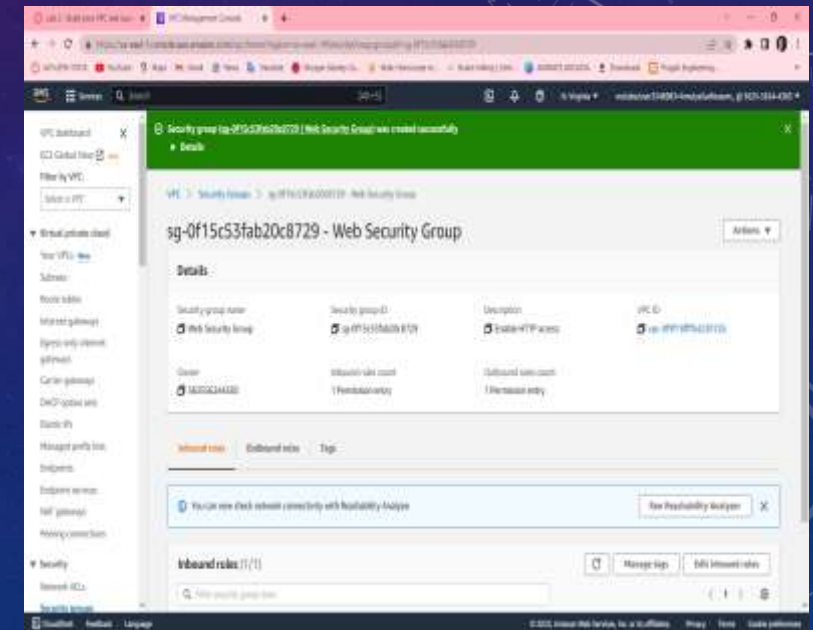
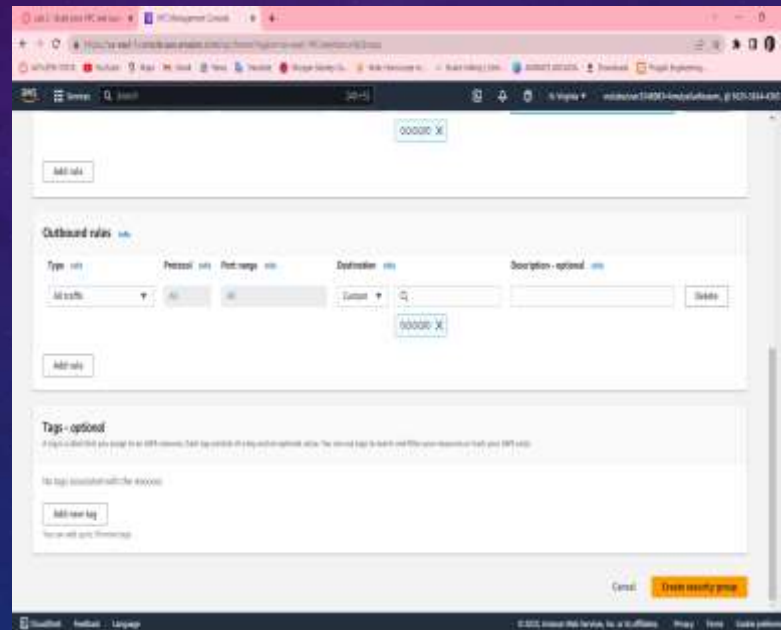
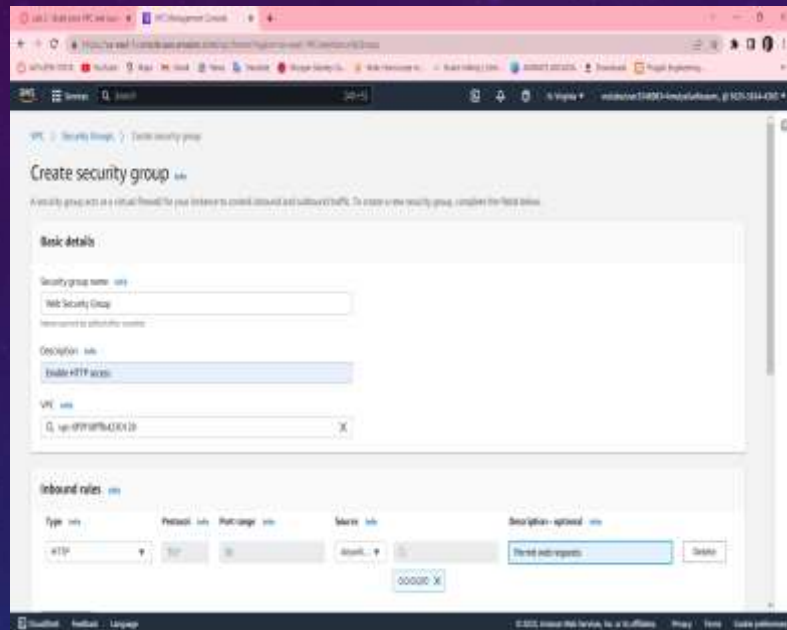




# CREATING A VPC SECURITY GROUP

Step 1: Choose to create a security group and in this choose the security group name to a web security group, have a description as enable HTTP access and choose vpc to lab-vpc

Step 2: In inbound rules choose to add rule and then in this chosen type to HTTP, source to Anywhere ipv4 and description to permit web requests



## LAUNCH A WEB SERVER INSTANCE

Step 1 : Choose EC2 to open EC2 console , from instances click launch instance and give name as web server 1

Step 2 : Keep Amazon Linux select and select amazon linux 2023 AMI , Choose t2.micro

Step 3 : Choose key pair name to vockey , in network settings choose edit select network to lab-vpc ,subnet to lab-subnet-public2 and auton assign to enable

Step 4 : Under firewall choose select existing security group , for common security groups select web security group

Step 5 : Expand the advanced details panel , scroll down upto the user data box appear

```
#!/bin/bash
# Install Apache Web Server and PHP
sudo dnf install -y httpd wget php mariadb105-server
# Download Lab files get https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-100-ACCLFO-2-9026/2-lab2-vpc/s3/lab-app.zip
unzip lab-app.zip -d /var/www/html/
# Turn on web server
chkconfig httpd on
service httpd start
```

Step 6: At the bottom SUMMARY panel choose launch instance ,click on view instances

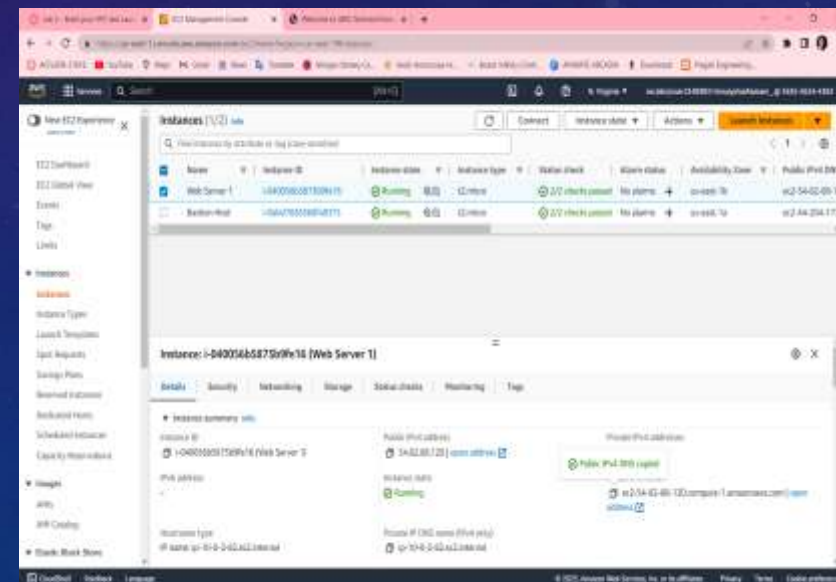
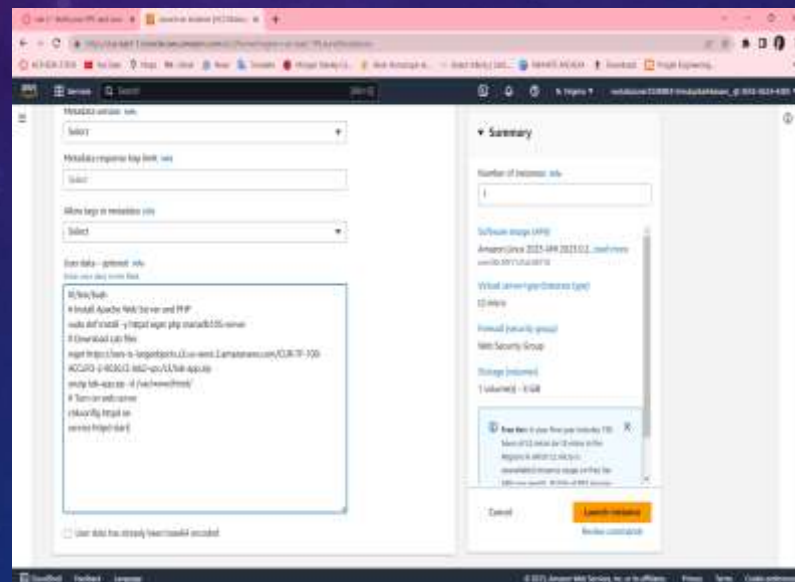
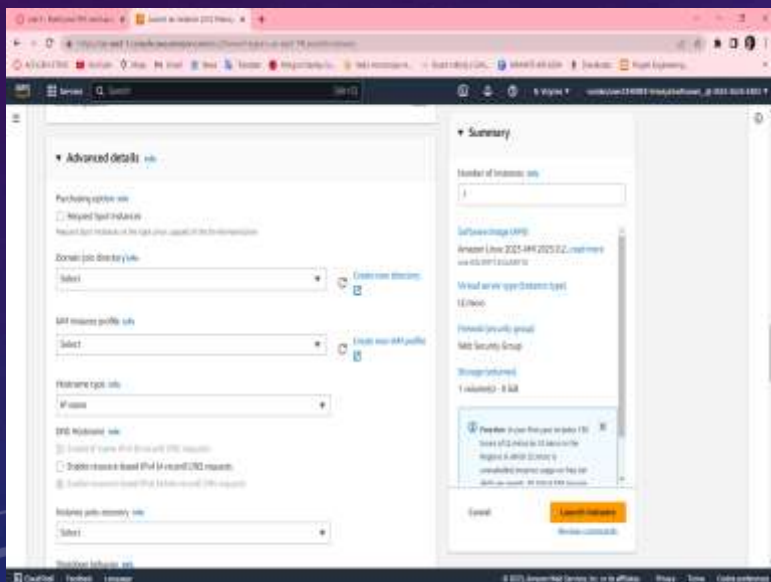
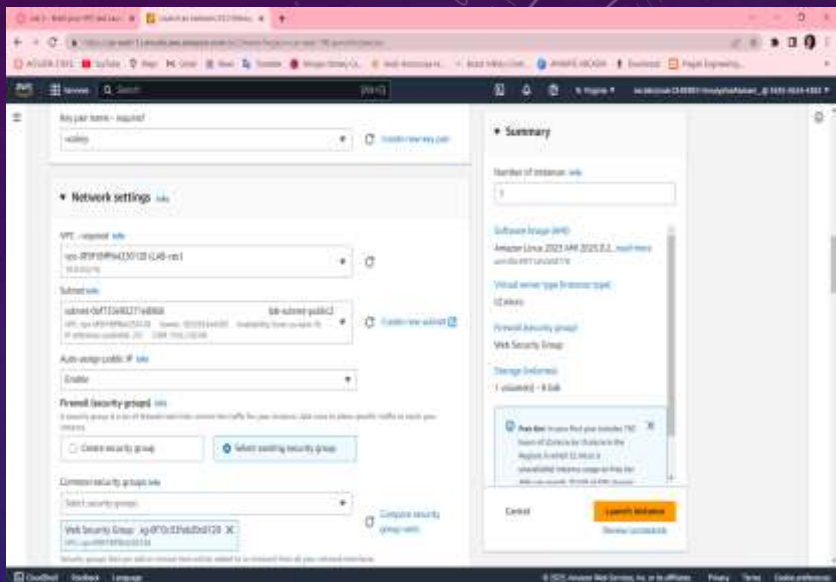
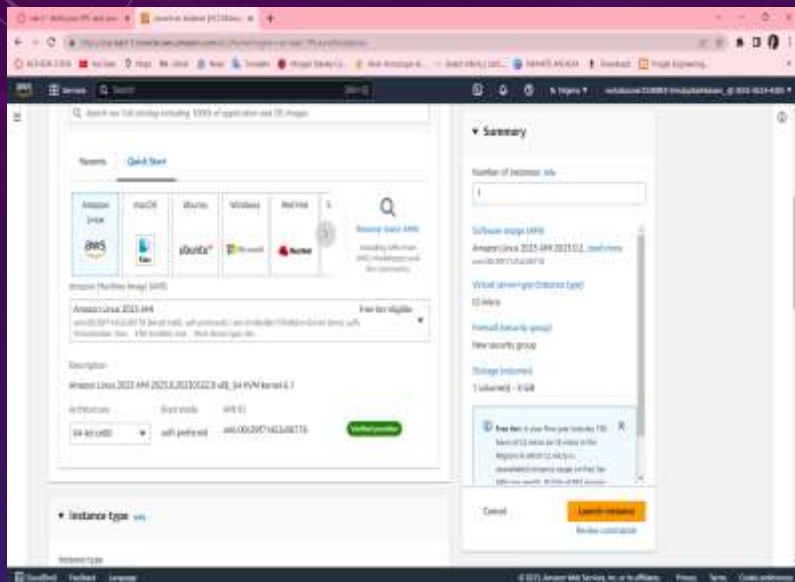
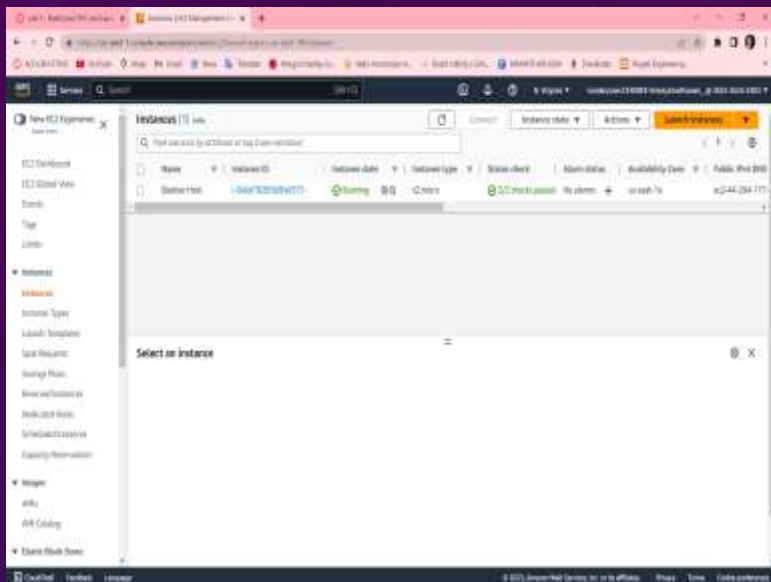
Step 7 : Wait until web server 1 shows 2/2 checks passed

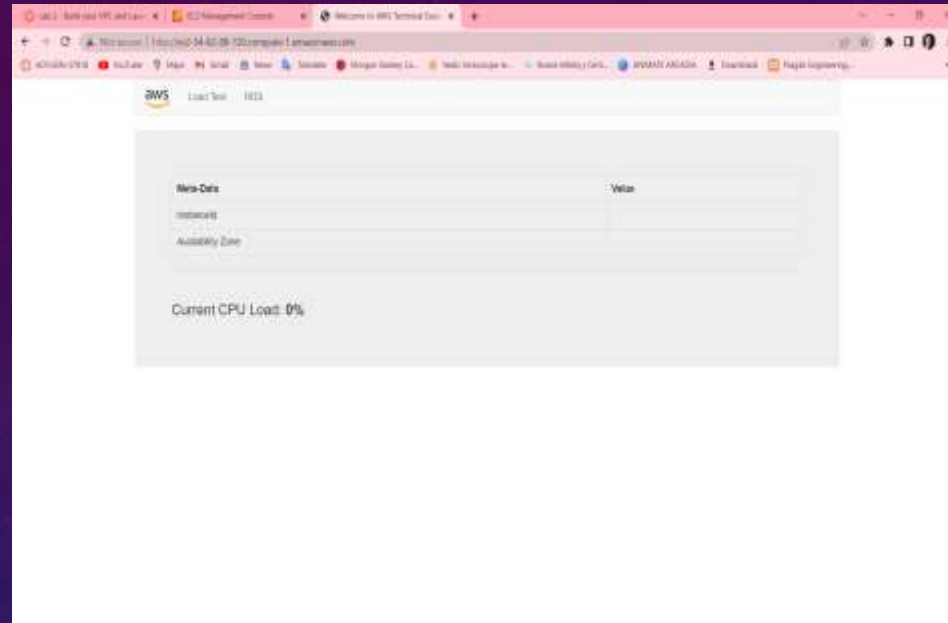
Step 8 : Copy the public ipv4 dns value present in details tab

Step 9 : Open a new browser , copy the public dns

Step 10 : Finally we see a web page displaying AWS logo and instances meta-data values







Finally, a web page opens displaying the AWS logo and instances of metadata values



# AWS EC2 INSTANCE

## CREATING AN EC2 INSTANCE:

Step-1: Go to AWS services , click EC2 and then select 'launch instances'.

Step-2: Name the instance, select an AMI(LINUX,WINDOWS server) , select a key pair and click launch instance.

Step-3: For linux-select ppk key and for windows server-select pem key.

Step-4: If a key pair is not available create a new key.

Step-5: For linux-click connect to instance you will be redirected to the CLI (or) open the putty file configure it to not timeout, and configure putty session. This will redirects you to the CLI.

For windows server-click connect→RDP client→ get password→ upload private key→ decrypt password. Open rdp file and enter the password. This will redirects you to the windows server.

Step-6: Terminate the instances .

Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags**

Name:  [Add additional tags](#)

**Application and OS Images (Amazon Machine Image)**

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instances. Search or browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images.

**Quick Start**

Amazon Linux macOS Ubuntu Windows Red Hat S

**Summary**

Number of instances:

Software image (AMI): Amazon Linux 2023 AMI 2023.02.2... [read more](#)

Virtual server type (instance type): t2.micro

Placement (security group): New security group

Storage (volume): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage at free tier.

[Launch instance](#)

Amazon Linux 2023

<https://aws.amazon.com/linux/amazon-linux-2023>

Last login: Mon Apr 3 13:18:51 2023 from 10.206.107.29  
[ec2-user@ip-172-31-54-87 ~]\$ whoami  
ec2-user  
[ec2-user@ip-172-31-54-87 ~]\$

i-0ca0332f686706307 (myserver)  
PublicIPs: 107.23.105.224 - PrivateIPs: 172.31.56.87

ALLv1-43288 : Modules : Learner Lab : Learner Lab

Home Modules Discussions

Putty Configuration

Categories: Appearance, Behaviour, Translation, Selection, Others

Connection: Data, Proxy, SSH, Fax, Host key, Cipher, Auth, Credentials, SSH, TTY, STX, Tunnel, Sftp, More logs, Serial, Telnet, Proxy, SFTP, SFTP

Choose Connection

Set Seconds between keepalives to 30

This allows you to keep the PuTTY session open for a longer period of time.

4. Configure your PuTTY session:

Choose Session

Host Name (or IP address): Copy and paste the IPv4 Public IP address for the instance. To find it, return to the EC2 Console and choose **Instances**. Check the box next to the instance and in the **Descriptive** tab copy the **IPv4 Public IP** value.

Back in PuTTY, in the **Connection** list, expand **SSH**

Choose **Auth** (don't expand it)

Choose **Browse**

Browse to and select the .ppk file that you downloaded

Choose **Open** to select it

Choose **Open**

5. Choose **Yes**, to trust the host and connect to it.

6. When prompted **login as:** enter: **ec2-user**

ec2-user@ip-172-31-54-87~\$

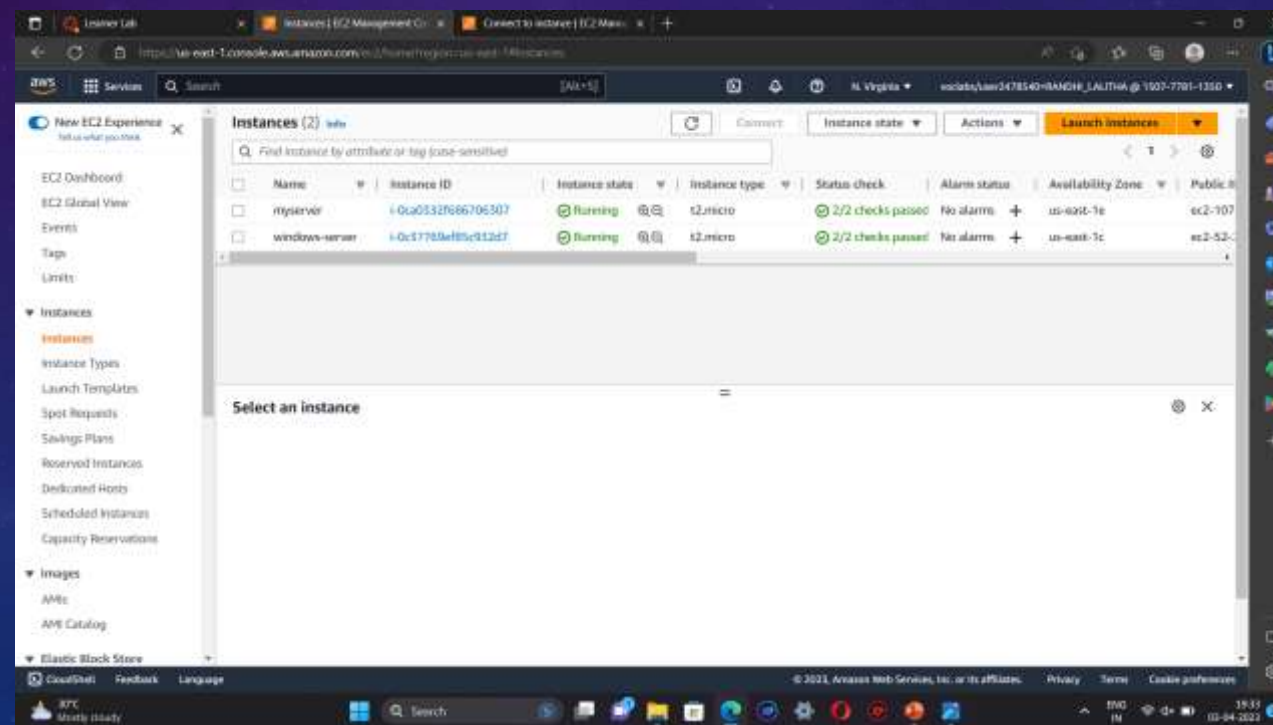
login as: ec2-user

Authenticating with public key "openssh"

Amazon Linux 2023

<https://aws.amazon.com/linux/amazon-linux-2023>

Last login: Mon Apr 3 13:20:42 2023 from 10.206.107.27  
[ec2-user@ip-172-31-54-87 ~]\$ whoami  
ec2-user  
[ec2-user@ip-172-31-54-87 ~]\$





# AWS ELASTIC LOAD BALANCING (ELB)

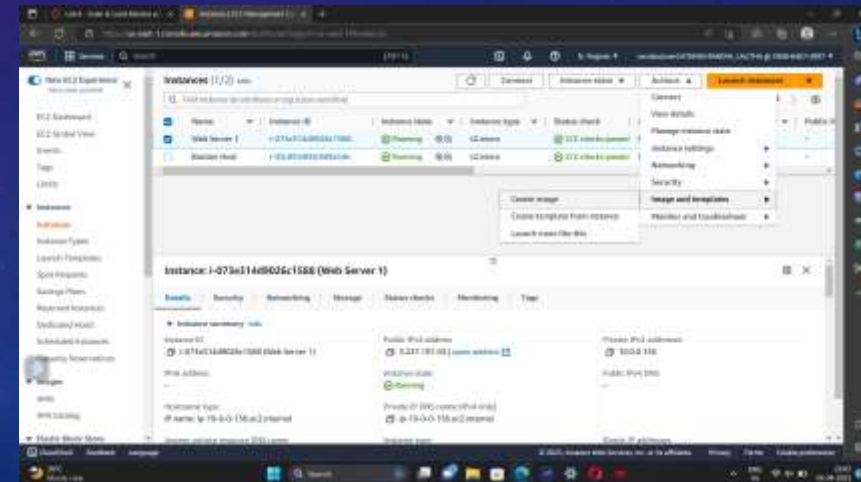
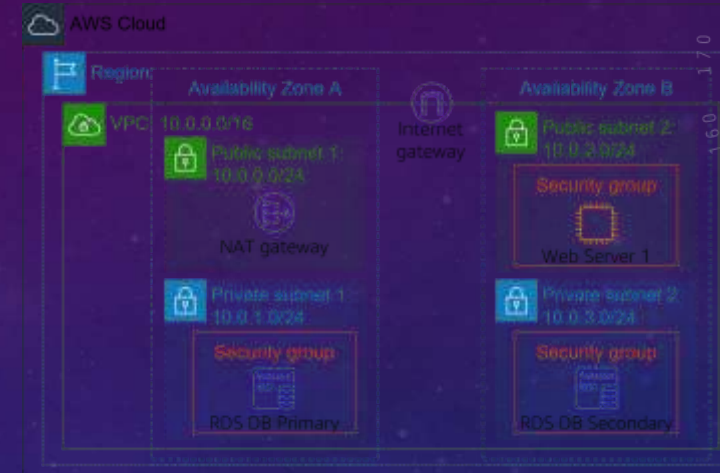
**Elastic Load Balancing** automatically distributes incoming application traffic across multiple Amazon EC2 instances

In this lab, We are provided with the given infrastructure.

Procedure:

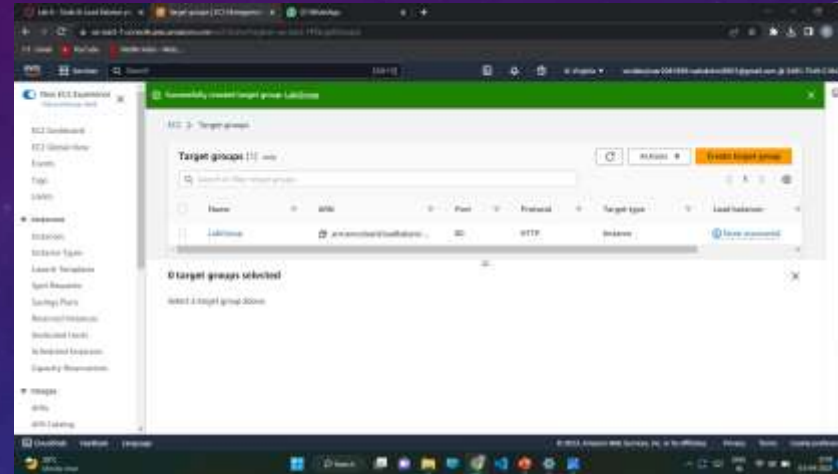
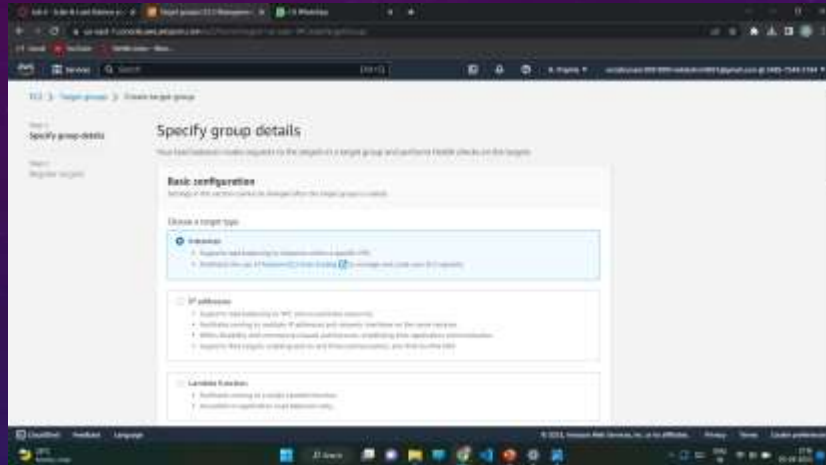
Task1: Creating an AMI for Auto Scaling

- ❖ Click start lab then click on AWS.
- ❖ You will navigate to AWS management console. Click on services and select EC2.
- ❖ Click instances. Make sure that **Status Checks** for **Web Server 1** displays 2/2 checks.
- ❖ Select Web Server 1 and in actions click images and templates > create image. Name the image and give the description.
- ❖ Click create image.

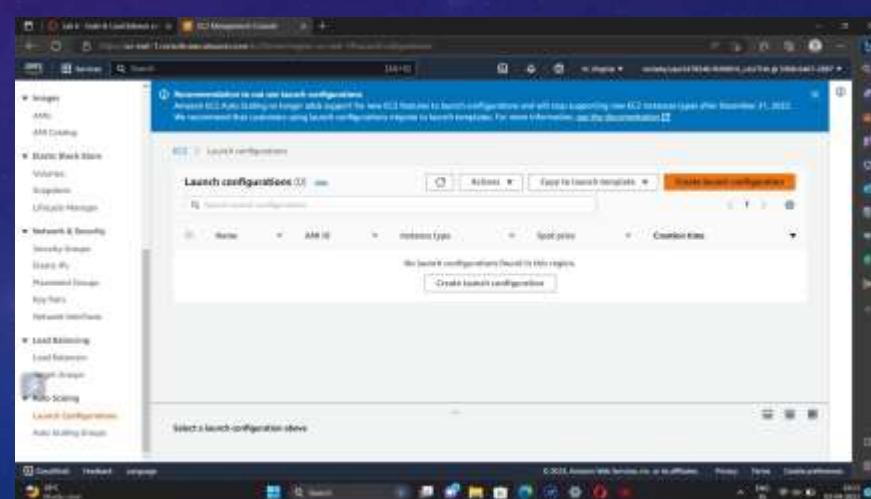
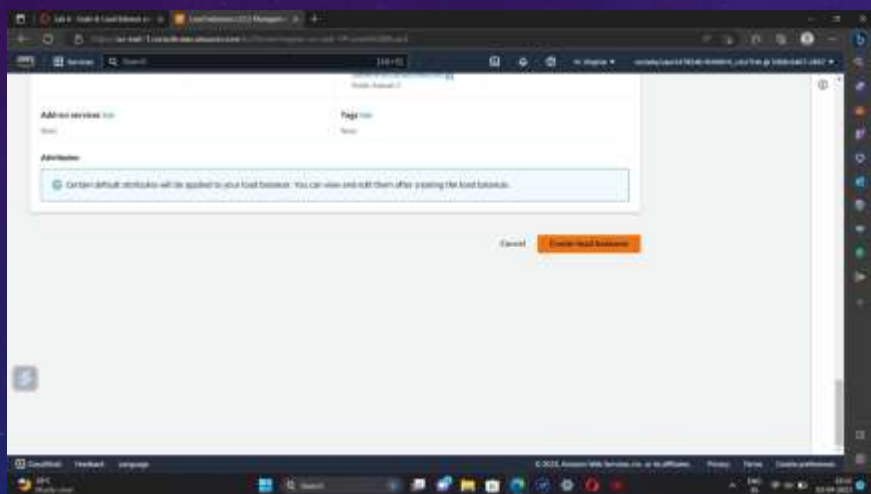
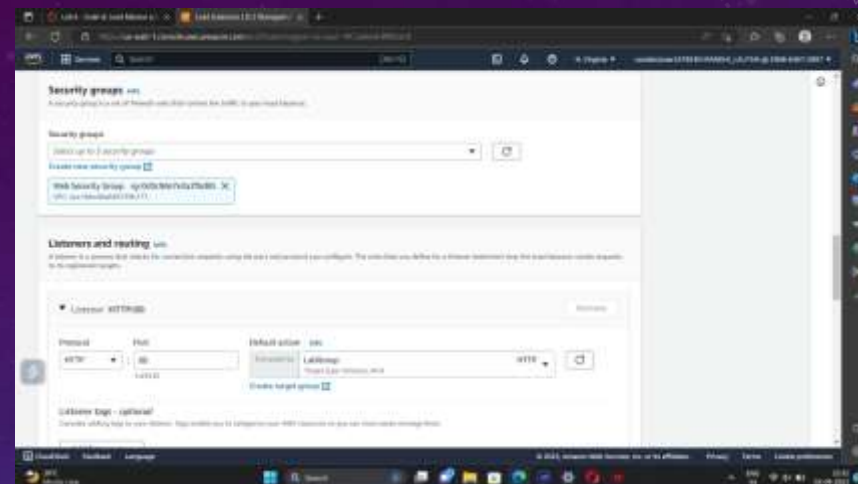
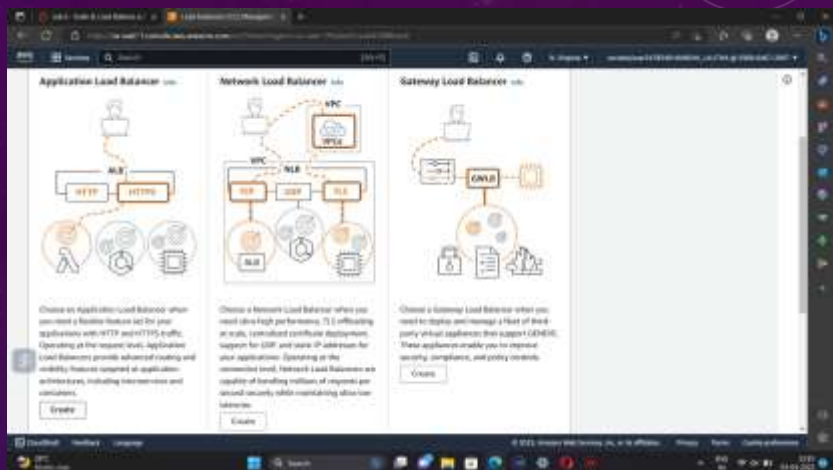


## Task 2: Creating a load balancer

- ❖ Choose Target Groups and then click on create target group.
- ❖ Select target type as instances. Name the target group. Select Lab VPC under VPC that is we are creating load balancer in Lab VPC .
- ❖ Choose next and then click on create target group.



- ❖ From the left navigation pane , select Load Balancers. Click create load balancer.
- ❖ To create a application balancer, click create under Application Load Balancer and Name it.
- ❖ In Networking mapping, select Lab VPC and specify the subnets that the load balancer should use.
- ❖ In security groups, select only Web Security Group and deselect all other than it .
- ❖ For the Listener HTTP:80 row, set the Default action to forward to **LabGroup**.

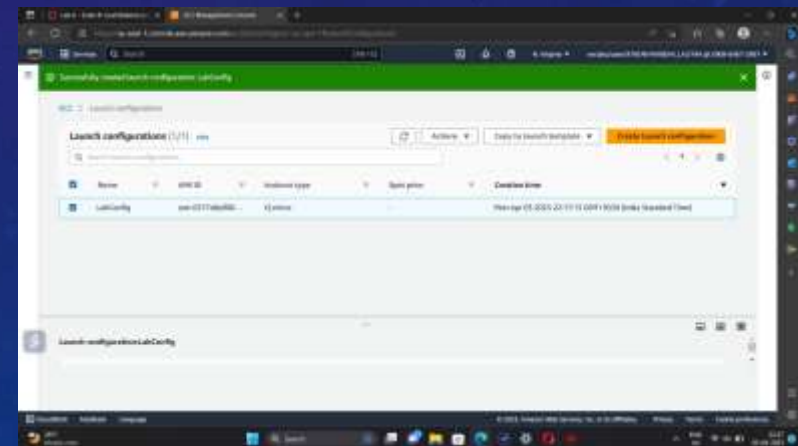
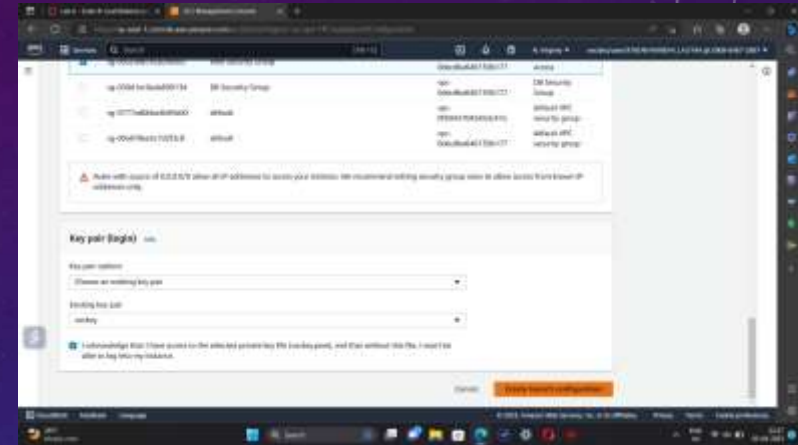




- ❖ Click create load balancer.

### Task 3: Create a Launch Configuration and an Auto Scaling Group

- ❖ In Launch Configurations, click create launch configuration.
- ❖ Name the configuration and for AMI choose web server AMI that you created in task 1.
- ❖ Select the instance type.
- ❖ Under Additional Configuration, for monitoring select Enable EC2 instance detailed monitoring within CloudWatch.
- ❖ Under security groups, choose an existing security group Web Security Group.
- ❖ Under key pair, choose an existing key pair vockey. Check I acknowledge...
- ❖ Click Create launch configuration.
- ❖ For created launch configuration, select create auto scaling group from actions.
- ❖ Name it and select Lab VPC under VPC, select the private subnets.
- ❖ Select an existing load balancer which was created earlier.
- ❖ In the **Additional settings - optional** pane, select **Enable group metrics collection within CloudWatch**



- ❖ Specify the values under Group size.
- ❖ Under **Scaling policies**, choose *Target tracking scaling policy* and name the policy. Specify metric type and target value. Then add a tag and click create auto scaling group.

The screenshot shows the 'Create Auto Scaling group' wizard in the AWS Management Console, specifically Step 6: Add tags. The interface is in English and shows the following sections:

- Instance scale-in protection:** A section with a checkbox labeled 'Enable instance protection from scale-in' which is currently unchecked.
- Step 5: Add notifications:** A section with an 'Edit' button and a 'Notifications' box containing the text 'No notifications'.
- Step 6: Add tags:** A section with an 'Edit' button and a 'Tags (1)' table.

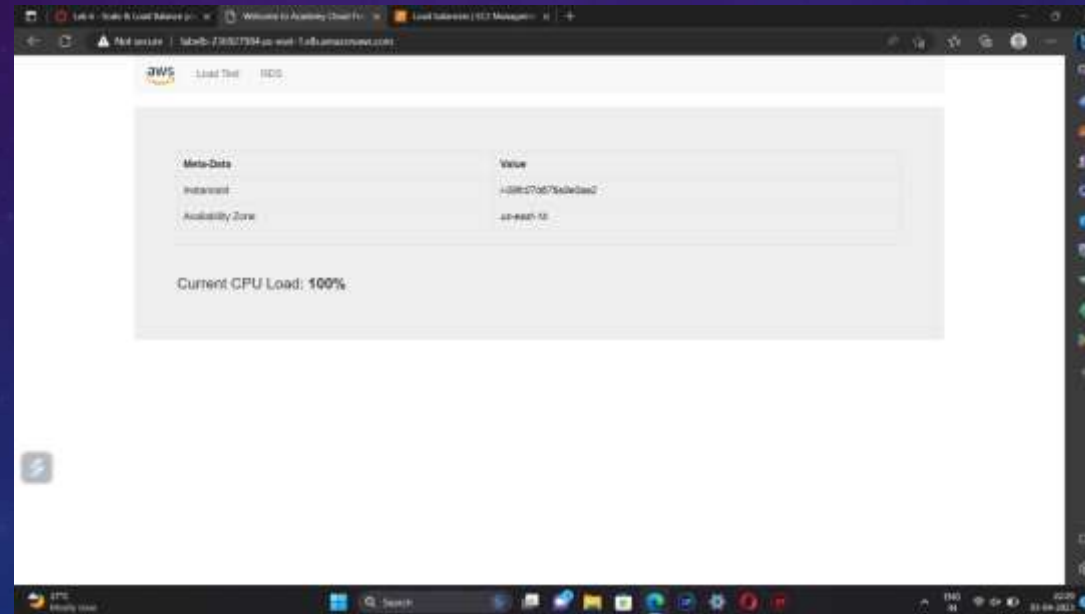
Key	Value	Tag new instances
Name	Lab Instance	Yes

At the bottom of the wizard, there are three buttons: 'Cancel', 'Previous', and 'Create Auto Scaling group' (highlighted in orange).

The bottom of the screenshot shows the Windows taskbar with the date '03-04-2022' and time '20:23'.

#### Task 4: Verify that Load Balancing is Working

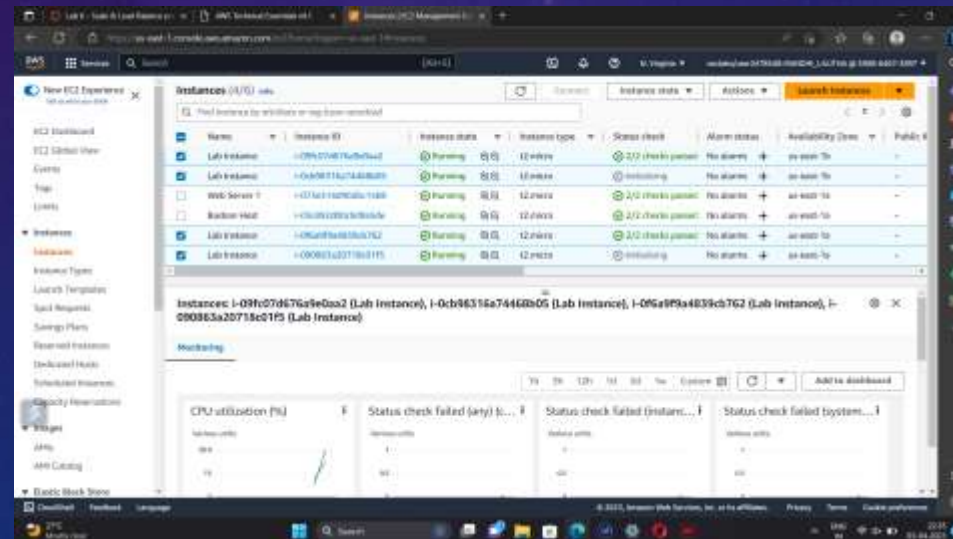
- ❖ click **Instances**. You should see two new instances named **Lab Instance**. These were launched by Auto Scaling.
- ❖ In the labgroup target group, two **Lab Instance** targets should be listed for this target group. Wait until the **Status** of both instances transitions to *healthy*.
- ❖ Now copy the DNS name of the created load balancer making sure to omit "(A Record)". and paste it in a new browser
- ❖ The application should appear in your browser. This indicates that the Load Balancer received the request, sent it to one of the EC2 instances, then passed back the result.





## Task 5: Test Auto Scaling

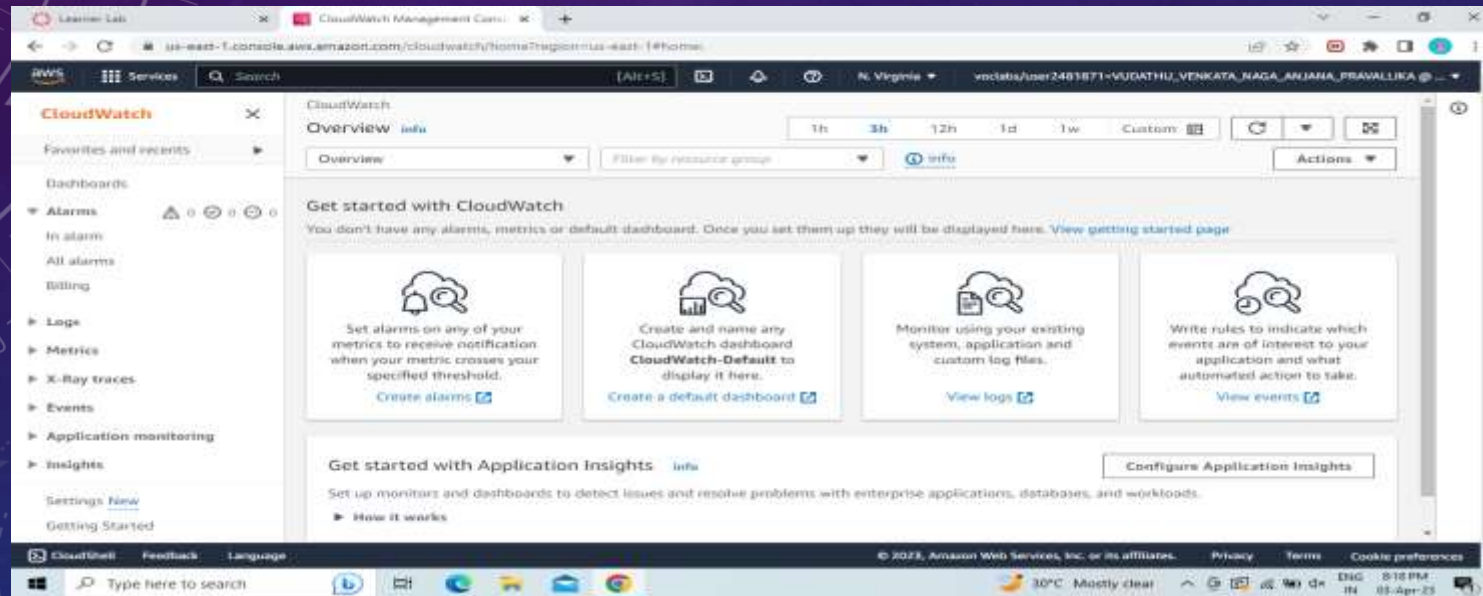
- ❖ On the **Services** menu, click **CloudWatch**. In the left navigation pane, choose **All alarms**. Two alarms will be displayed. These were created automatically by the Auto Scaling group.
- ❖ From services select EC2 and choose Auto Scaling Groups and select Lab Auto Scaling Group which you created.
- ❖ choose the **Automatic Scaling** tab. Select **LabScalingPolicy** and from actions change the target value to 50. click update.
- ❖ To go cloudwatch and click all alarms and verify.
- ❖ Click the **OK** alarm, which has *AlarmHigh* in its name. Return to the browser tab with the web application. Click **Load Test** beside the AWS logo. This will cause the application to generate high loads.
- ❖ You should see the **AlarmHigh** chart indicating an increasing CPU percentage. Once it crosses the 60% line for more than 3 minutes, it will trigger Auto Scaling to add additional instances.
- ❖ In EC2 instances, you notice that more than two instances labeled **Lab Instance** should now be running.
- ❖ Finally terminate the Web Server 1.



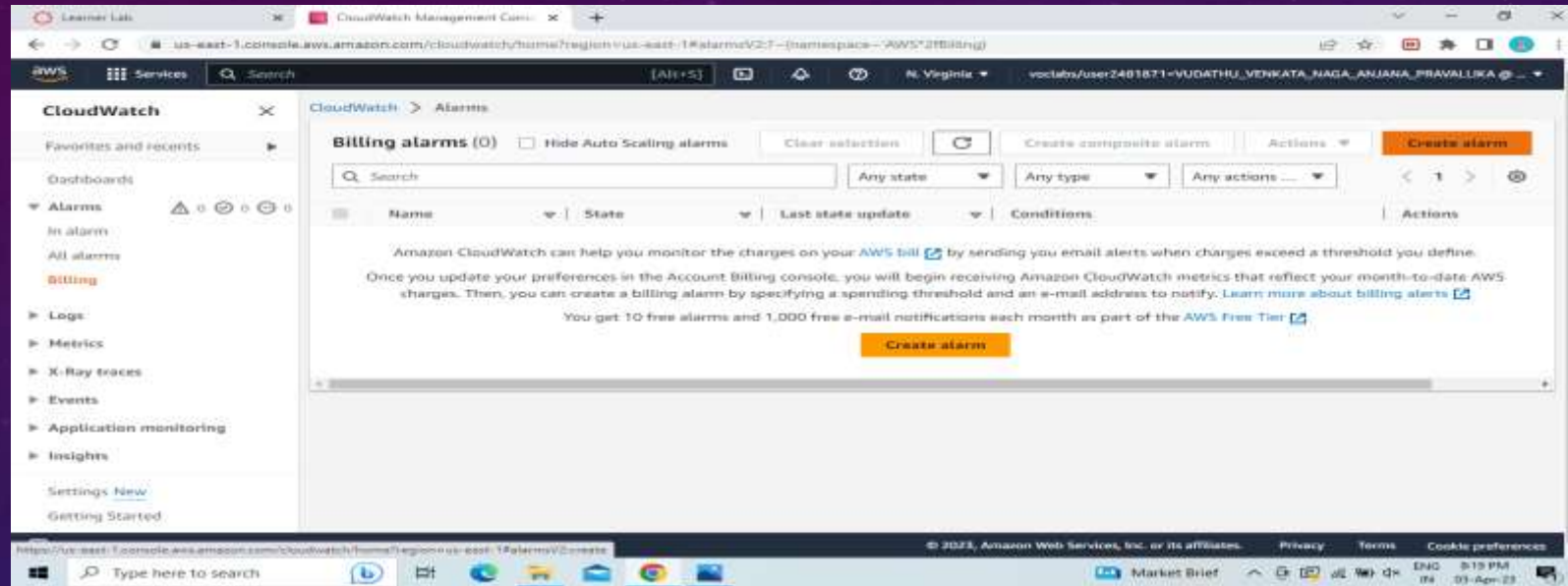
# AWS CLOUDWATCH

## PROCEDURE

1. Go to AWS Services, Click on CloudWatch and then in the Dashboard go to Alarms section and select Billings.



2. Then Click on CREATE ALARM.



3. Then follow the steps.

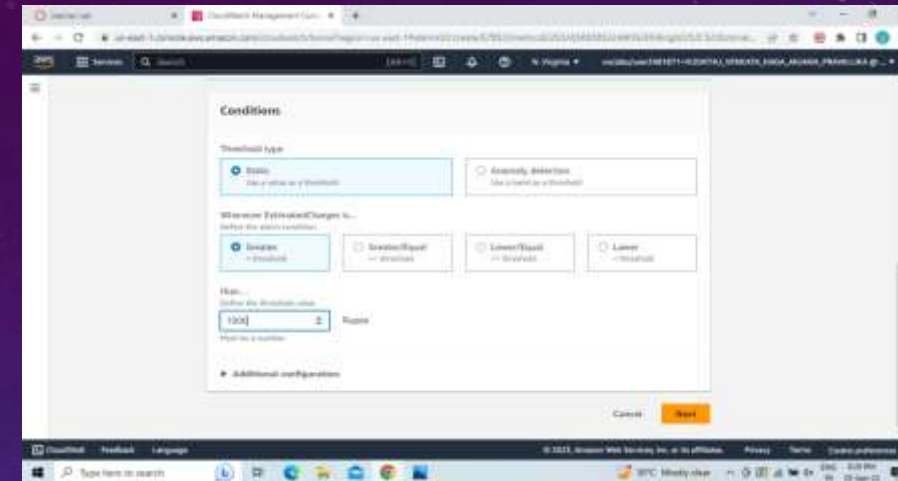
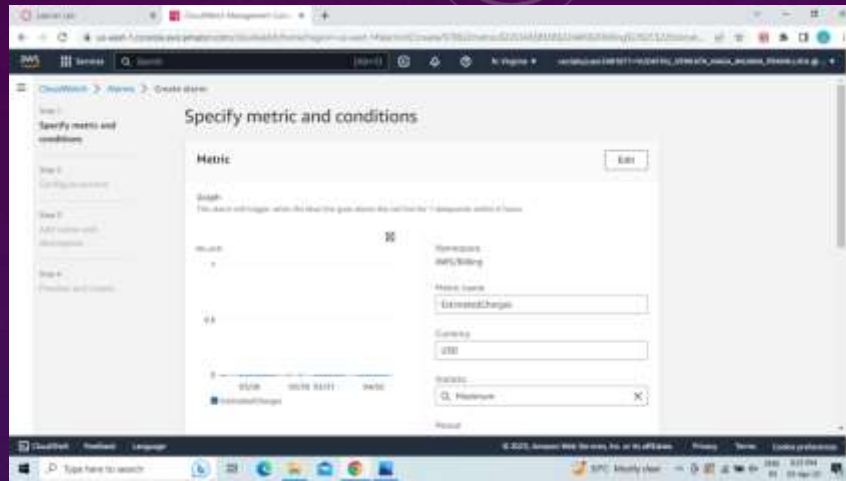
In the first step it will ask us to Specify metric and conditions. Click on Select Metric.

Change the Currency to Rupee.

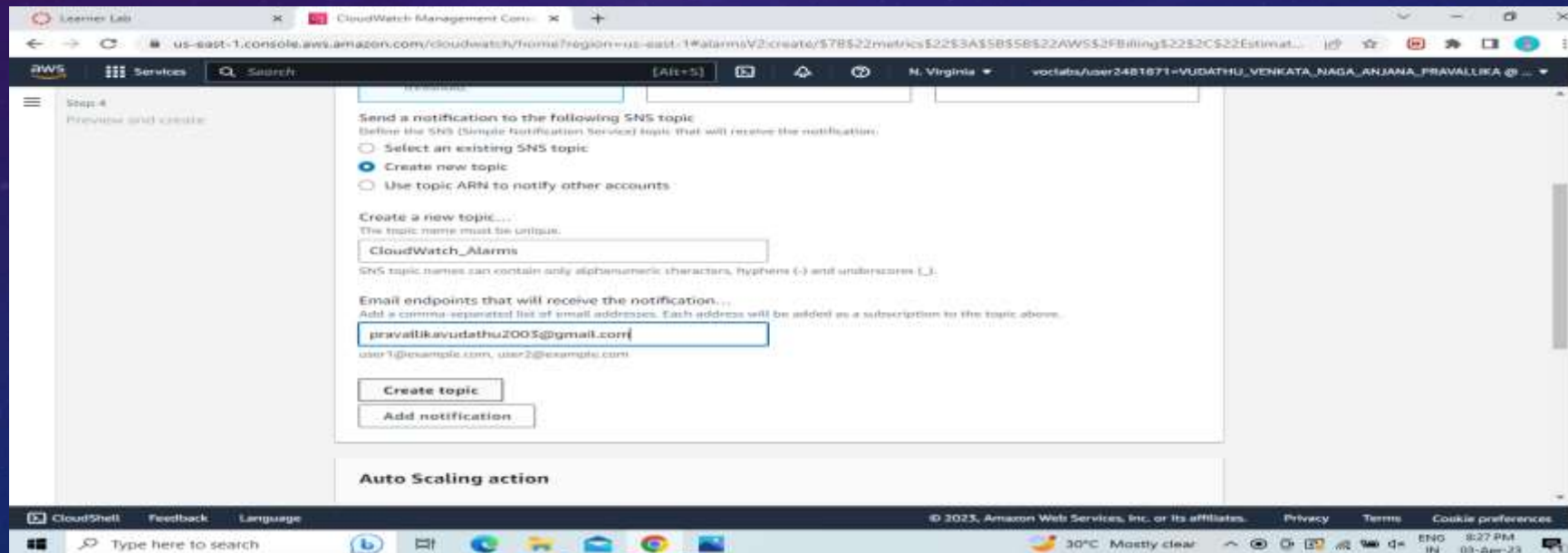
In the Conditions section choose the EstimatedCharges like Greater/GreaterEqual/Lowerequal/Lower and also define the threshold value.

4. Click on Next.

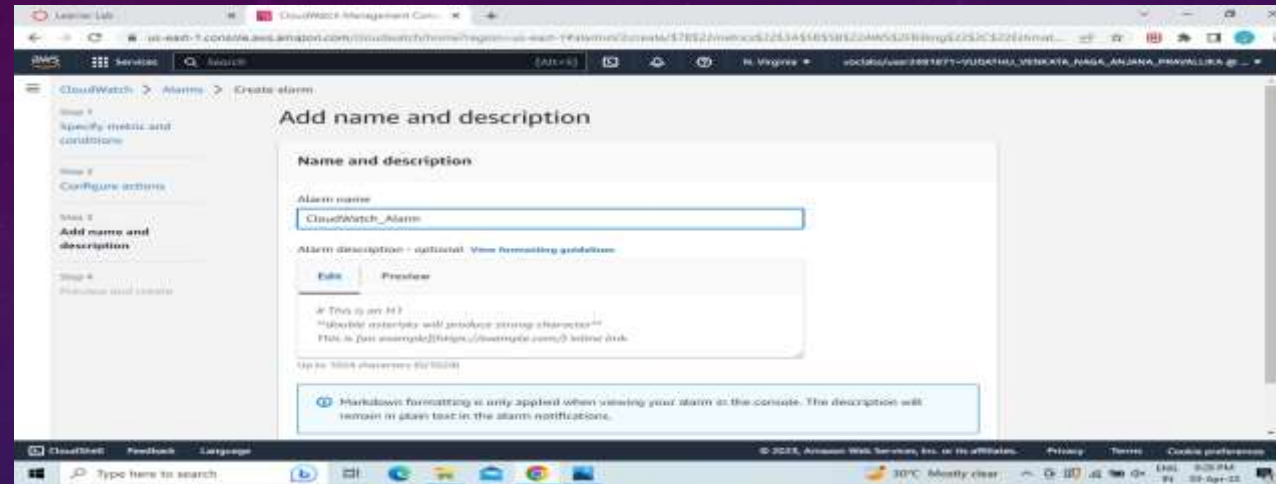




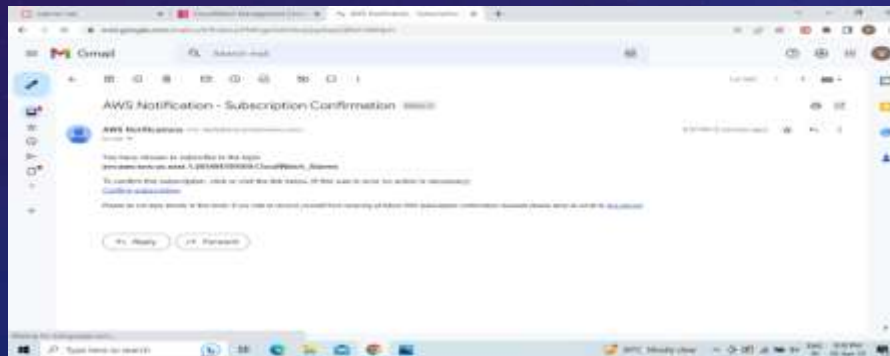
5.Now for Cofigure Actions choose Create new topic.Give a name to the topic and enter your email to receive a notification.Click on Create Topic,then Next.



6. Give a name to your Alarm and Click on next.



7. You will get a AWS Notification-Subscription Confirmation mail to the email which you have provided. Click on Confirm Subscription. Then it will open a window showing Subscription Confirmed.



8. Preview the details you have entered .

9. Click on Create alarm. This will Create your Alarm.

The screenshot displays the AWS CloudWatch console interface. At the top, a green banner indicates "Successfully created alarm CloudWatch\_Alarm." with a "View alarm" button. The left sidebar shows the "Alarms" section selected. The main content area, titled "Billing alarms (1)", includes a search bar, filters for "Any state" and "Any type", and a table of alarms. The table lists one alarm, "CloudWatch\_Alarm", with a state of "Insufficient data", a last state update of "2023-04-03 20:30:53", and conditions "EstimatedCharges > 1000 for 1 datapoints within 6 hours". The alarm is marked as "Actions enabled". The bottom of the screen shows the Windows taskbar with the time 8:31 PM on 03-Apr-23.

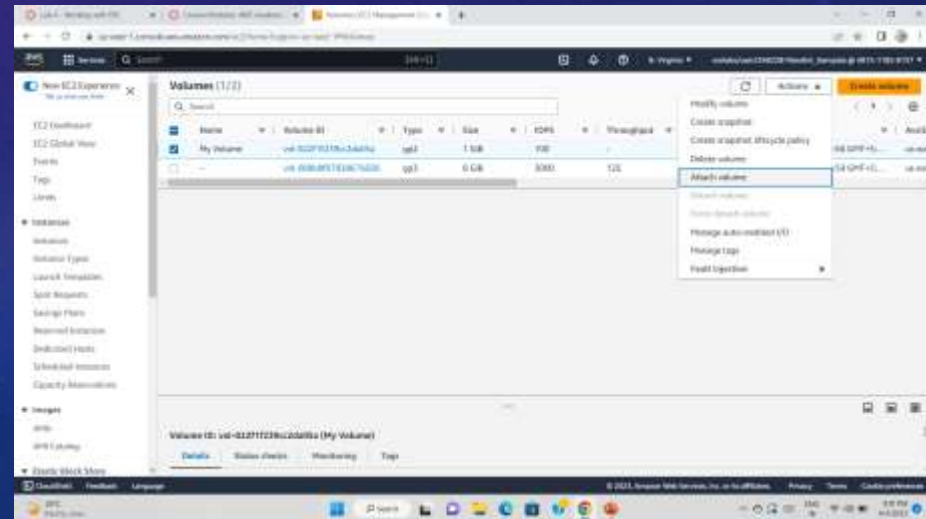
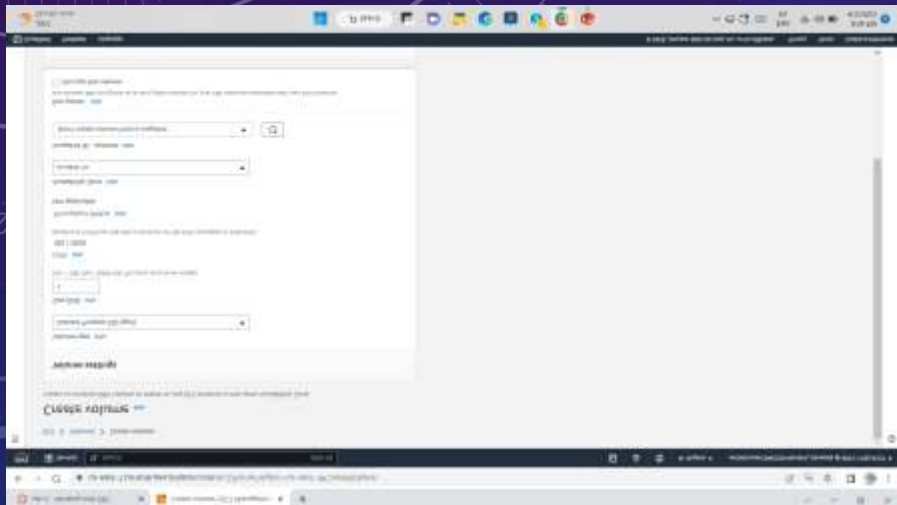
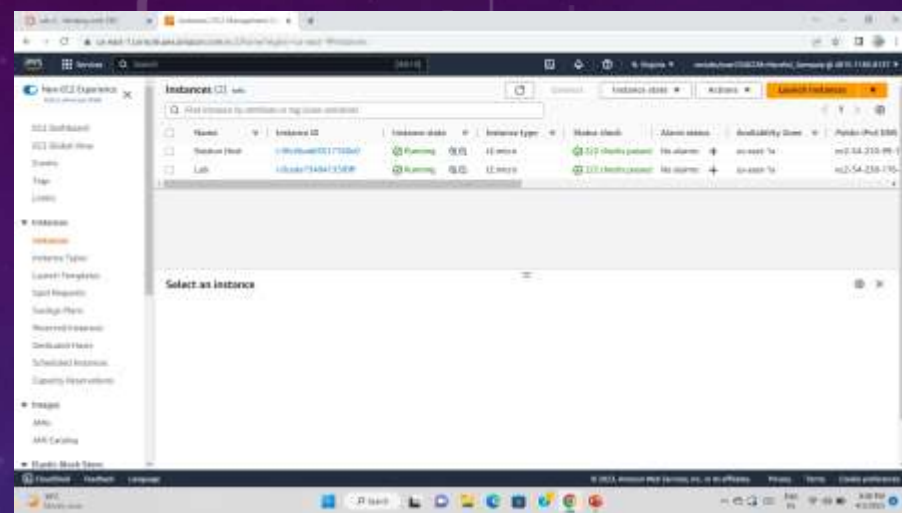
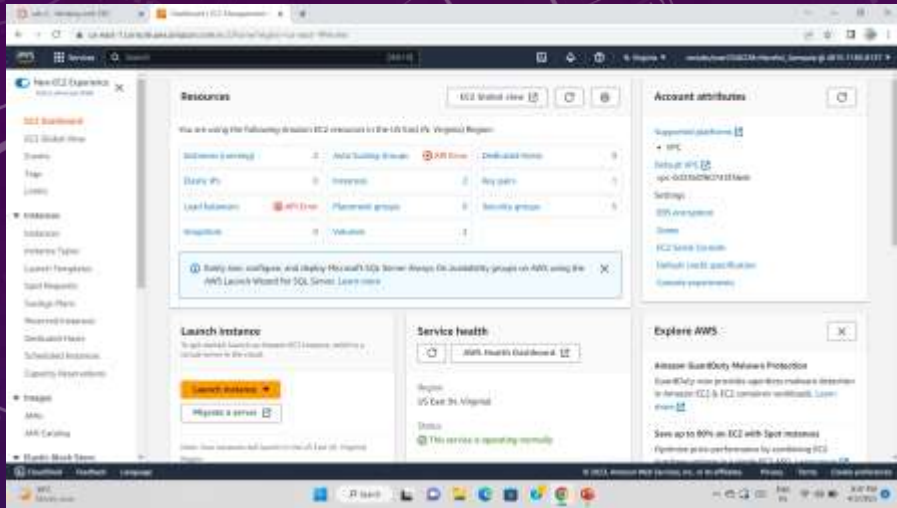
Name	State	Last state update	Conditions	Actions
CloudWatch_Alarm	Insufficient data	2023-04-03 20:30:53	EstimatedCharges > 1000 for 1 datapoints within 6 hours	Actions enabled

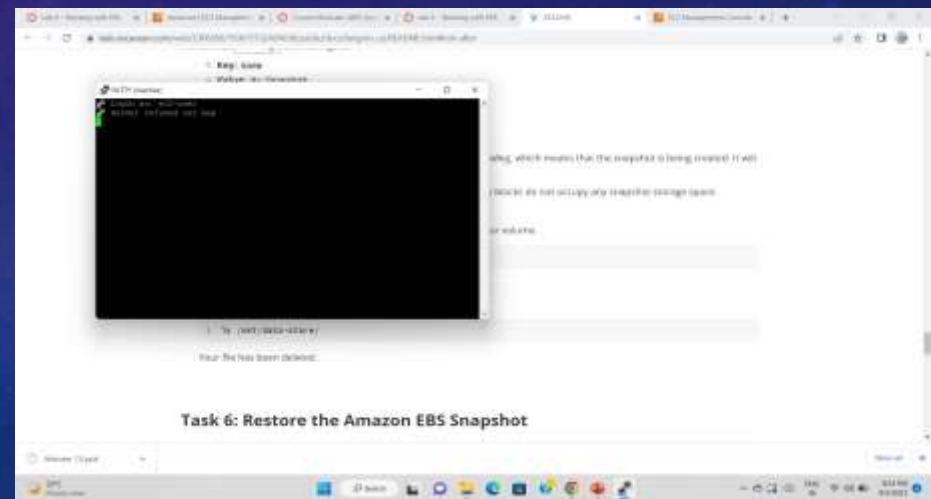
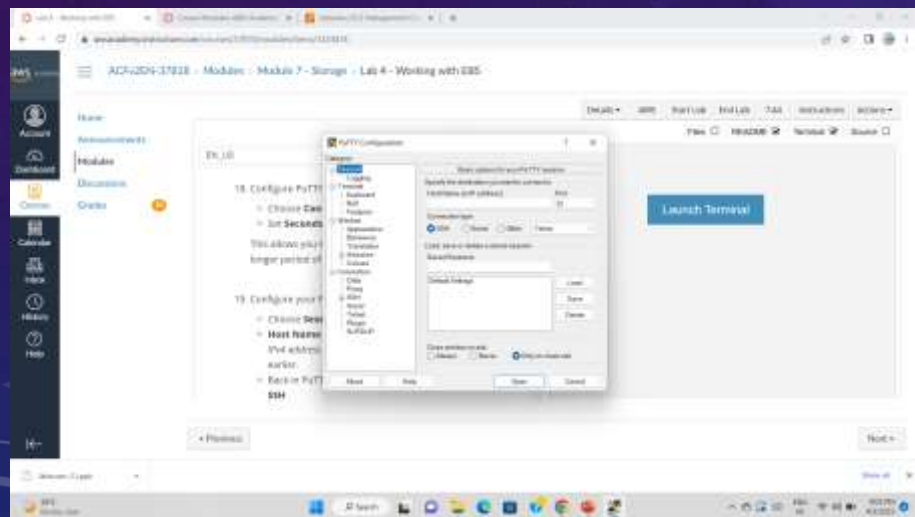
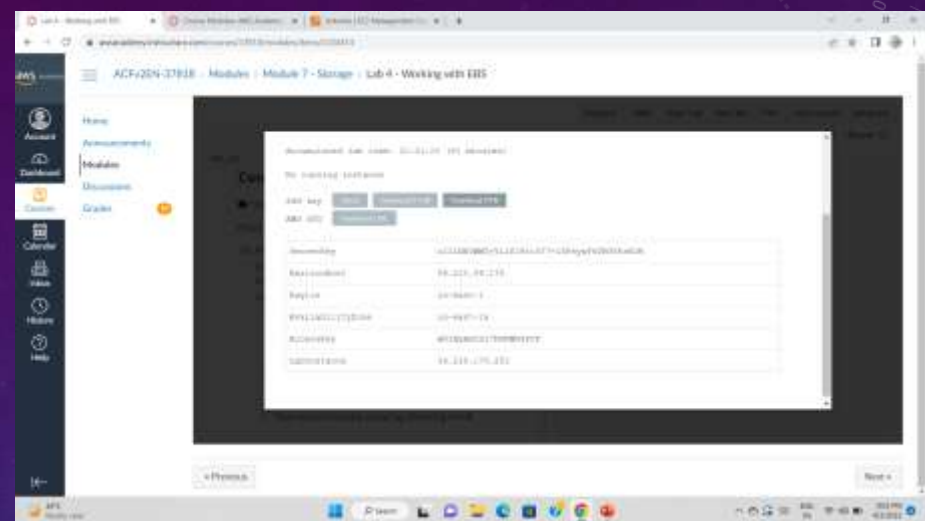
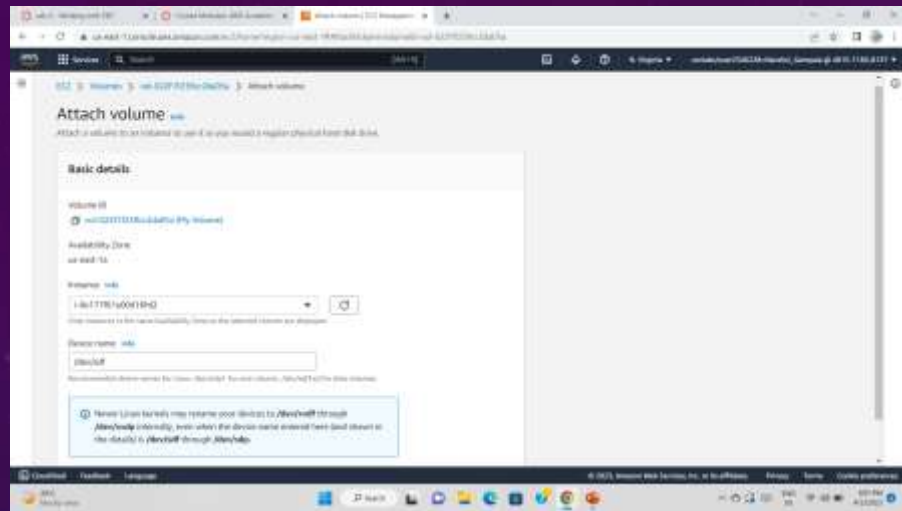


# AWS EBS(ELASTIC BLOCK STORE)

## CREATING A EBS VOLUME

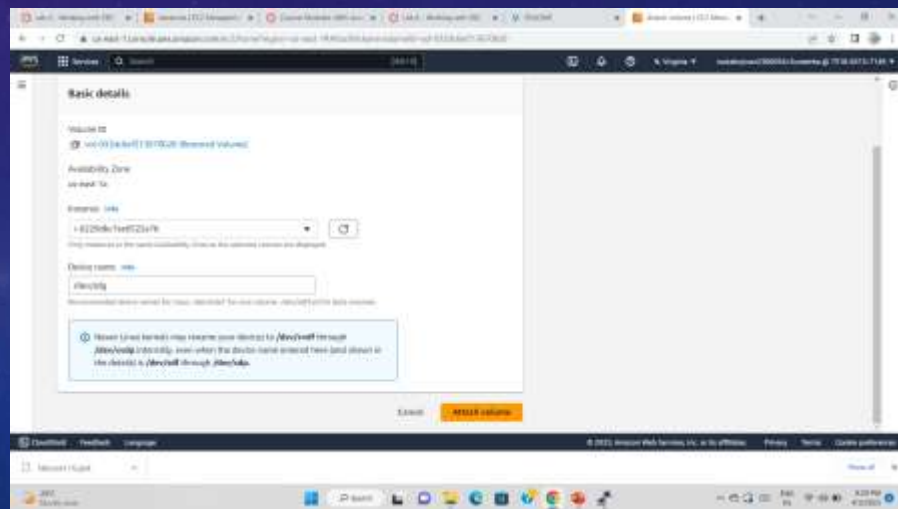
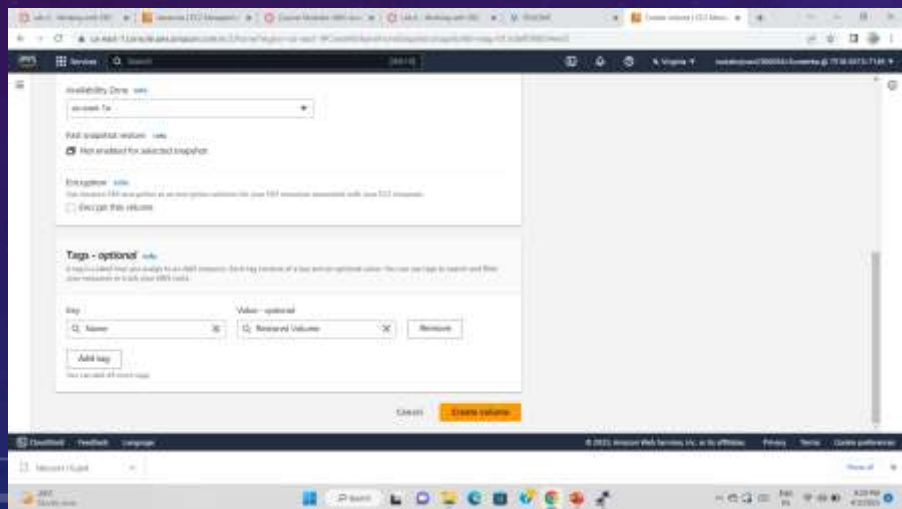
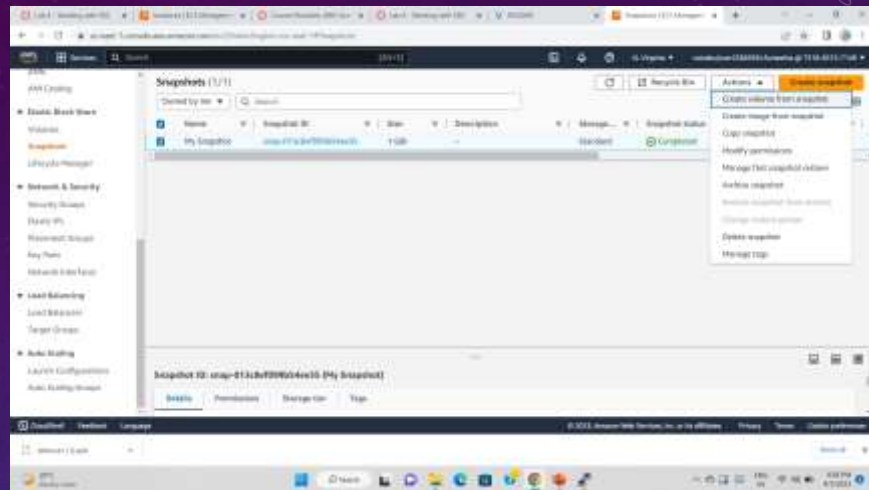
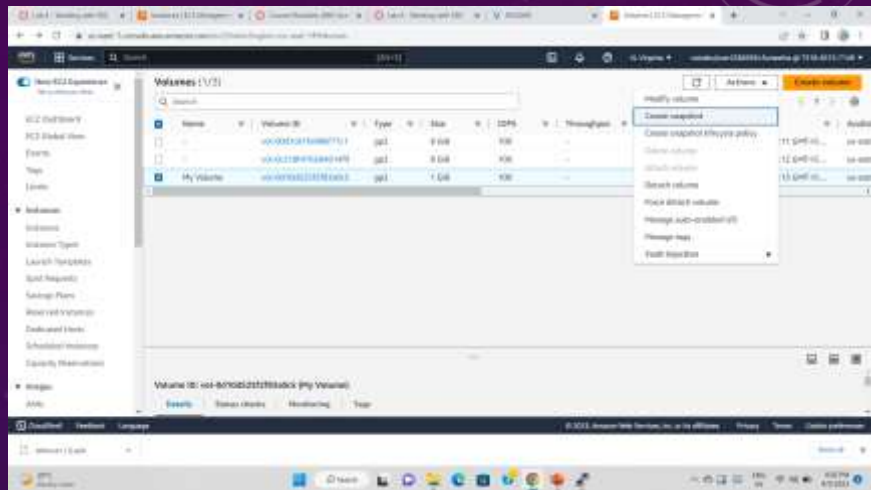
1. Open Management Console, on the services menu open Ec2
2. In the left navigation pane choose instances and create a instance with a name
3. Next, In the left navigation pane choose Volumes
4. Click on Create Volume
5. Select volume type, size(Gib),Availability Zone and in Add tag section add key and value names.
6. Then click on create volume
7. Click on volumes on left navigation pane select the created volume and attach a previously created instance to it.
8. Then, go to “Details” drop down, choose “show”
9. Download the ppk file
10. Download putty
11. Open putty set the fields (such as Host name, public key, private key) as per the requirement.
12. The putty shell will open , then login into it and run the commands.
13. The commands looks like:  
`df -h`  
`sudo mkfs -t ext3/dev/sdf` etc.,
14. Create a EBS snapshot by giving the necessary fields.
15. Create a volume using snapshot.
16. Attach the volume to the created EC2 instance





Task 6: Restore the Amazon EBS Snapshot



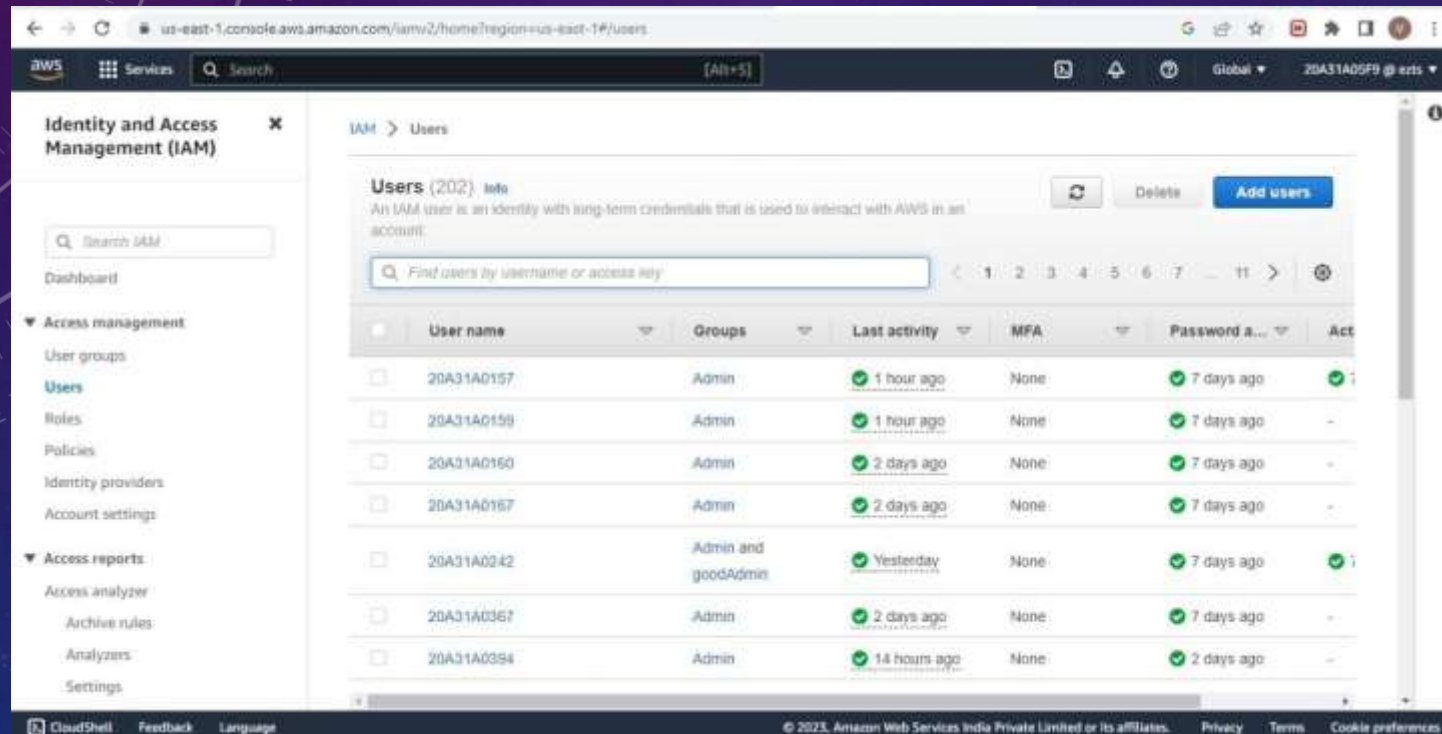


# AWS COMMAND LINE INTERFACE

STEP 1 - Download and install AWS CLI and complete the installation steps.

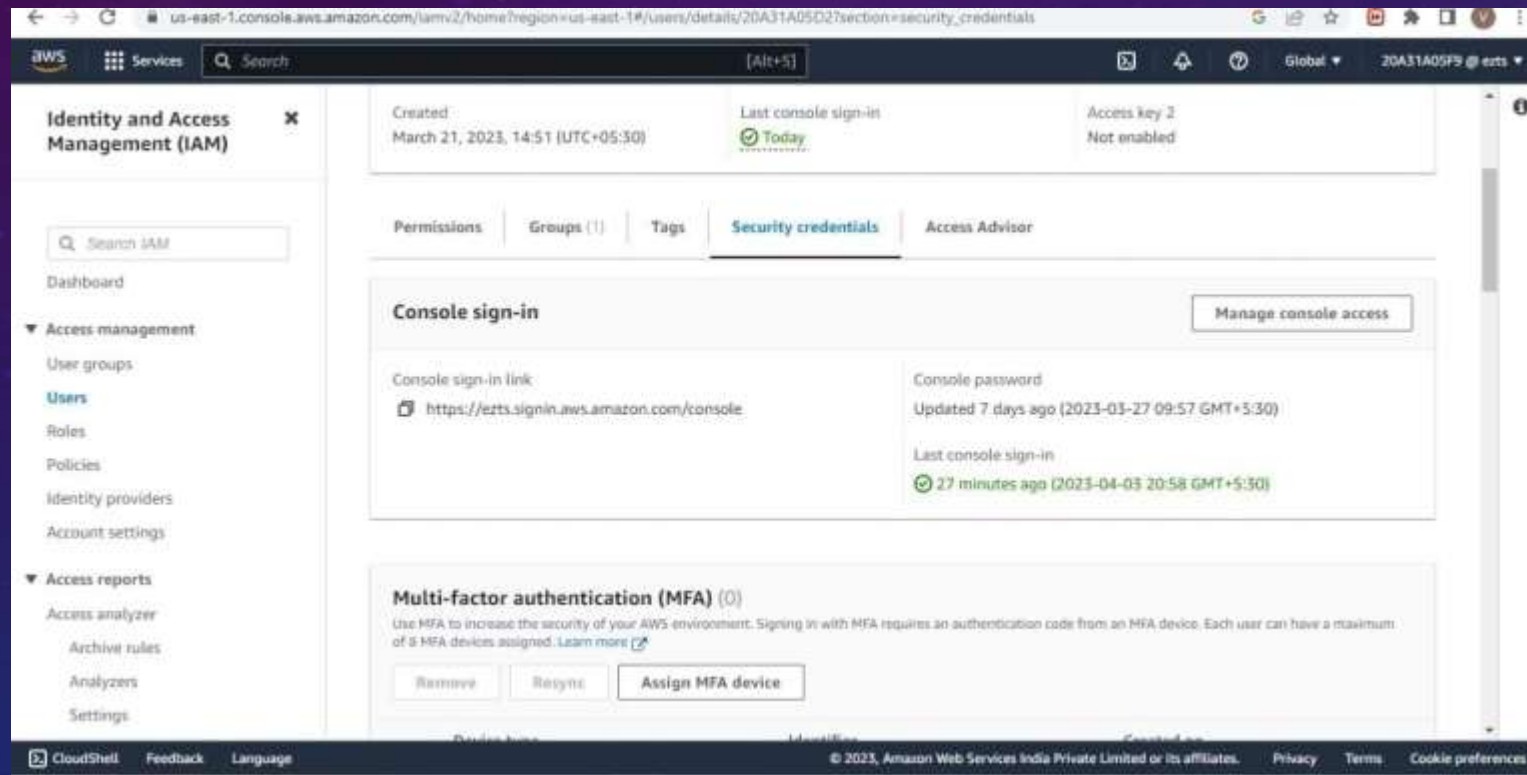
STEP 2 - Login to AWS Management Console and search for IAM.

STEP 3 - In the navigation pane ,select Users



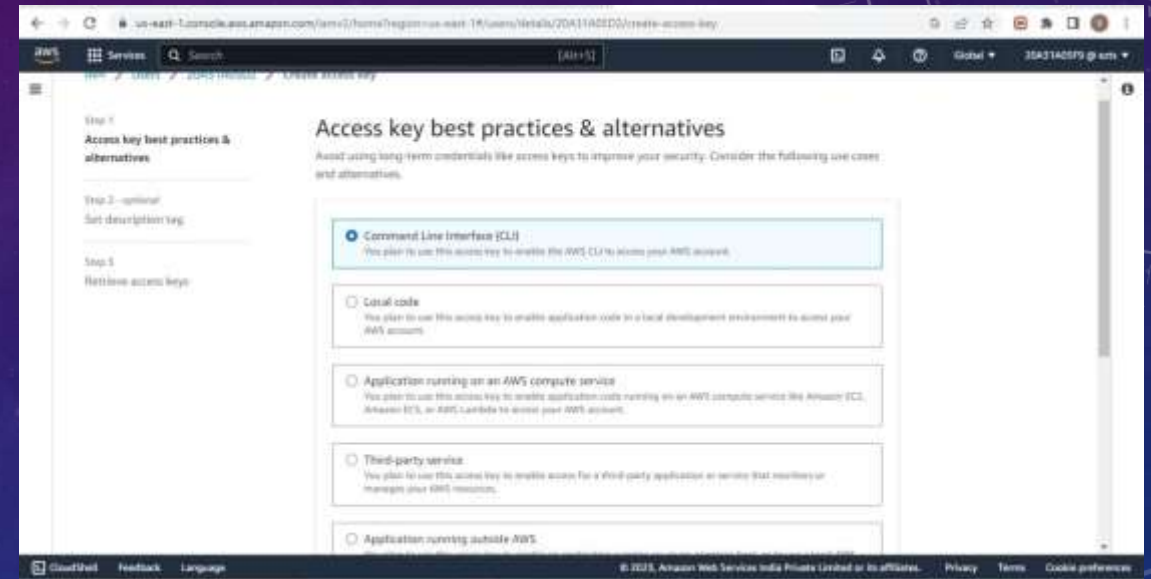
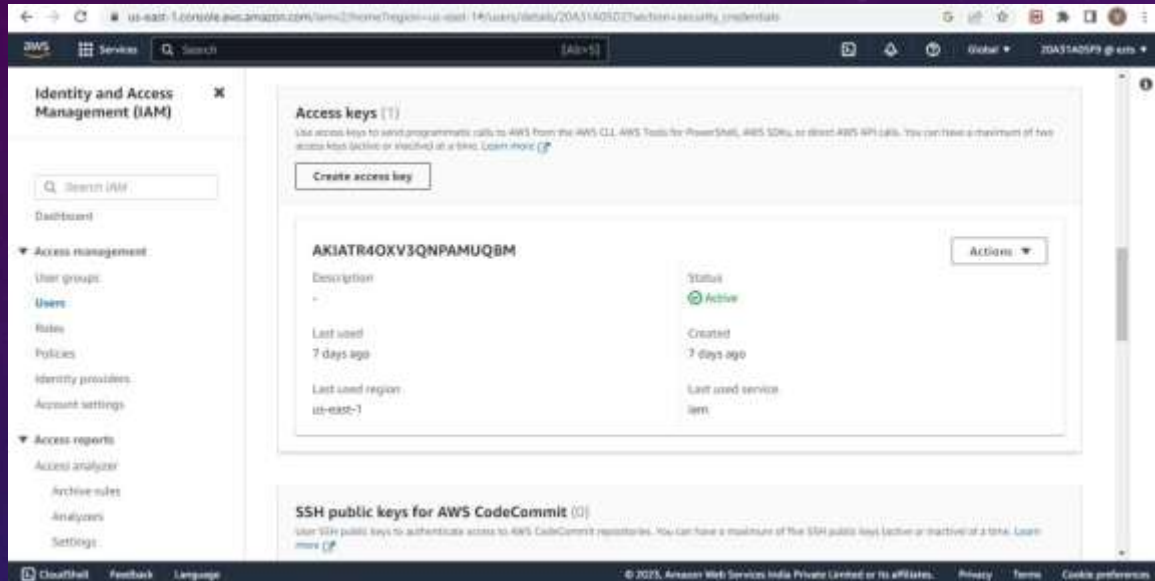
STEP 4 - In the users select the name of the user whose access keys you want to create.

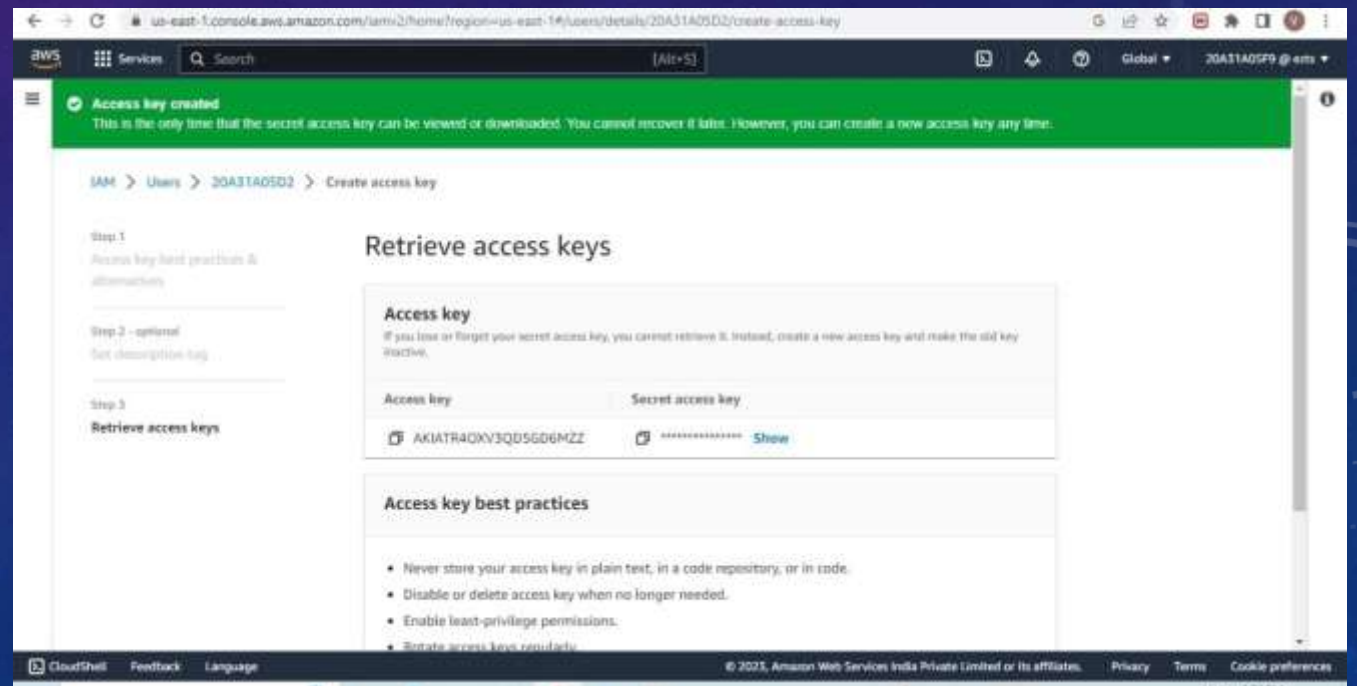
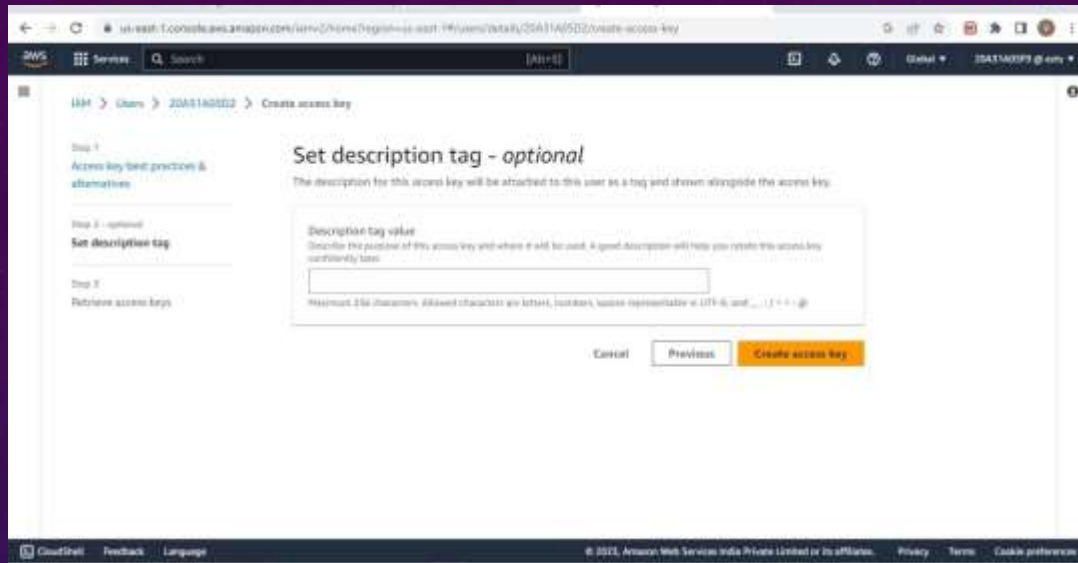
STEP 5 - Click on Security Credentials tab.





STEP 6 - In the access Keys section , choose Create access key.





STEP 6 – Now you can use this access key to configure CLI

STEP 7 - Open Command Line Interface and run the following command

>aws configure

After entering this command AWS CLI prompts us with four pieces of information

1. Access Key ID: (enter your ID)
2. Secret Access Key: ( enter your key)
3. AWS Region: (enter the desired region )
4. Output Format: (enter the desired output)

```
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

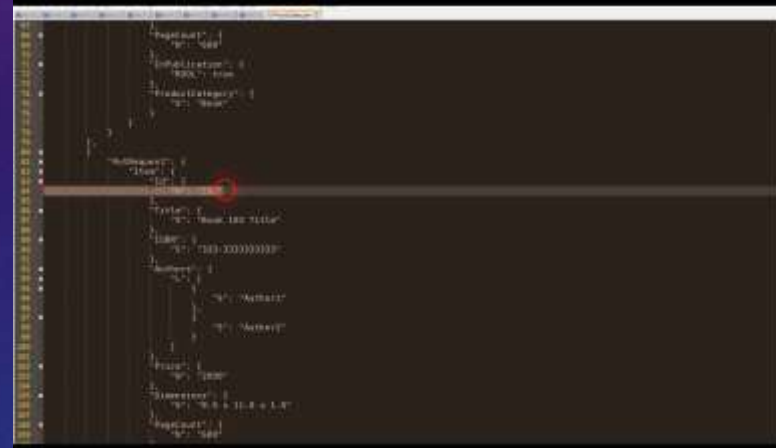
C:\Users\sivas>aws configure
AWS Access Key ID [None]: AKIATR40XV3QD5GD6MZZ
AWS Secret Access Key [None]: vMQP4GL99CbDSxsPWSgiTkkozMiRsUUZ0i+hDdNT
Default region name [None]: us-east-1
Default output format [None]: json
```

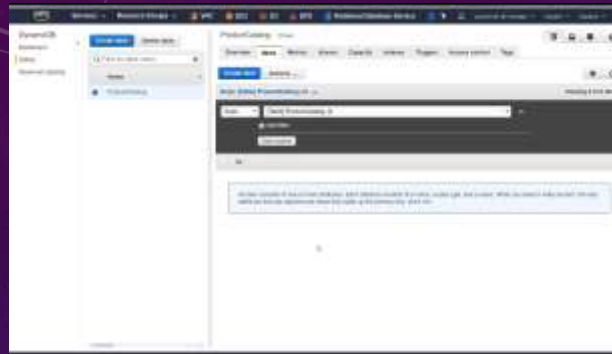
Finally we get Javascript Object Notation of all the users as output.



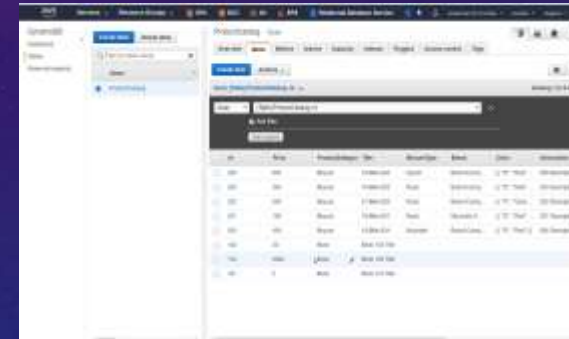
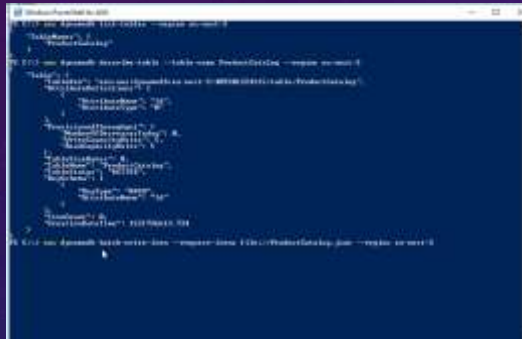
# AWS DYNAMO DB

- Setting up the Amazon DynamoDB
- here, we will be having an JSON file which is a product catalog
- the products have a lot of different attributes and **id** is only common.
- the interface looks like this:

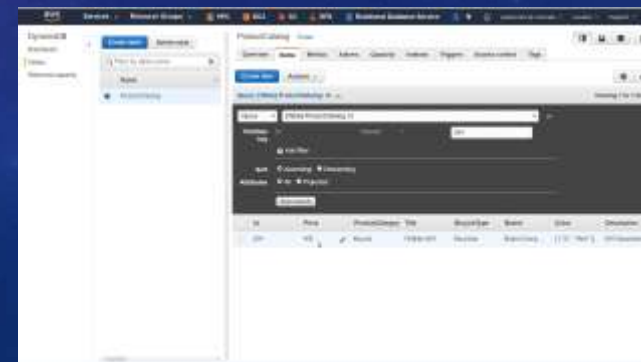
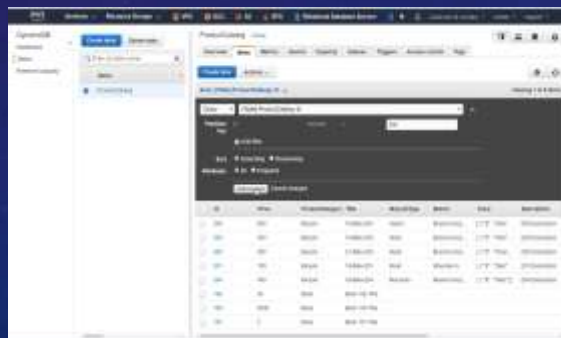




- After creating the table , we can see that there are no items present.



- So we will use the CLI to populate the table. Open powershell of AWS.



# AWS LIGHT SAIL

## **PROCEDURE:**

- 1.ON THE HOME PAGE, CHOOSE CREATE INSTANCE.
- 2.SELECT A LOCATION FOR YOUR INSTANCE (AN AWS REGION AND AVAILABILITY ZONE).CHOOSE CHANGE REGION AND ZONE TO CREATE YOUR INSTANCE IN ANOTHER LOCATION.
- 3.OPTIONALLY, YOU CAN CHANGE THE AVAILABILITY ZONE.CHOOSE AN AVAILABILITY ZONE FROM THE DROPDOWN LIST.
- 4.PICK AN APPLICATION (APPS + OS) OR AN OPERATING SYSTEM (OS ONLY).
- 5.CHOOSE YOUR INSTANCE PLAN.
- 6.ENTER A NAME FOR YOUR INSTANCE.

## **RESOURCE NAMES:**

1. Must be unique within each AWS Region in your Lightsail account.
2. Must contain 2 to 255 characters.
3. Must start and end with an alphanumeric character or number.
4. Can include alphanumeric characters, numbers, periods, dashes, and underscores.



7. Choose one of the following options to add tags to your instance:

- Add key-only tags or Edit key-only tags (if tags have already been added). Enter your new tag into the tag key text box, and press Enter. Choose Save when you're done entering your tags to add them, or choose Cancel to not add them.



Key-only tags

Version 1 × Customer 1

Add a tag key and press Enter.

Save Cancel

- Create a key-value tag, then enter a key into the Key text box, and a value into the Value text box. Choose Save when you're done entering your tags, or choose Cancel to not add them. Key-value tags can only be added one at a time before saving. To add more than one key-value tag, repeat the previous steps.



Key-value tags

Key Value

Project → Earth

Save Cancel

8. Choose Create instance.

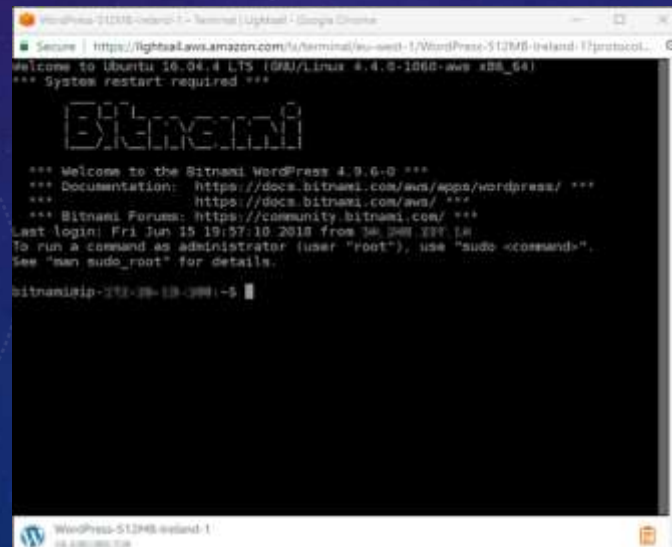
Within minutes, your Lightsail instance is ready and you can connect to it via SSH, without leaving Lightsail!

1. FROM THE LIGHTSAIL HOME PAGE, CHOOSE THE MENU ON THE RIGHT OF YOUR INSTANCE'S NAME, AND THEN CHOOSE CONNECT.



Alternately, you can open your instance management page and choose the Connect tab.

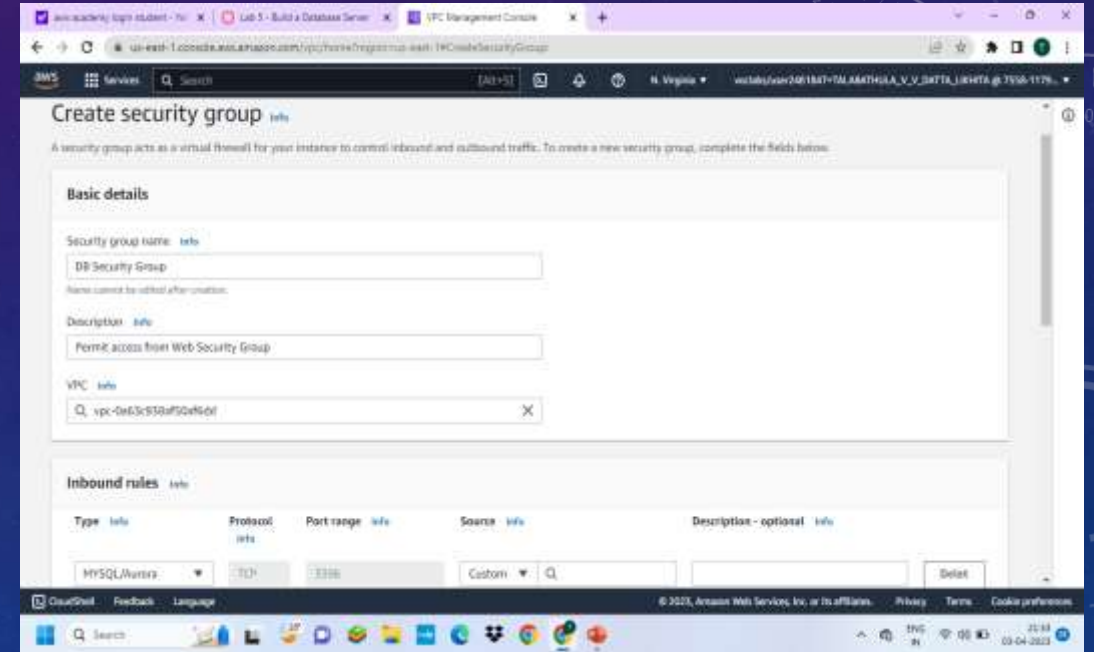
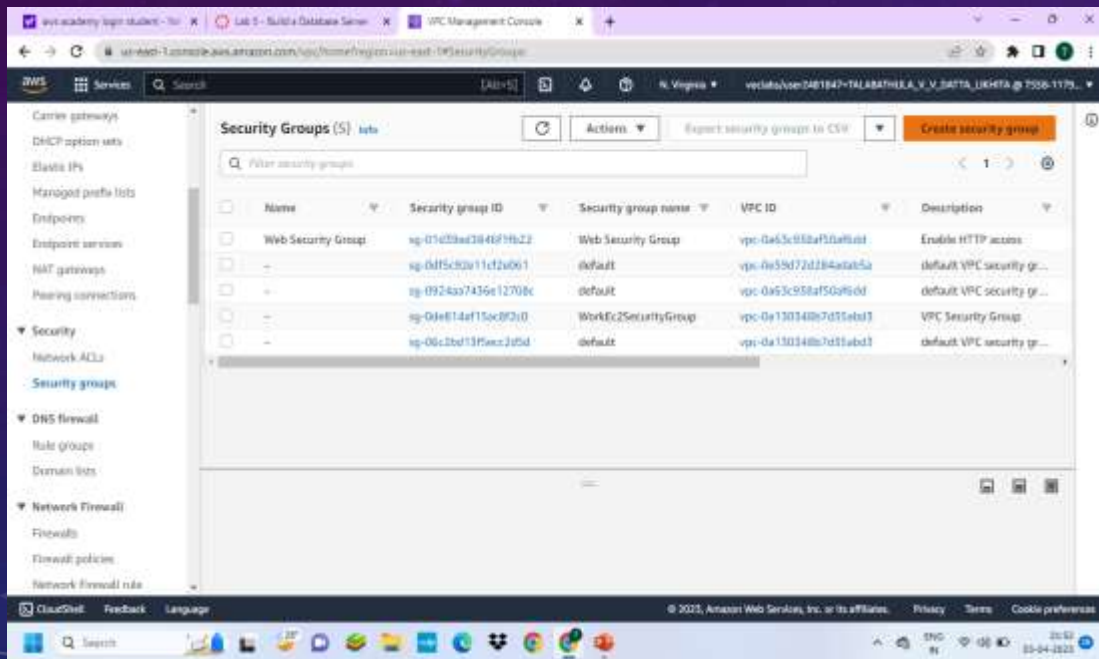
2. You can now type commands into the terminal and manage your Lightsail instance without setting up an SSH client.



# AWS RDS

**Step 1:** Create a Security Group for the RDS DB Instance.

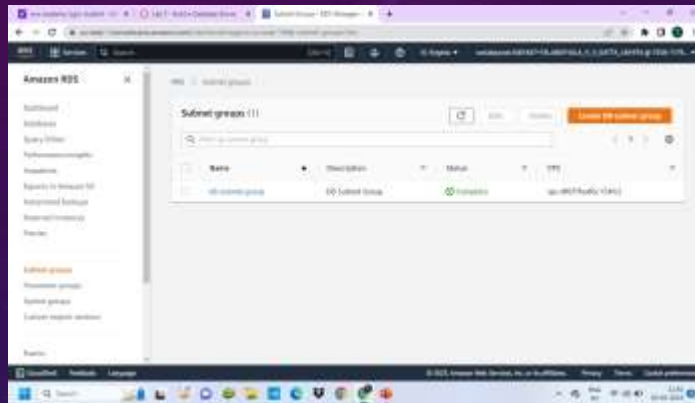
aws management console → vpc → security groups → choose create security group → add inbound rule → create security group.



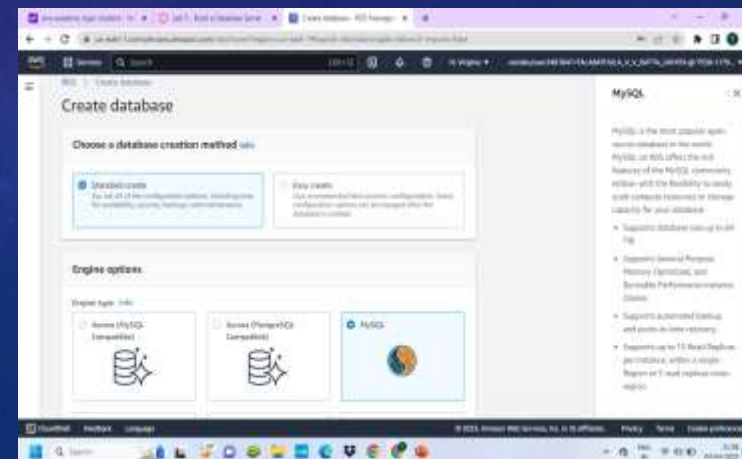
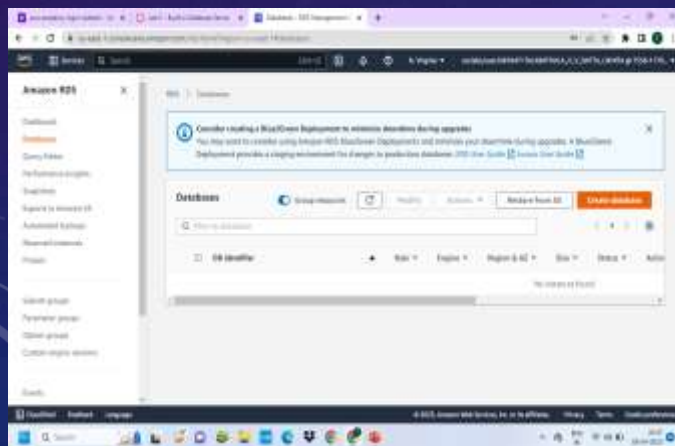


## Step 2 : Create a DB Subnet Group.

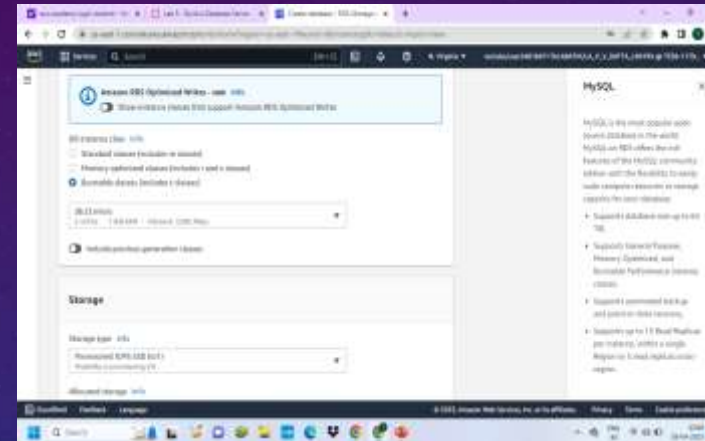
Rds → subnet groups → choose create DB subnet group → add subnets → create DB subnet group.



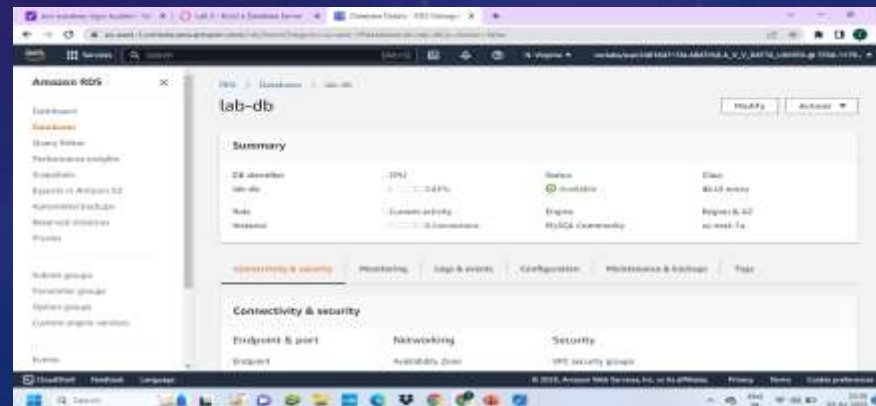
## Step 3: In the left navigation pane, choose **Databases** → choose create database → MYSQL



**Step 4:** In Availability and durability ,choose Multi –AZ DB instance then configure settings , DB instance class, Storage, connectivity, choose existing vpc security group and setup additional configuration.



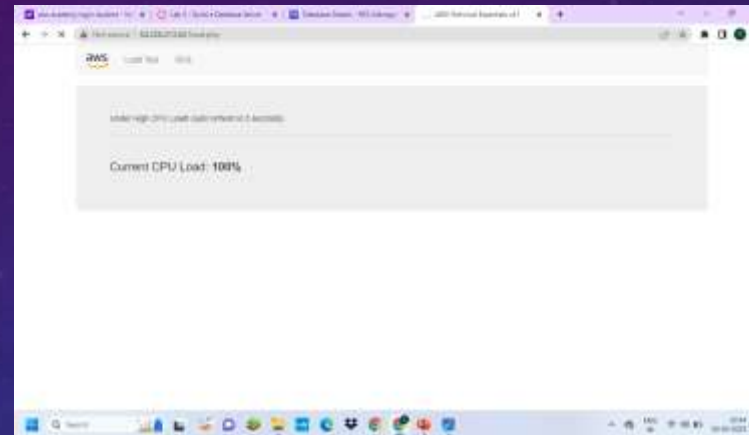
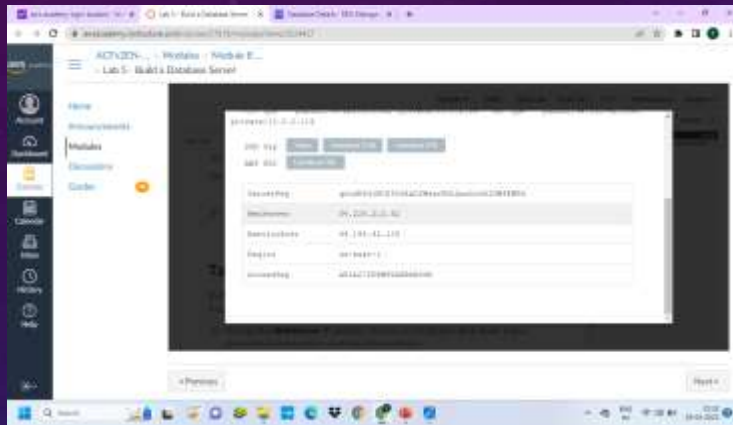
**Step 5:** Wait until Info changes to Modifying or Available.  
Scroll down to the Connectivity & security section and copy the **Endpoint** field.



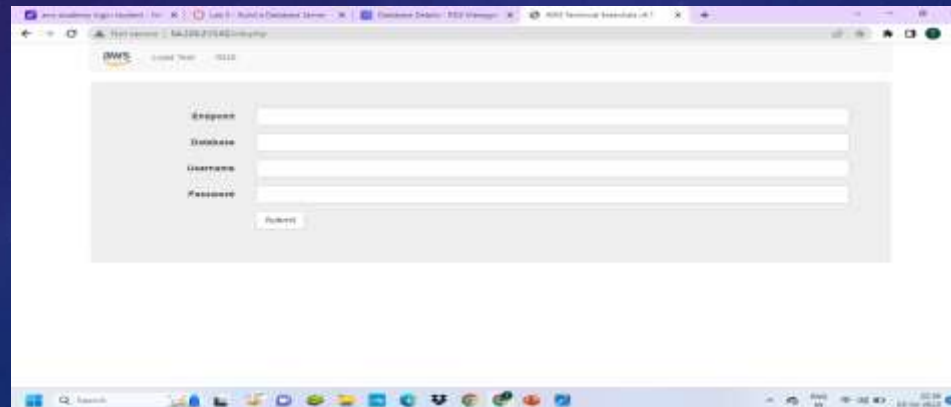
## Step 6 : Interact with Your Database.

On Details , copy the **WebServer** IP address. Open a new web browser tab, paste the WebServer IP address and press Enter.

The web application will be displayed, showing information about the EC2 instance.

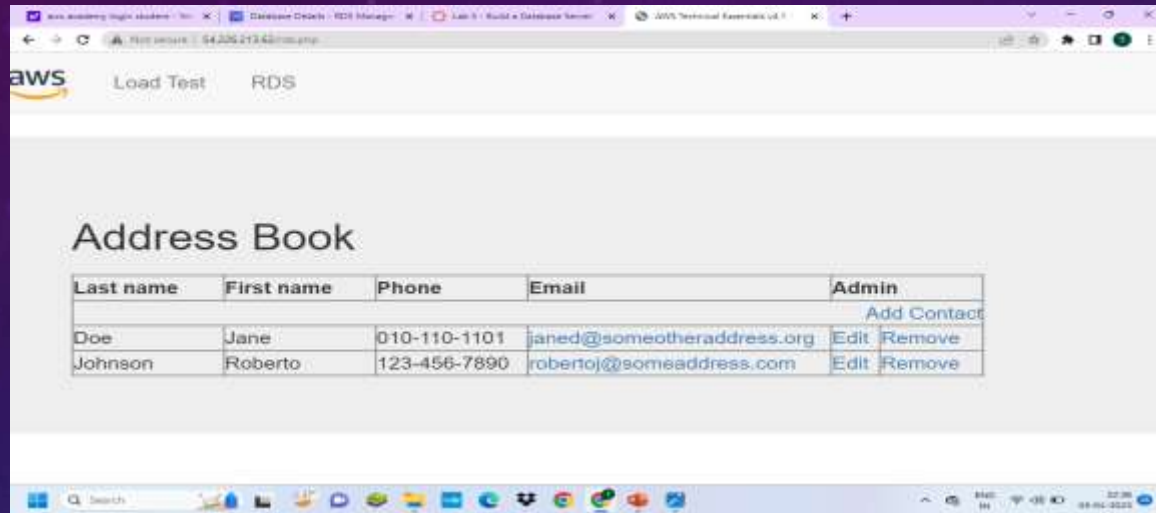


## Step 7 : Choose the **RDS** link at the top of the page and configure the settings.





**Step 8:** After a few seconds the application will display an **Address Book**.  
The Address Book application is using the RDS database to store information



# AWS S3 (SIMPLE STORAGE SERVICE)

## TASKS FOR CONFIGURING S3:

1. Log into the AWS Management Console.
2. Create an S3 bucket.
3. Upload an object to S3 Bucket.
4. Access the object on the browser.
5. Change S3 object permissions.
6. Setup the bucket policy and permission and test the object accessibility.

## STEPS :

**Step 1:** Click on **create group**.

**Step 2:** Set up the bucket name. S3 bucket name are globally unique, choose a name which is available. Leave other settings as default and click on **create group**.

**Step 3:** Click on your bucket name.

**Step 4:** Click Upload.

**Step 5:** Click on Add Files , and choose a file from your computer.

**Step 6:** After choosing your file, click on Next.

**Step 7:** Click on Upload.

**Step 8:**Now you have a private S3 bucket with a private object uploaded, which means you cannot visit it through Internet.

**Step 9:**Now you have a private S3 bucket with a private object uploaded, which means you cannot visit it through Internet.

## CHANGE BUCKET PERMISSIONS:

**Step 10:**Go back to your bucket and click on Permissions.

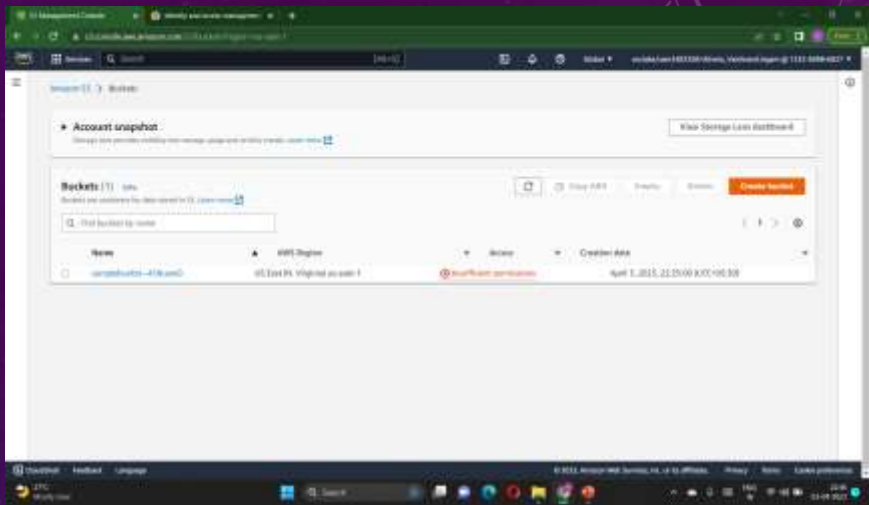
**Step 11:**Click on Everyone under the Public access, and click on Read object on the right of pop-up window. Then click on Save.

**Step 12 :**Now its state switches to Read Object - Yes

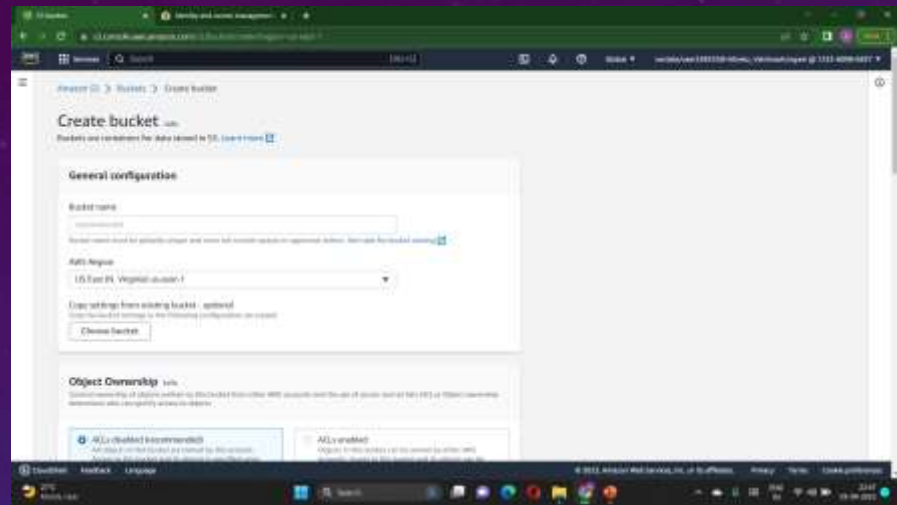
**Step 13:**Click on Overview, and click on your Object URL again .

**Step 14:**Notice the URL on your browser

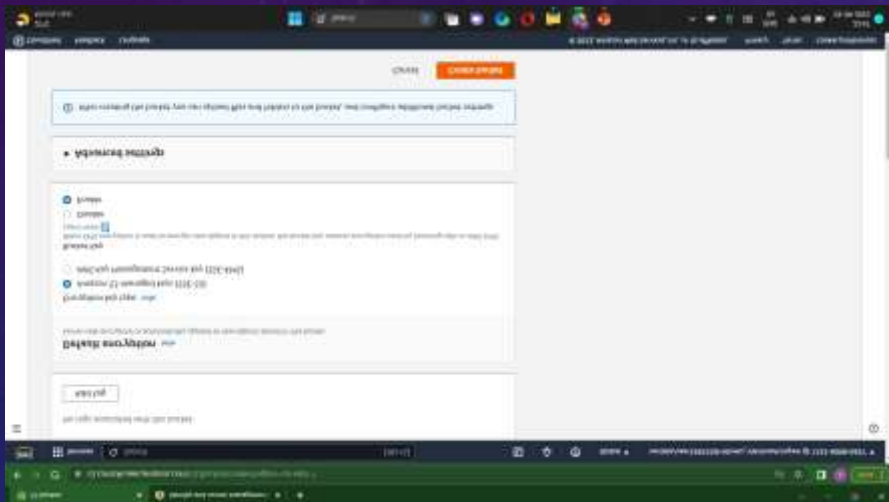




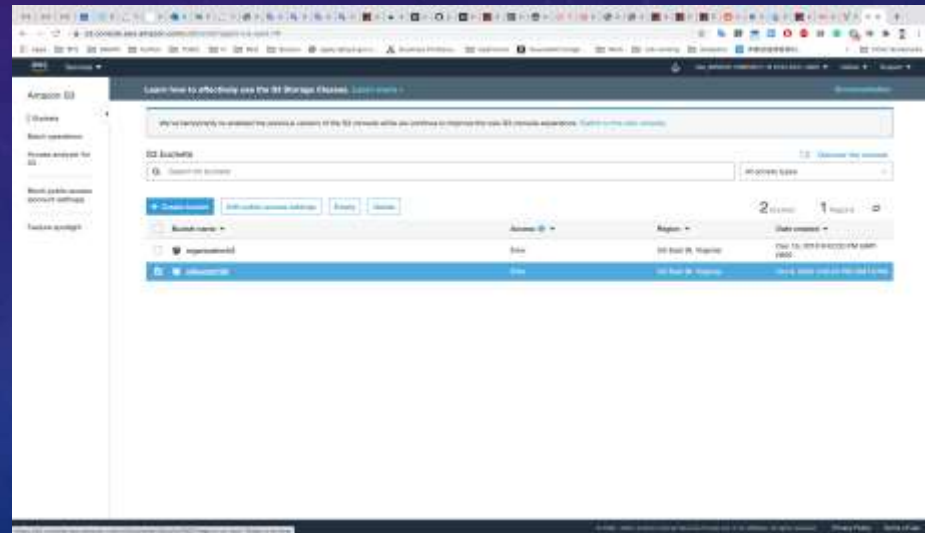
Step 1



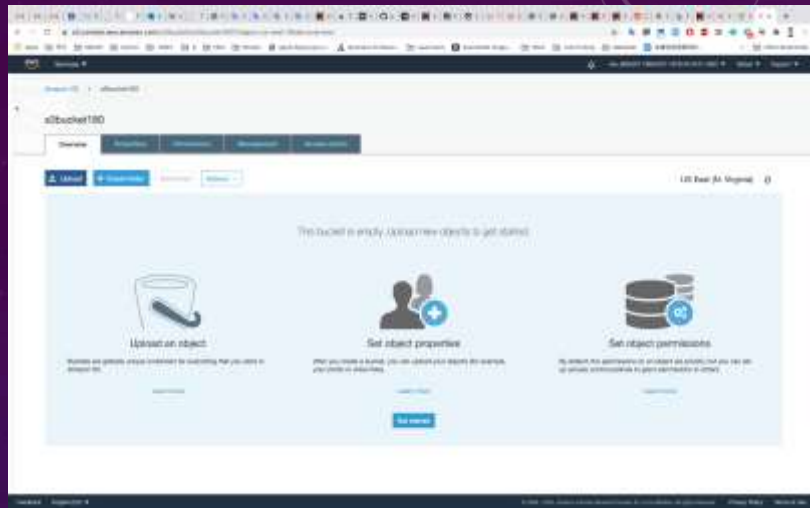
Step 2



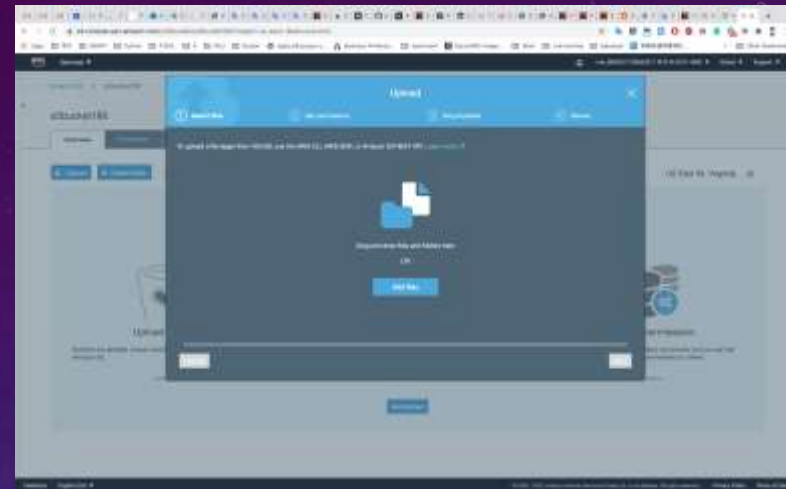
Step 2



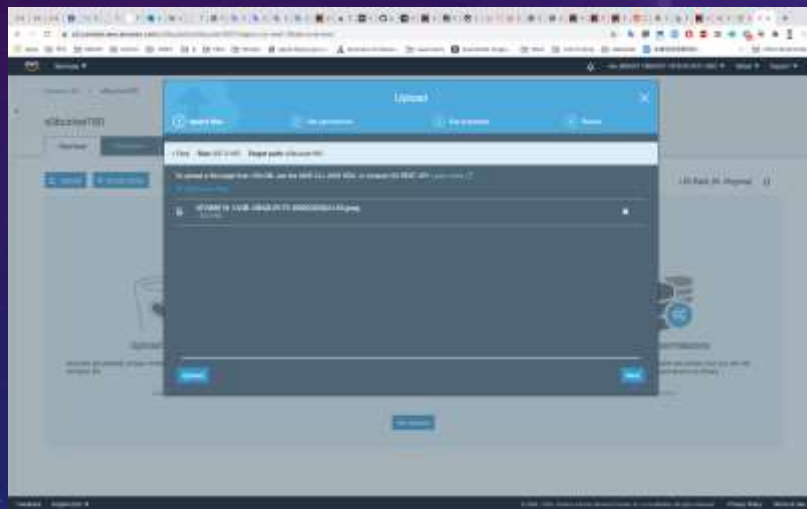
Step 3



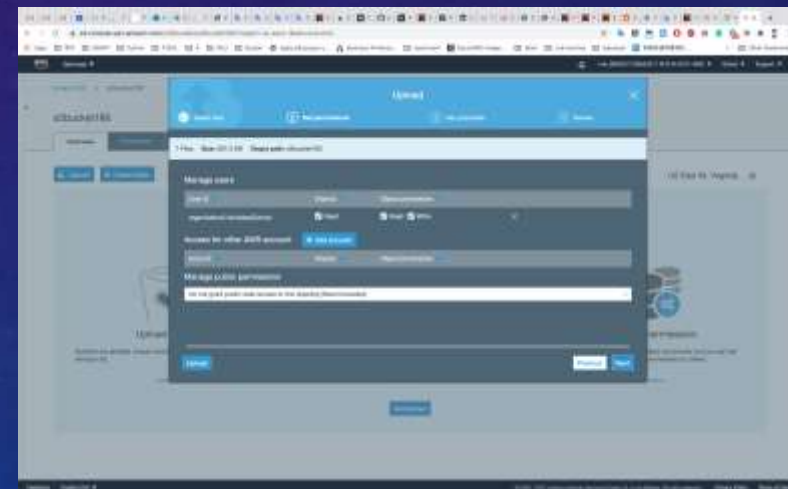
Step 4



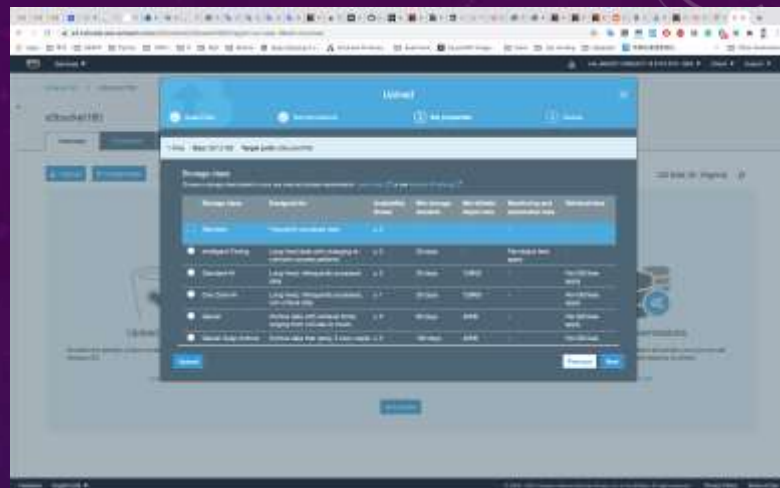
Step 5



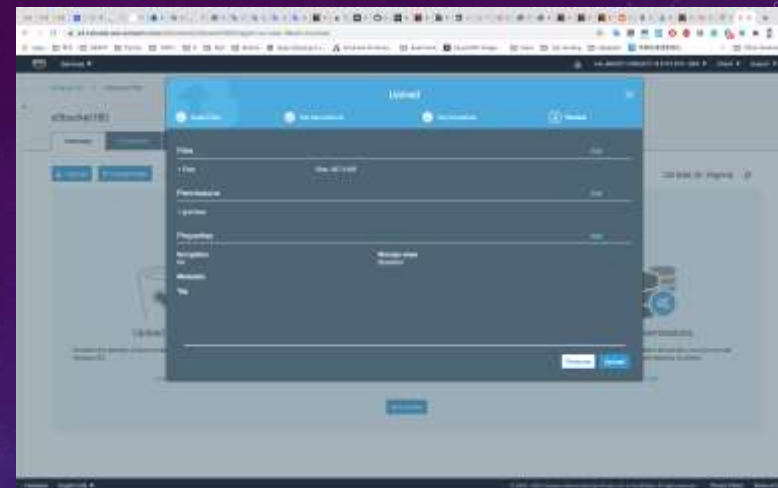
Step 6



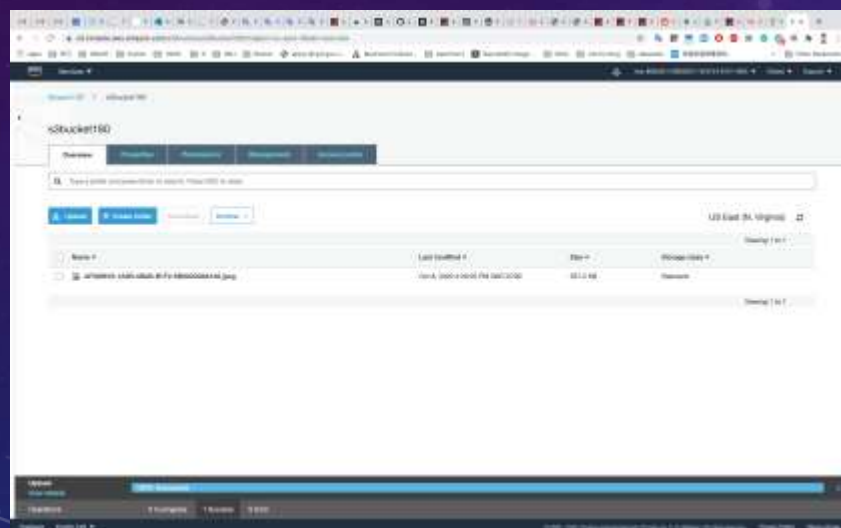
Step 7



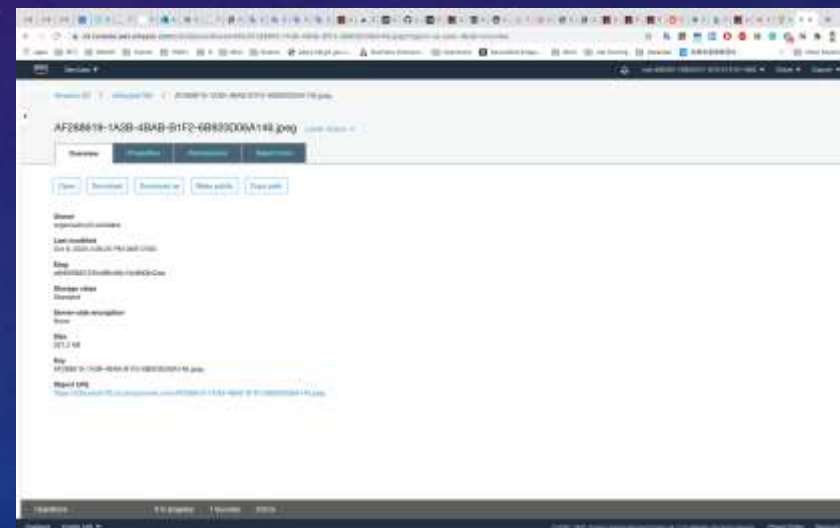
Step 8



Step 9

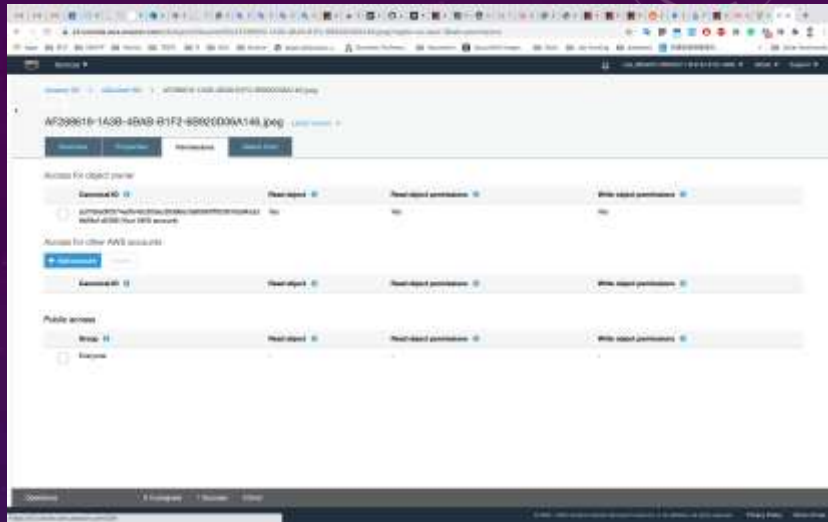


Step 10

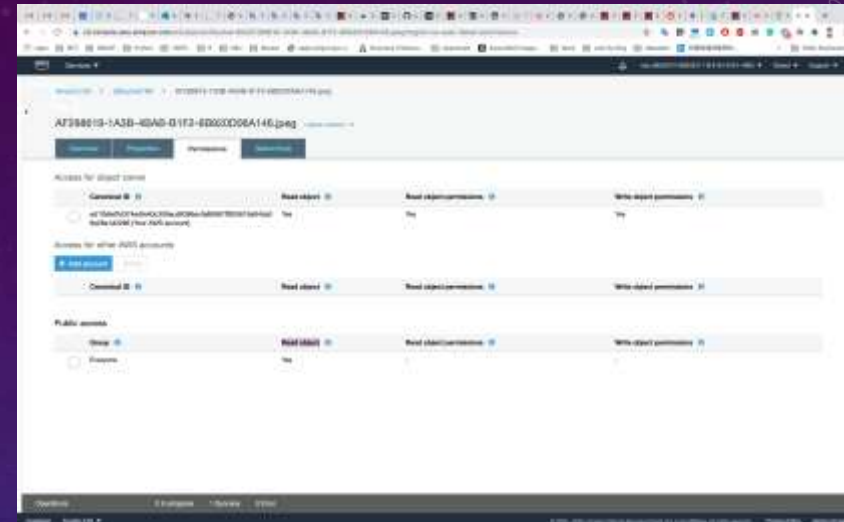


Step 11





Step 12



Step 13



Step 14

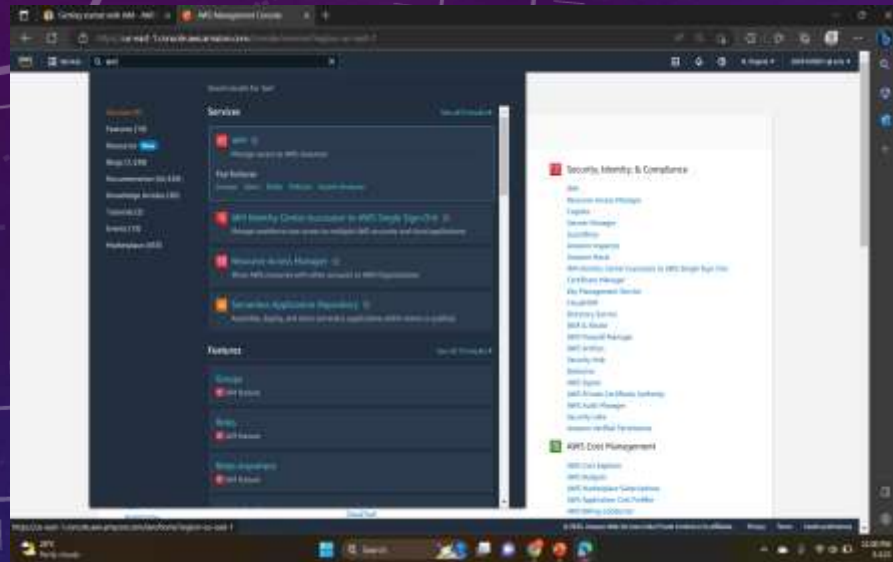
# AWS Identity and Access Management (IAM)

**AWS Identity and Access Management (IAM)** is a web service that enables Amazon Web Services (AWS) customers to manage users and user permissions in AWS. With IAM, you can centrally manage **users**, **security credentials** such as access keys, and **permissions** that control which AWS resources users can access.

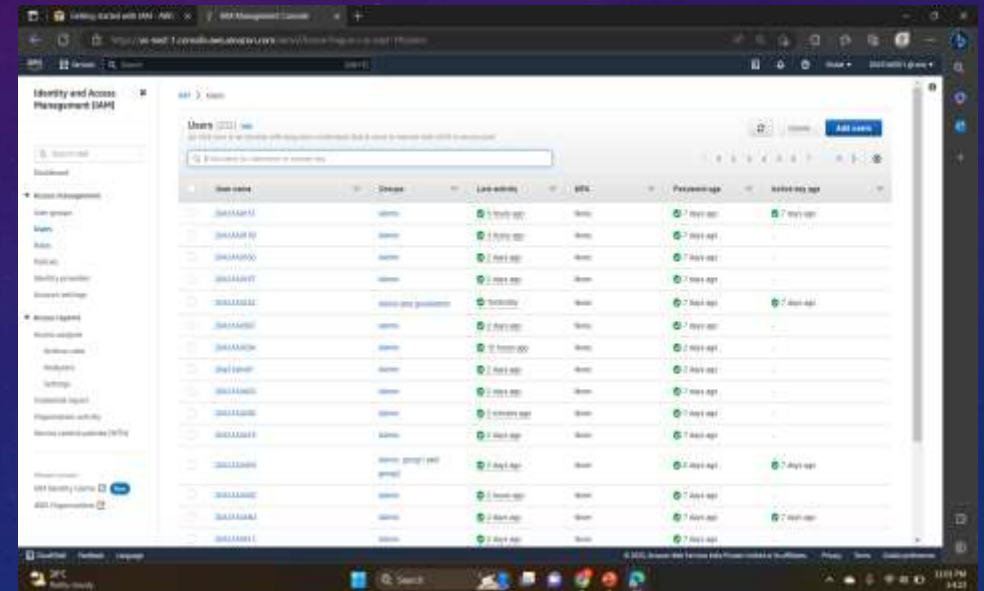


# Steps to create IAM User and User Groups

1. On the **Console Home** page, select the IAM service.



2. In the navigation pane, select **Users** and then select **Add users**.

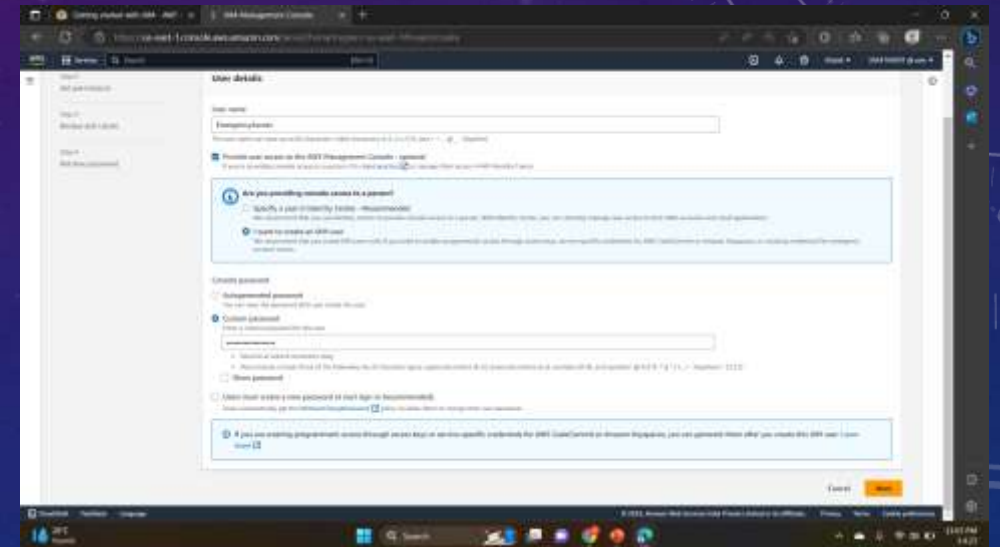
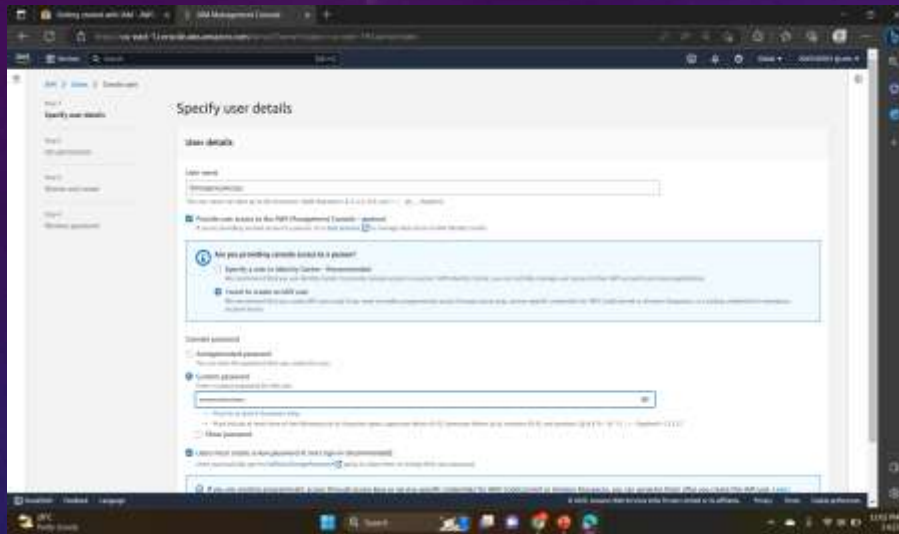




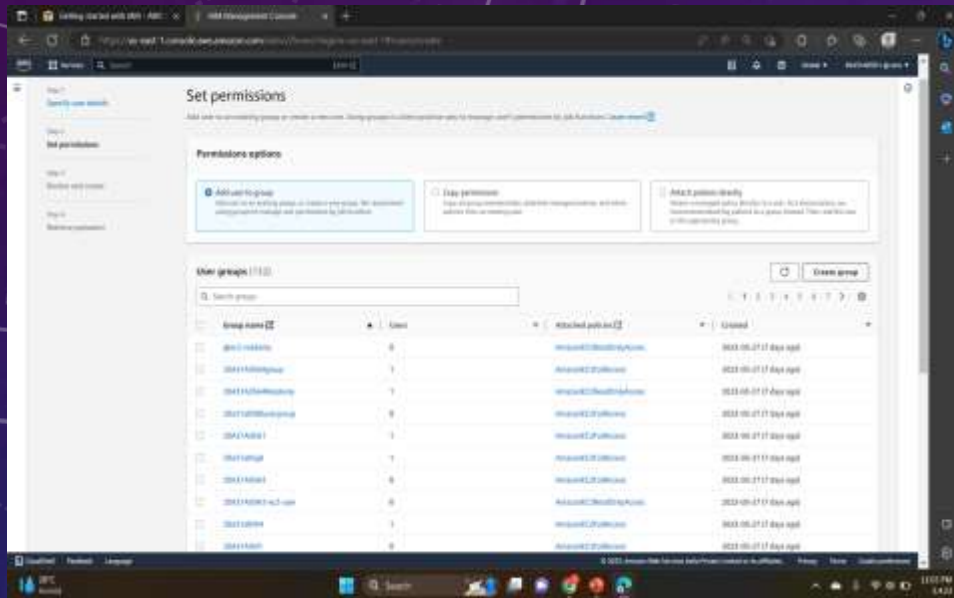
3. For Username, enter EmergencyAccess and ,Select the check box next to **Provide user access to the AWS Management Console— optional** and then choose **I want to create an IAM user**.

4. Under **Console password**, select **Custom Password** and create your own password.

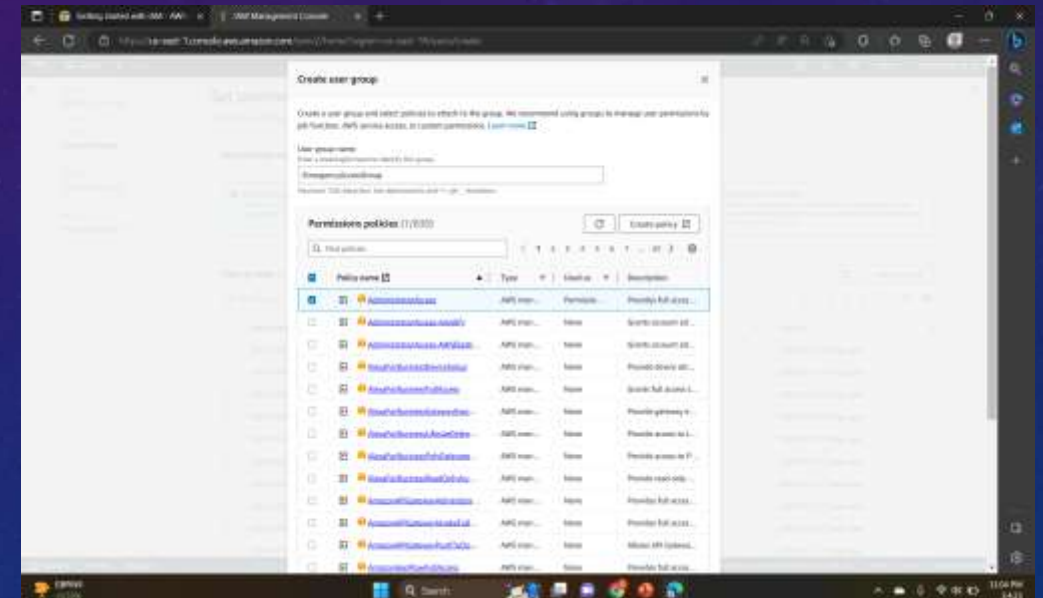
5. Clear the check box next to **User must create a new password at next sign-in (recommended)**. Then click on **Next**.



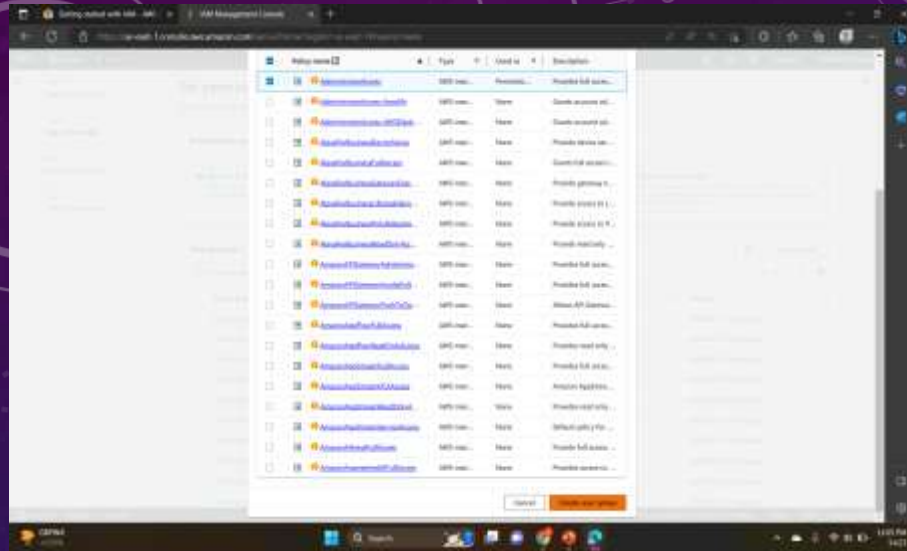
6. On the **Set permissions** page, under **Permissions options**, select **Add user to group**. Then, under **User groups**, select **Create group**.



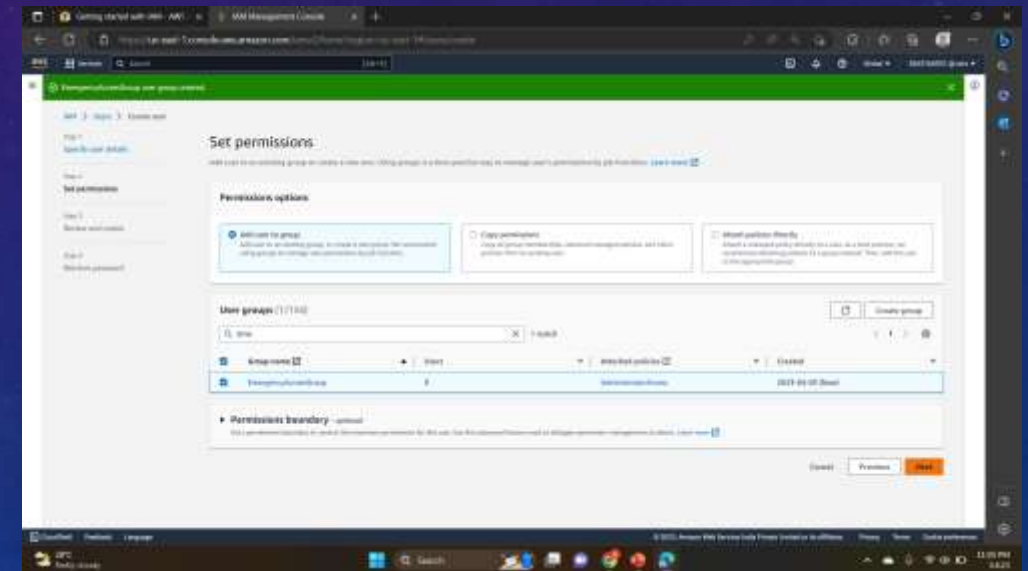
7. On the **Create user group** page, in **User group name**, enter **EmergencyAccessGroup**. Then, under **Permissions policies**, select **AdministratorAccess**.



8. Select **Create user group** to return to the **Set permissions** page.



9. Select **Next** to proceed to the **Review and create** page.

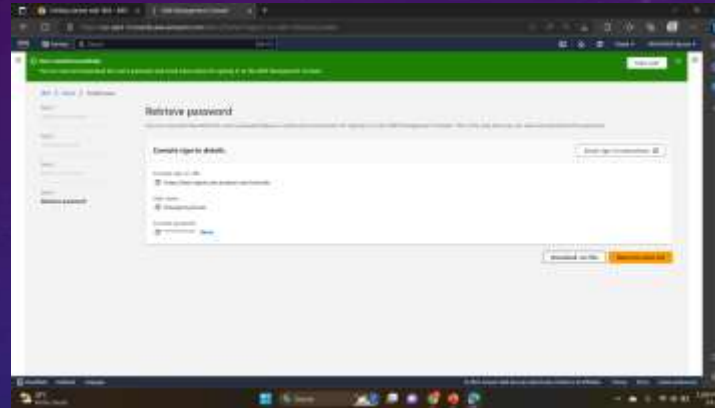




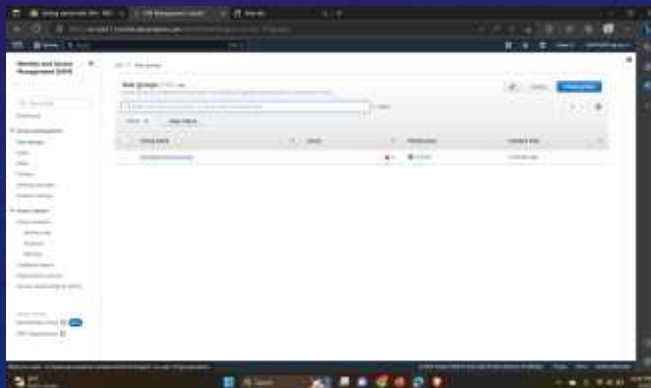
10. On the **Review and create** page, review the list of user group memberships to be added to the new user. When you are ready to proceed, select **Create user**.

11. On the **Retrieve password** page, select **Download .csv file** to save a .csv file with the user credential information (Connection URL, user name, and password).

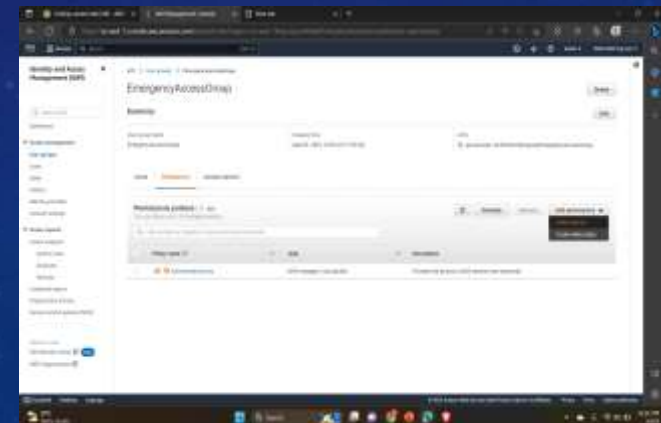
12. Save this file to use if you need to sign-in to IAM and do not have access to your federated identity provider.



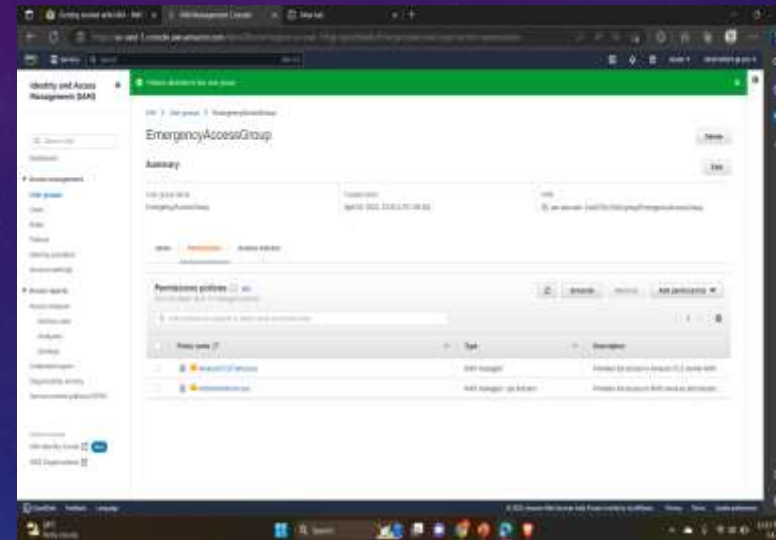
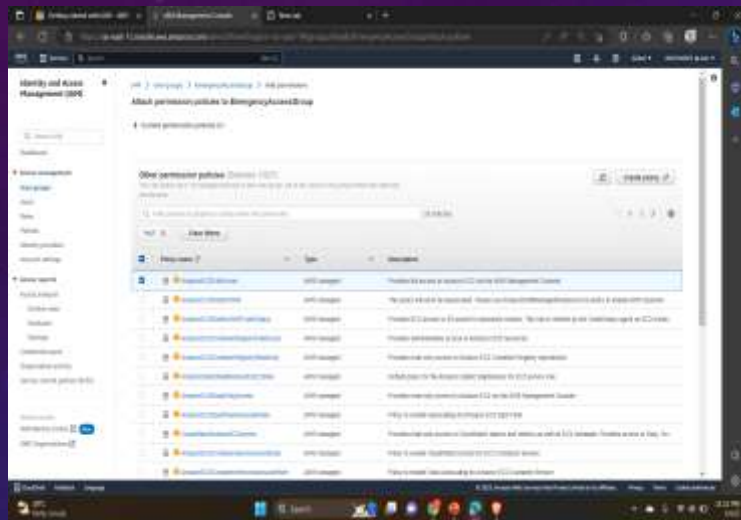
13. If you want to add permissions to the user group, then go to **User groups** and click on the respective **User group**.



14. Go to **Permissions** → **All permissions** → **Attach policies**



## 15. Add the permission policy and the policy is attached to the User group.



# AWS LAMBDA

1) In the search box to the right of Services, search for and choose Lambda to open the AWS Lambda console.

2) Choose Create function.

3) In the Create function screen, configure these settings:

> Choose Author from scratch

> Function name: myStopinator

> Runtime: Python 3.8

> Choose Change default execution role

> Execution role: Use an existing role

> Existing role: From the dropdown list, choose myStopinatorRole

4) Choose Create function.

5) Choose Add trigger.

6) Choose the Select a trigger dropdown menu, and select EventBridge (CloudWatch Events).

7) For the rule, choose Create a new rule and configure these settings:

Rule name: everyMinute

Rule type: Schedule expression

Schedule expression: rate(1 minute)



Below the Function overview pane, choose Code, and then choose `lambda_function.py` to display and edit the Lambda function code.

In the Code source pane, delete the existing code. Copy the following code, and paste it in the box:

```
import boto3
region = '<REPLACE_WITH_REGION>'
instances = ['<REPLACE_WITH_INSTANCE_ID>']
ec2 = boto3.client('ec2', region_name=region)
def lambda_handler(event, context):
    ec2.stop_instances(InstanceIds=instances)
    print('stopped your instances: ' + str(instances))
```

9) Replace the `<REPLACE_WITH_REGION>` placeholder with the actual Region that you are using. To do this:

10) Choose on the region on the top right corner and use the region code. For example, the region code for US East (N. Virginia) is `us-east-1`.

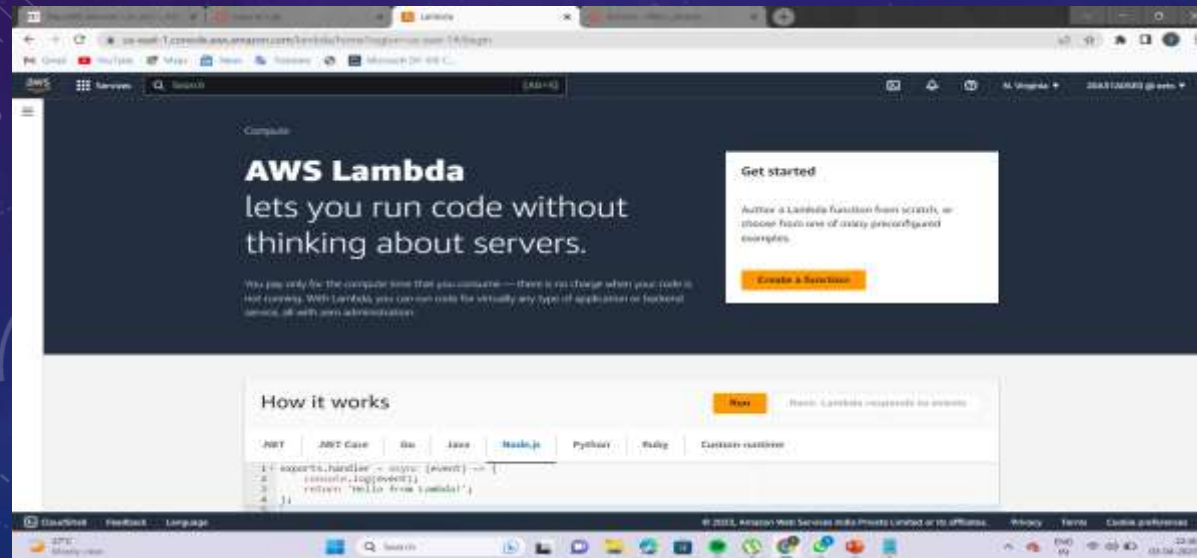
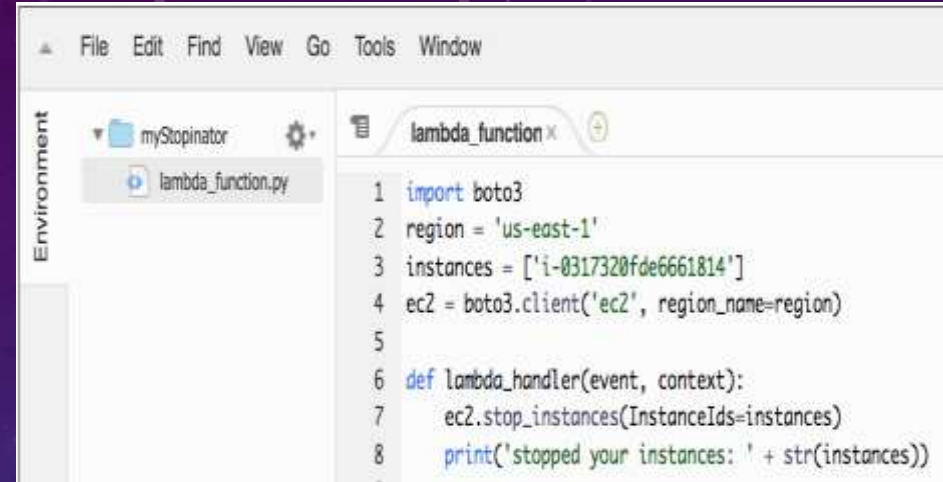
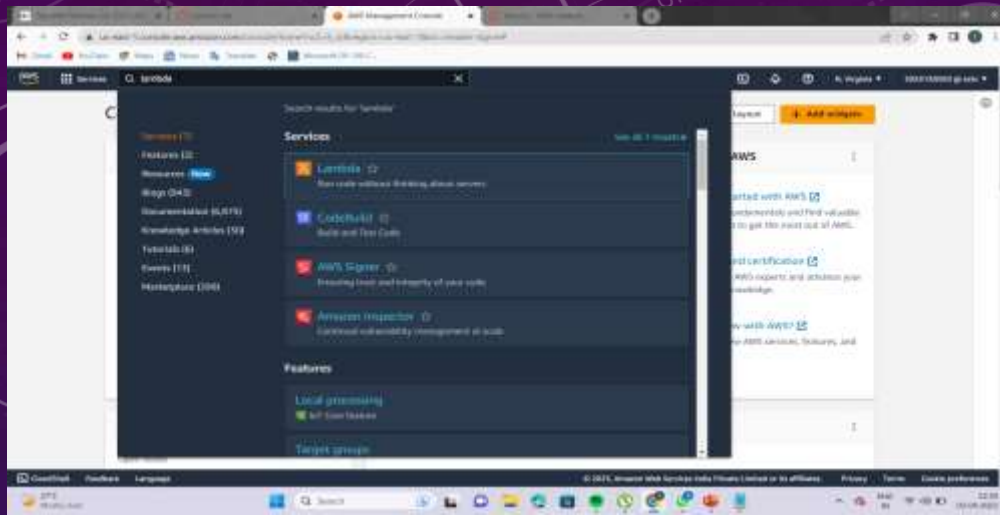
11) Verify that an EC2 instance named `instance1` is running in your account, and copy the `instance1` instance ID.

12) Return to the AWS Lambda console browser tab, and replace `<REPLACE_WITH_INSTANCE_ID>` with the actual instance ID that you just copied.

13) Choose the File menu and Save the changes. Then, in the top-right corner of the Code source box, choose Deploy.

14) Choose Monitor

15) Return to the Amazon EC2 console browser tab and see if your instance was stopped.



The background is a gradient of dark blue to purple, overlaid with a pattern of small white stars. Several faint, light-blue technical diagrams are visible. In the top left, there is a small circular diagram with a curved arrow. In the top right, a large circular diagram features concentric circles, radial lines, and degree markings from 0 to 210. In the bottom right, another circular diagram shows concentric circles with dashed lines and arrows. In the bottom left, a partial circular diagram with a dashed arrow is visible.

THANK YOU