

PRACTICAL IMPLEMENTATION ON AWS



A.AMULYA SAI
20A31A05D0

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

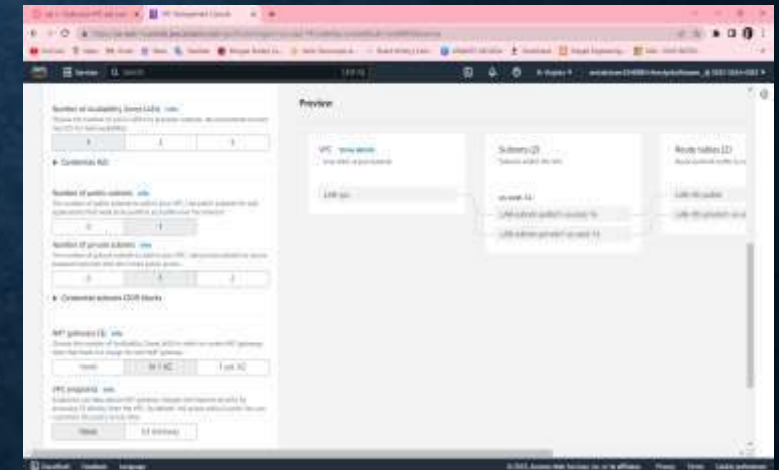
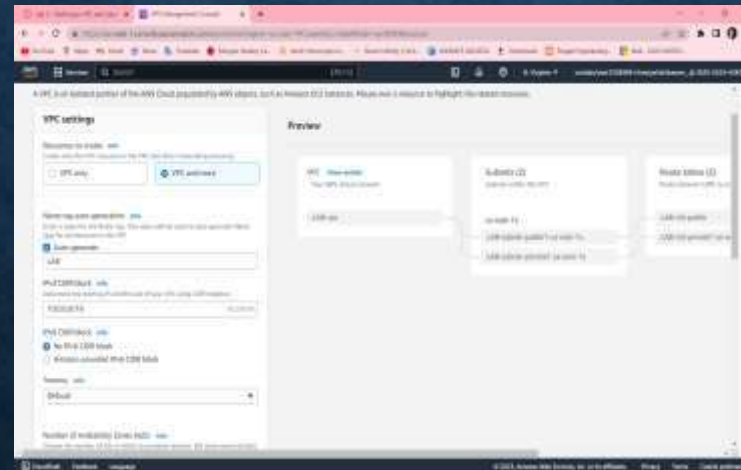
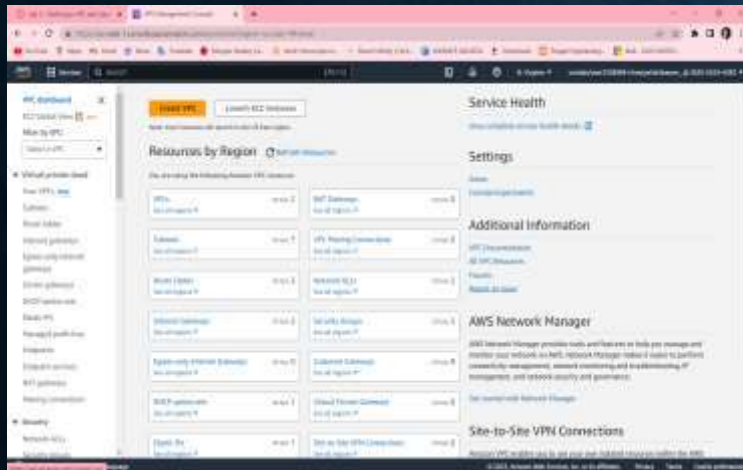
Step 4: Choose VPC and more, in name tag auto-generation, keep auto-generate select and change it to a lab.

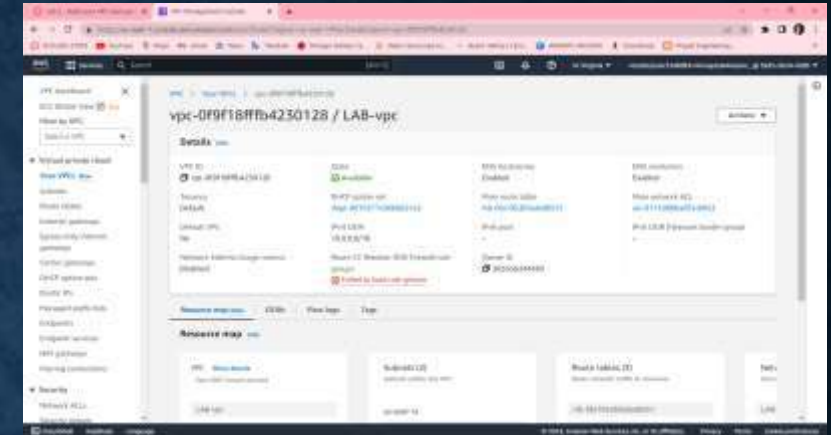
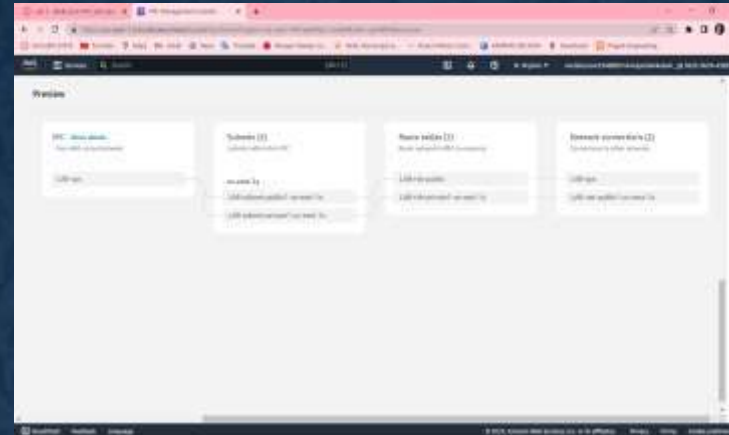
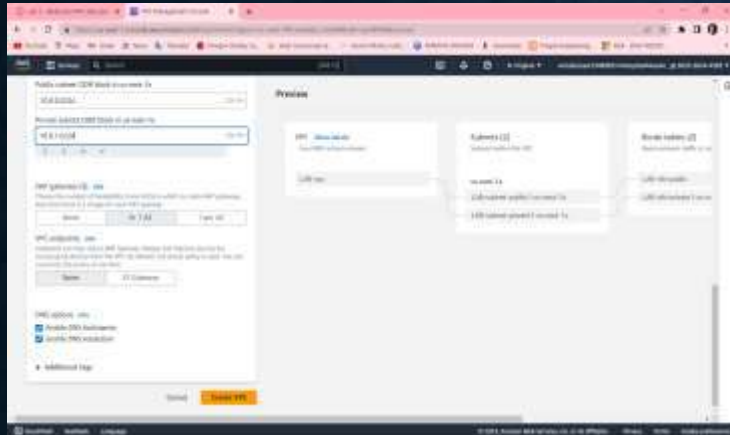
Step 5: Keep the IPv4 CIDR block to 10.10.0.0/16 and next for a number of availability zones select 1, number of public subnets select 1, number of private subnets select 1

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

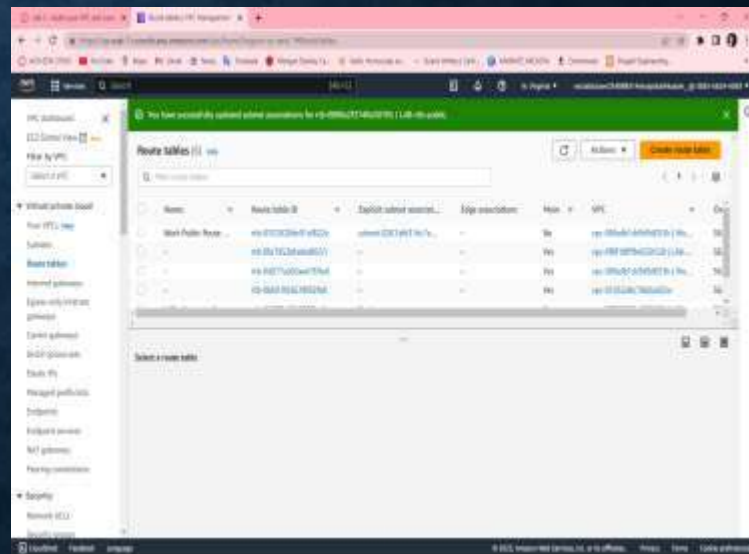
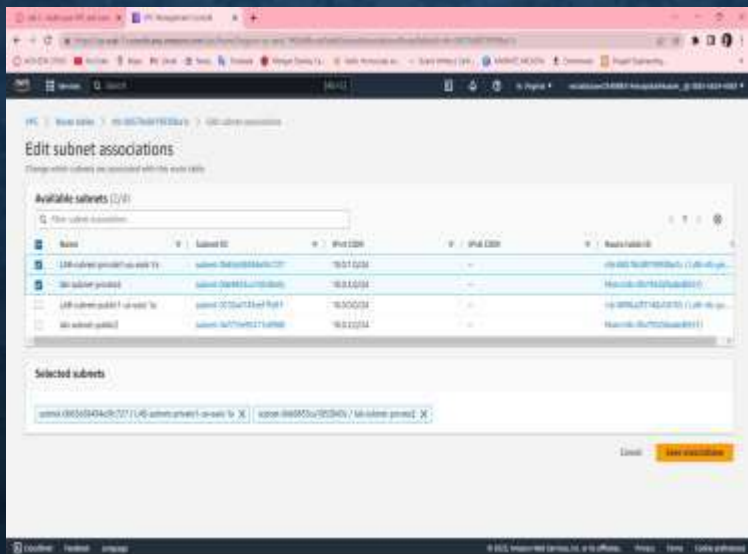
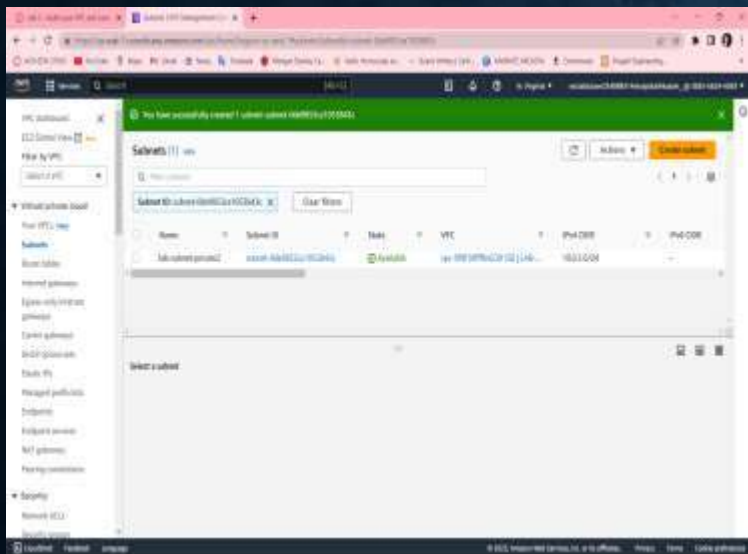
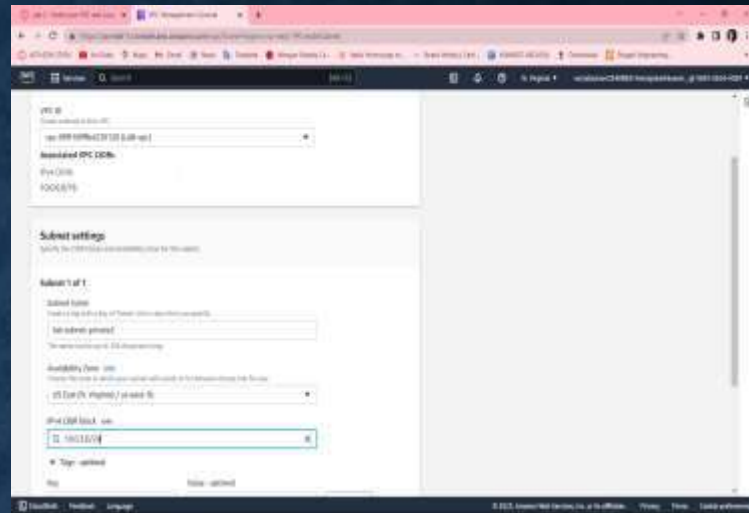
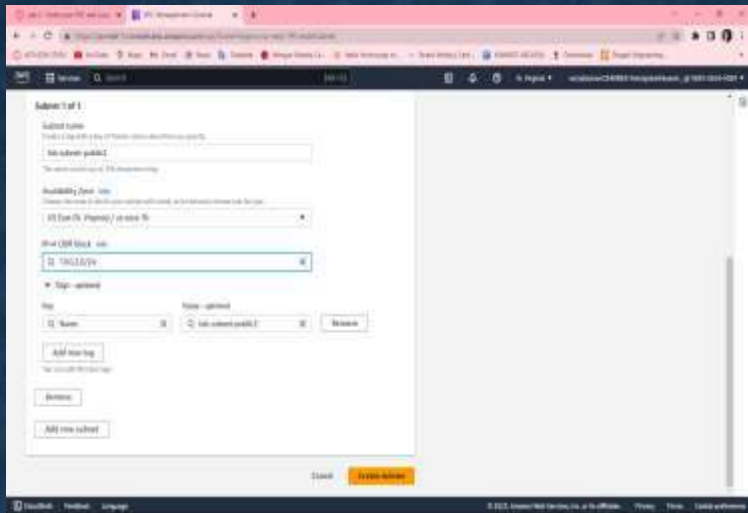
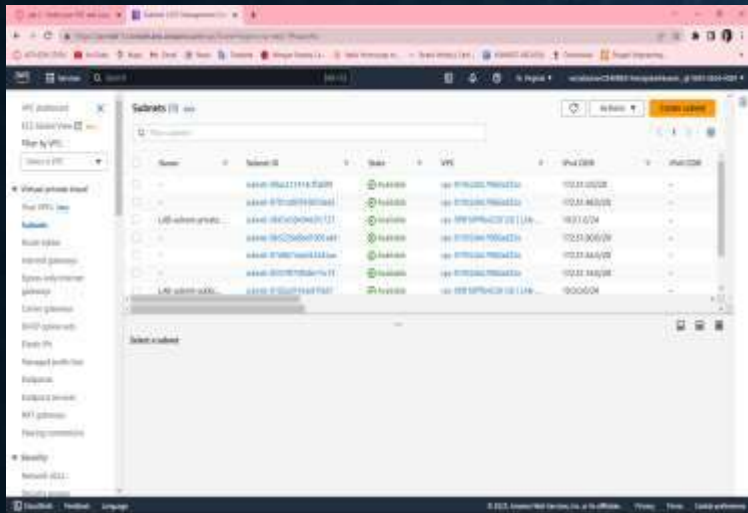
Step 5 : Leave lab-subnet-private1-us-east-1a and also select lab-subnet-private2

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

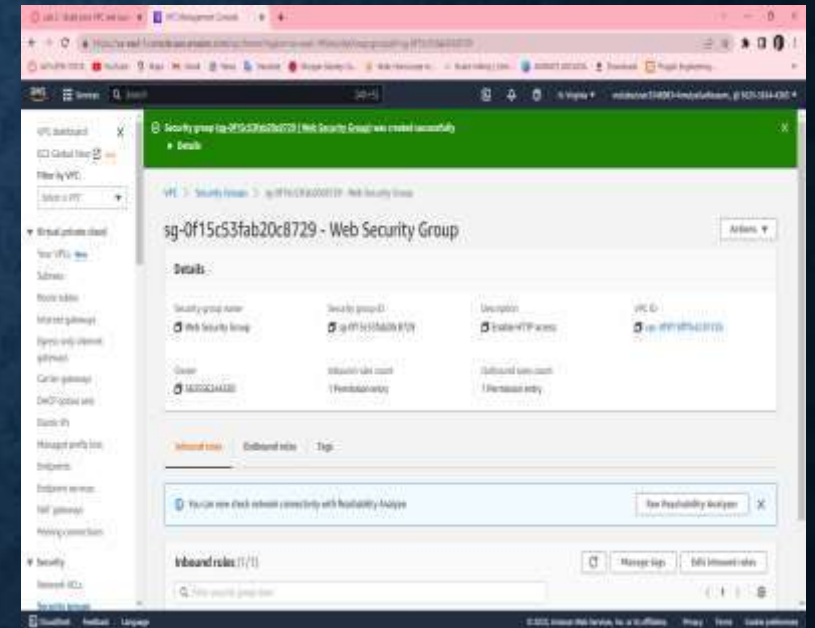
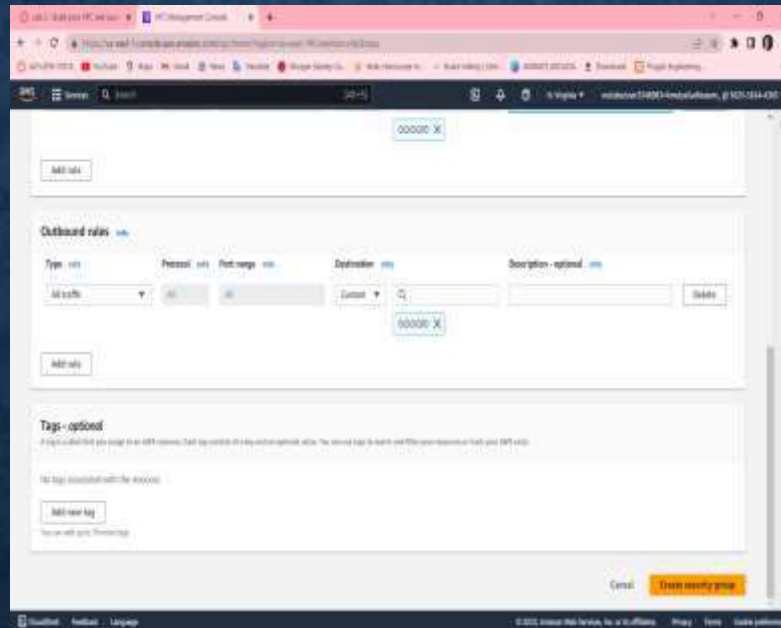
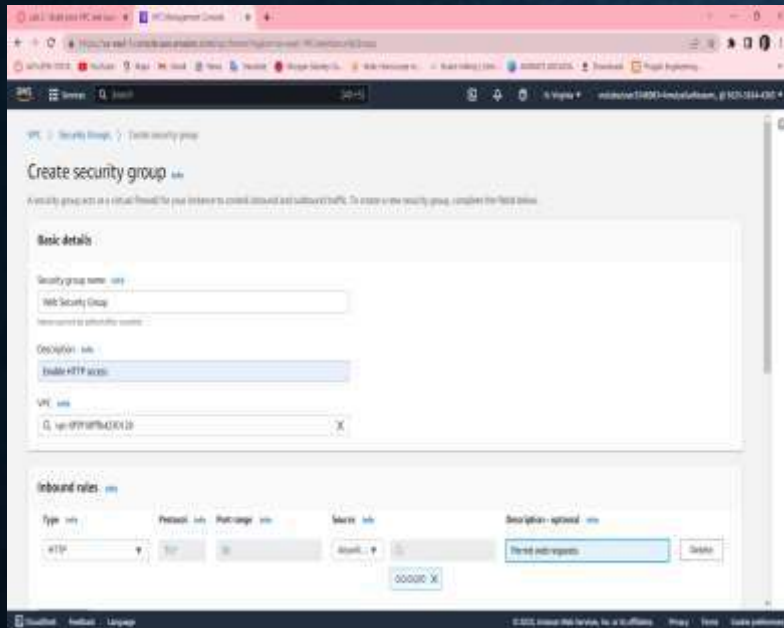
Step 9: Choose SAVE ASSOCIATIONS



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 auto 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
```

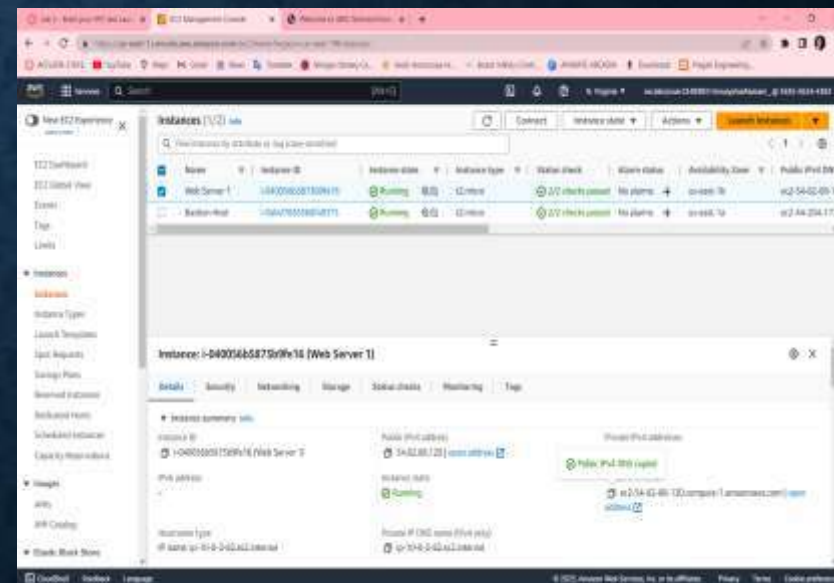
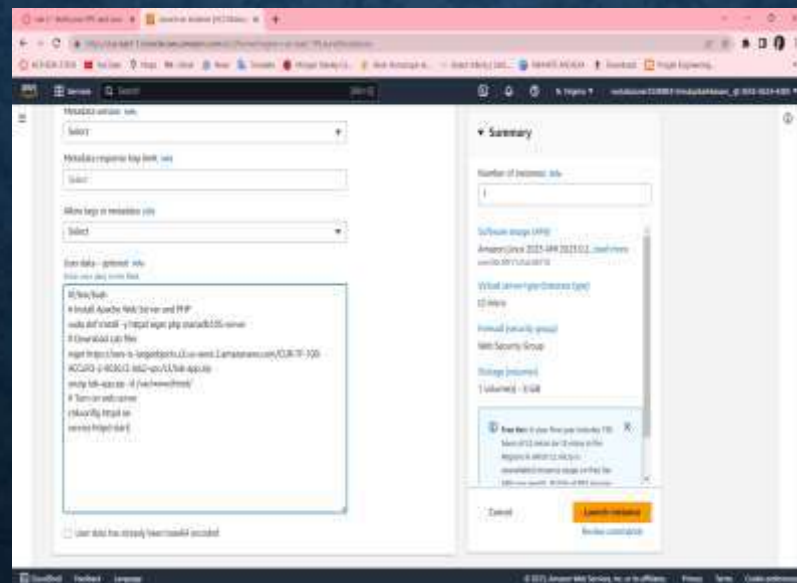
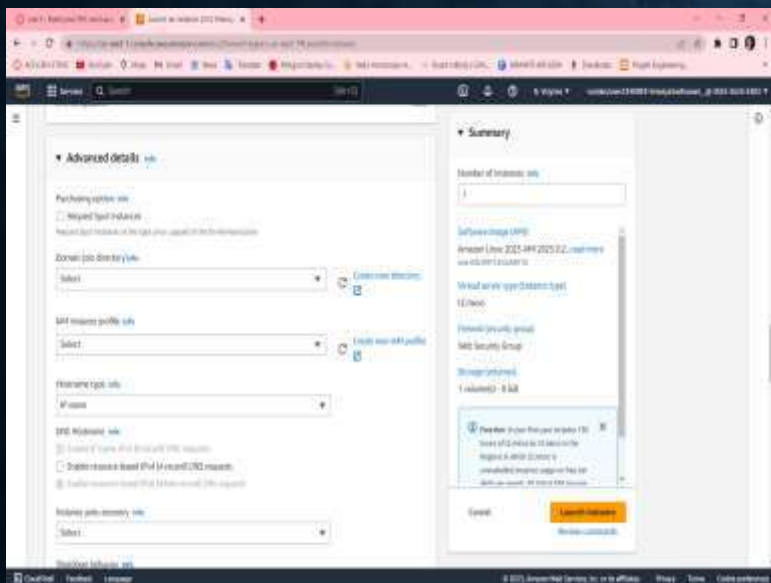
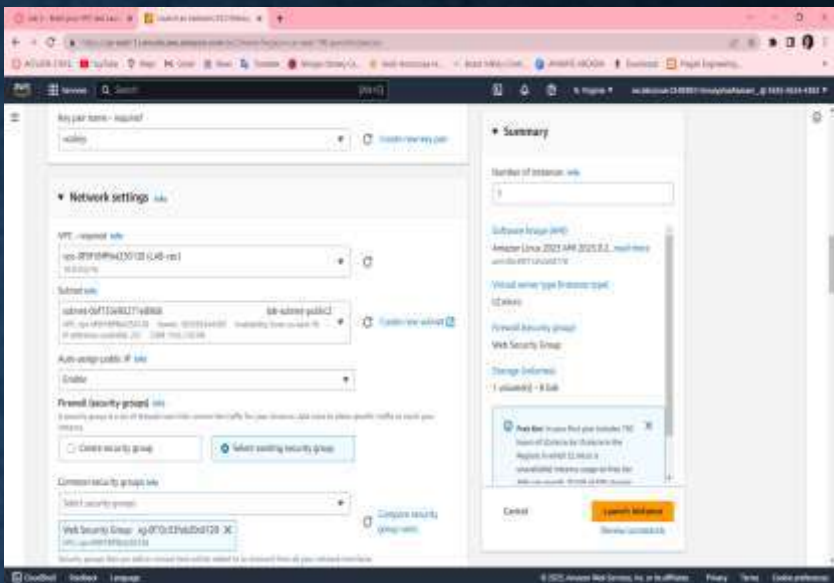
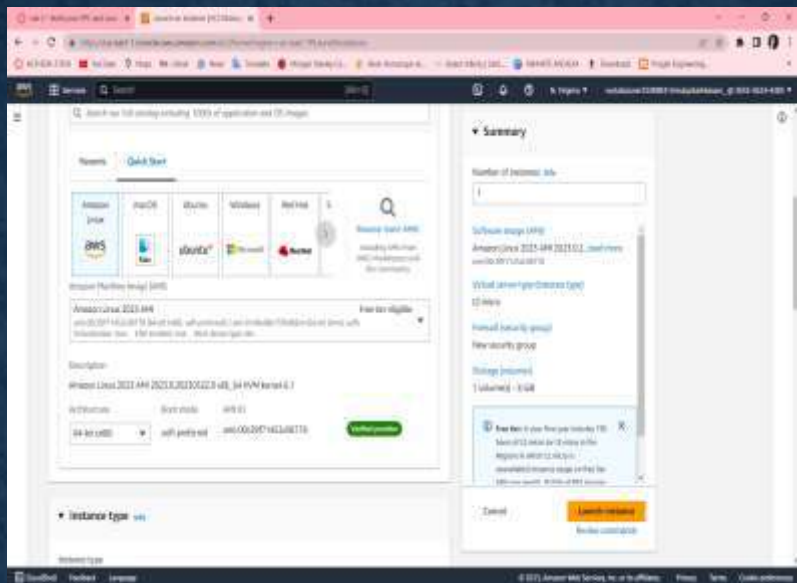
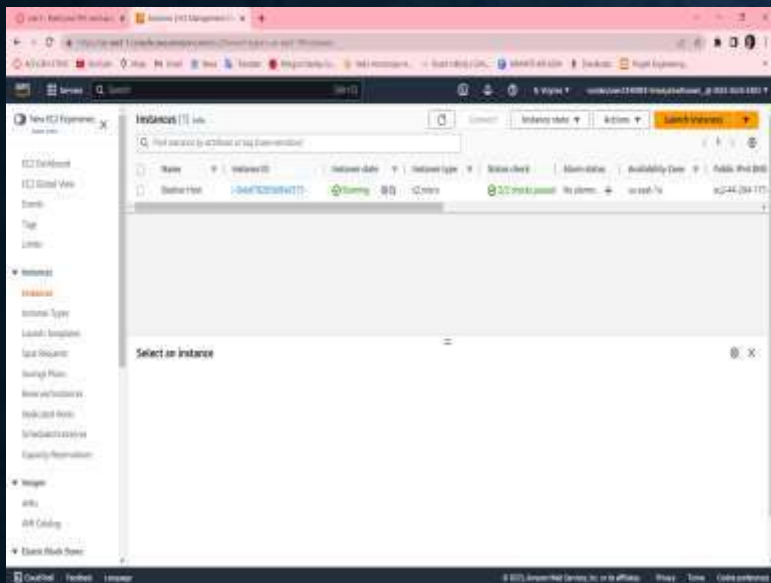
Step 6: At the bottom SUMMARY panel choose launch instance ,click on view instances

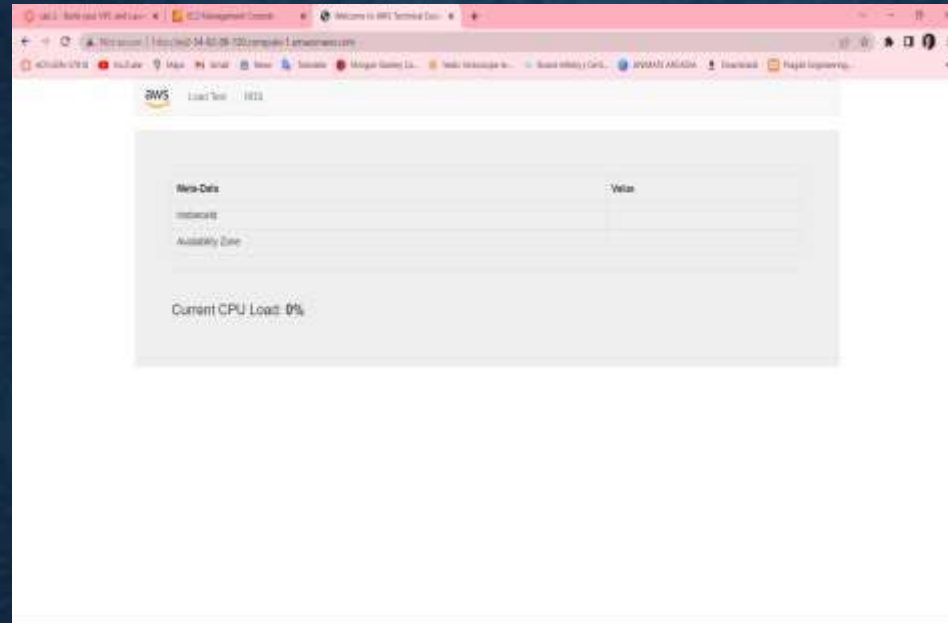
Step 7 : Wait until web server 1 shows 2/2 checks passed

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

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.

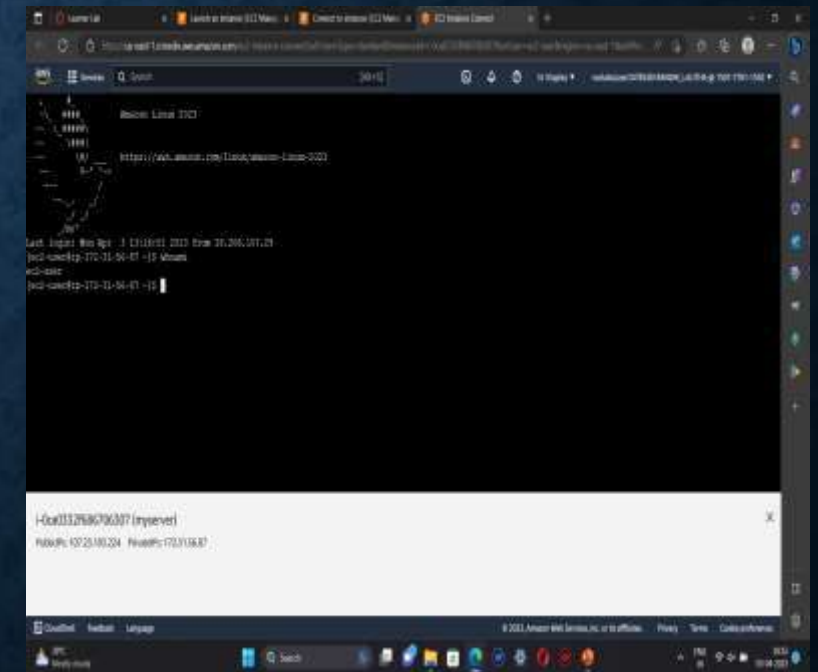
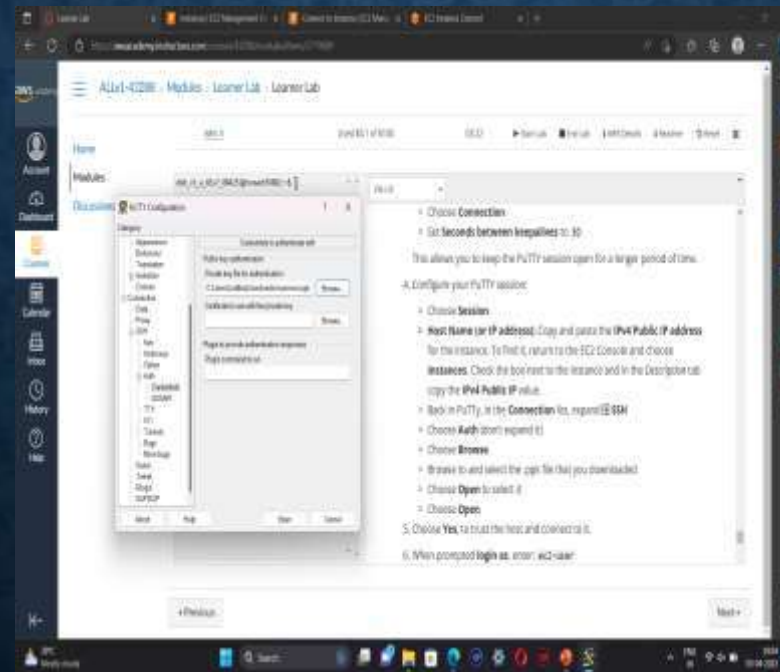
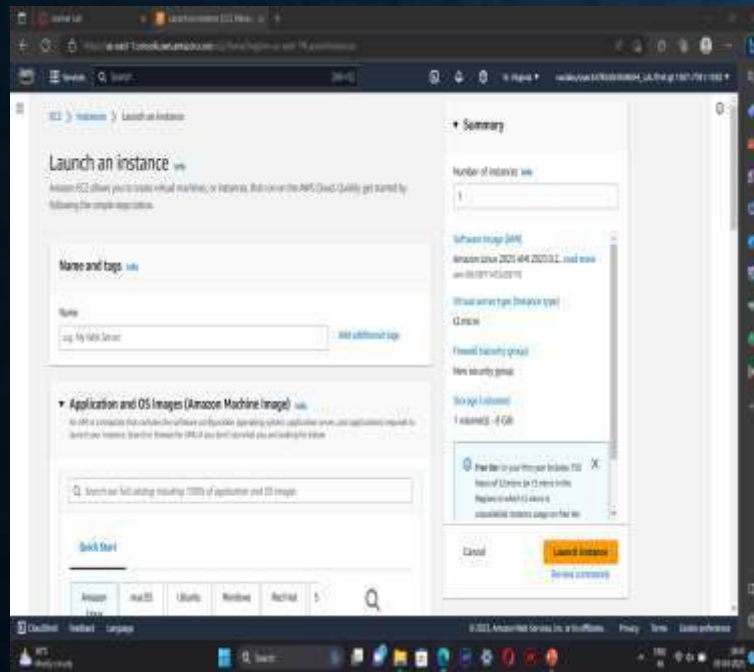
Step-3: For linux-select ppk key and for windows server-select pem key.

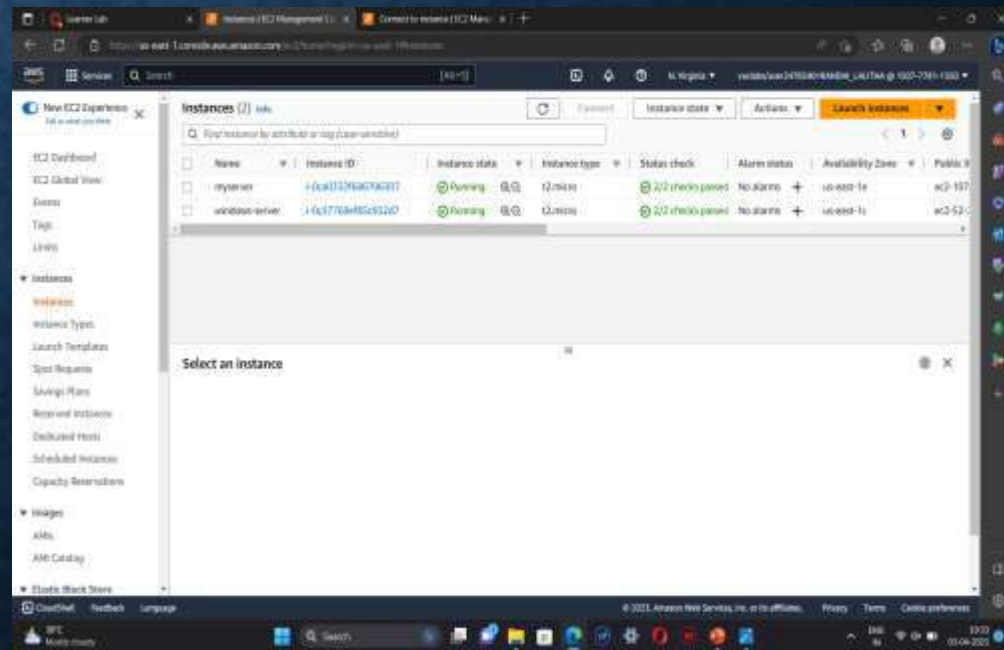
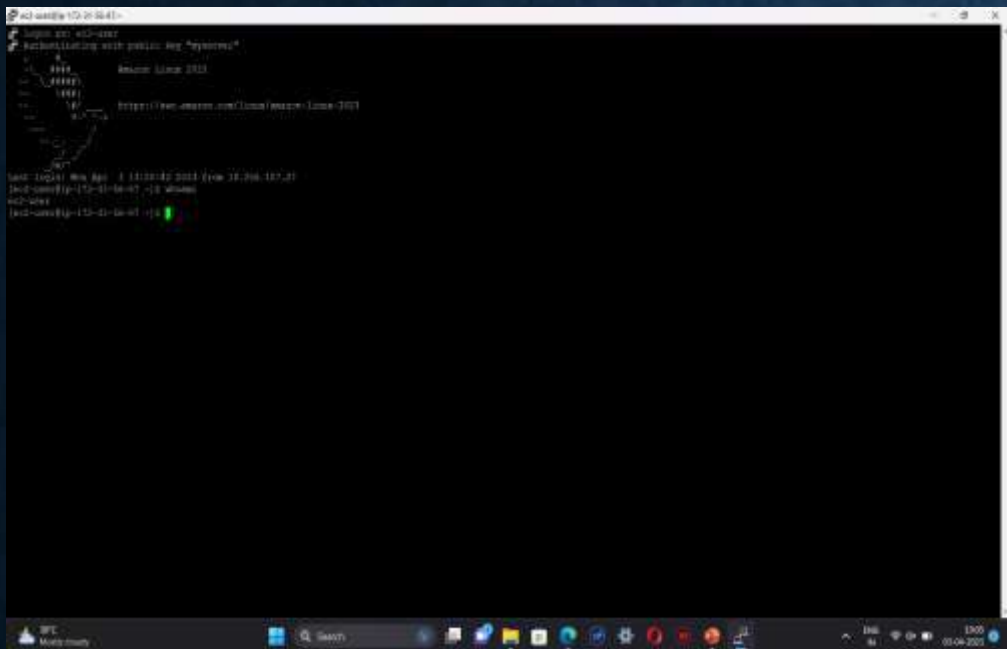
Step-4: If a key pair is not available create a new key.

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 .





AMAZON LIGHT SAIL

PROCEDURE:

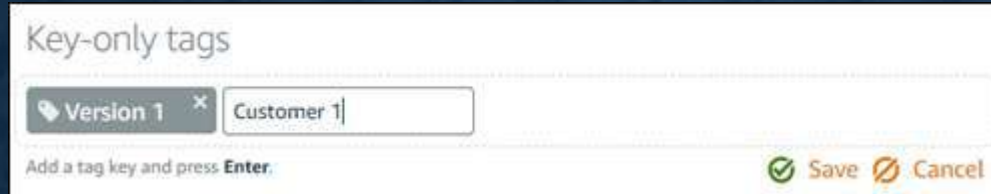
1. On the home page, choose Create instance.
2. Select a location for your instance (an AWS Region and Availability Zone). Choose Change Region and zone to create your instance in another location.
3. Optionally, you can change the Availability Zone. Choose an Availability Zone from the dropdown list.
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.

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.



A dialog box titled "Key-only tags". It features a tab labeled "Version 1" with a close button (X). Below the tab is a text input field containing the text "Customer 1". At the bottom left, there is a prompt: "Add a tag key and press Enter." At the bottom right, there are two buttons: "Save" with a green checkmark icon and "Cancel" with a red X icon.

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.



A dialog box titled "Key-value tags". It contains two text input fields. The first field is labeled "Key" and contains the text "Project". The second field is labeled "Value" and contains the text "Earth". An arrow points from the "Key" field to the "Value" field. To the right of the input fields are two buttons: "Cancel" with a red X icon and "Save" with a green checkmark icon and a mouse cursor pointing at it.

Choose Create instance.

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

How to connect to your instance

1. From the Lightsail home page, choose the menu on the right of your instance's name, and then choose connect.



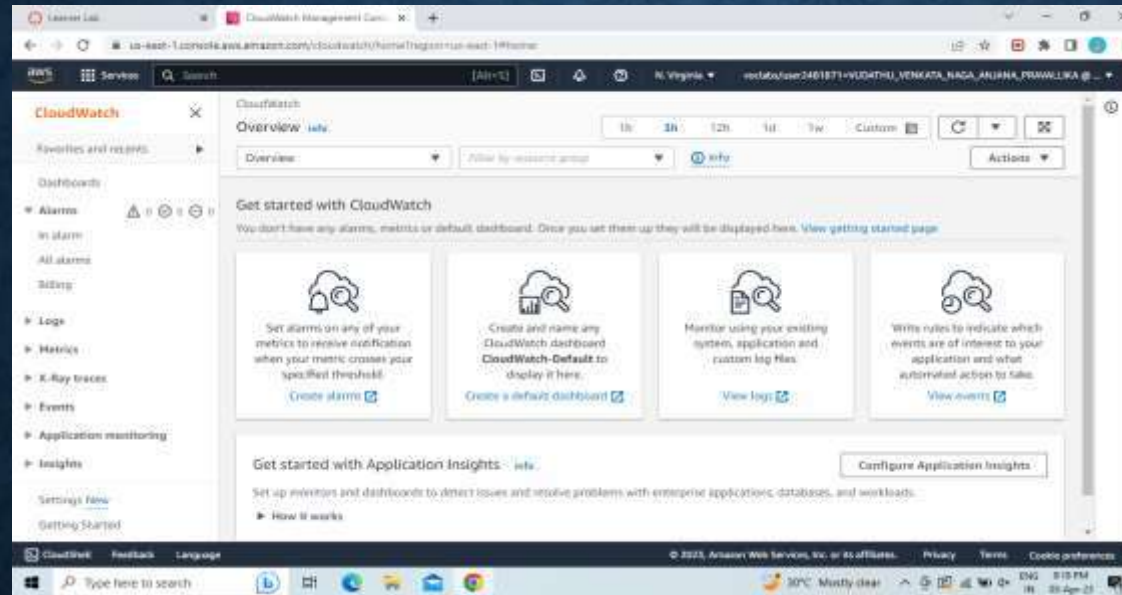
Alternately, you can open your instance management page and choose the Connect tab.
.You can now type commands into the terminal and manage your Lightsail instance without setting up an SSH client.



AWS CLOUD WATCH

PROCEDURE

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



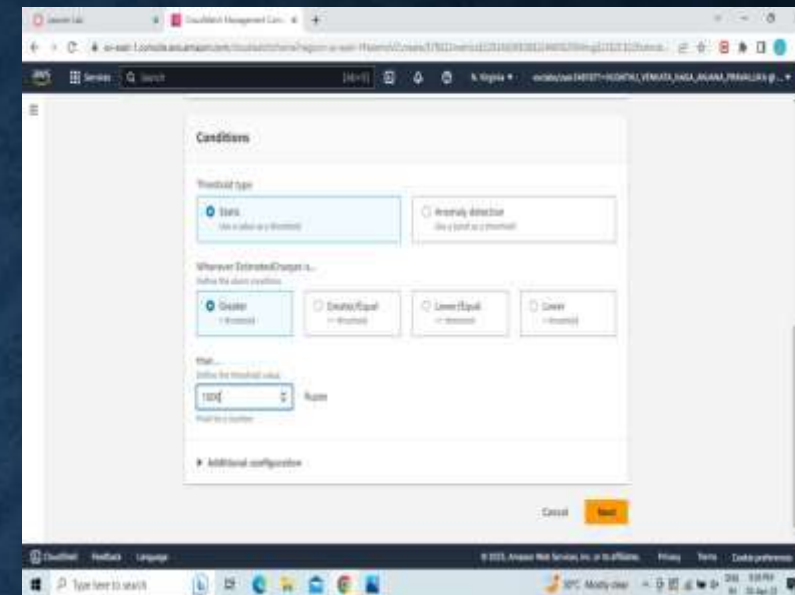
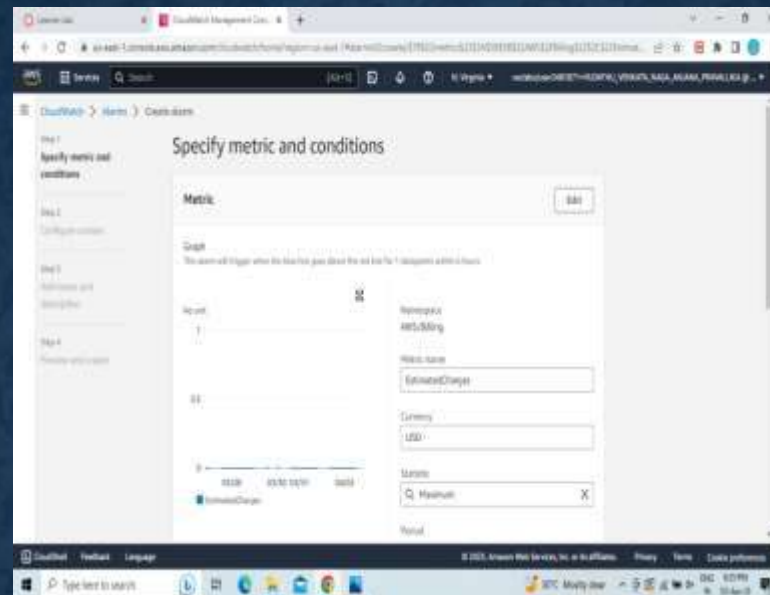
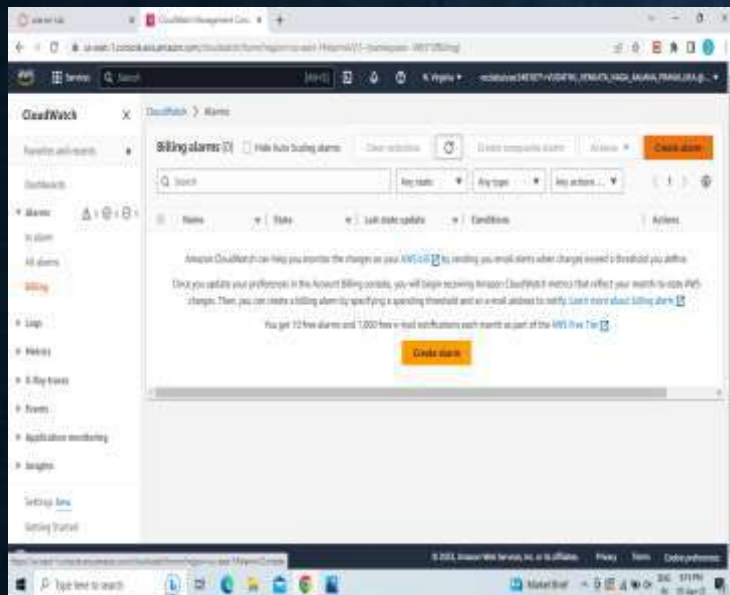
2.Then Click on CREATE ALARM.

3. Then follow the steps.

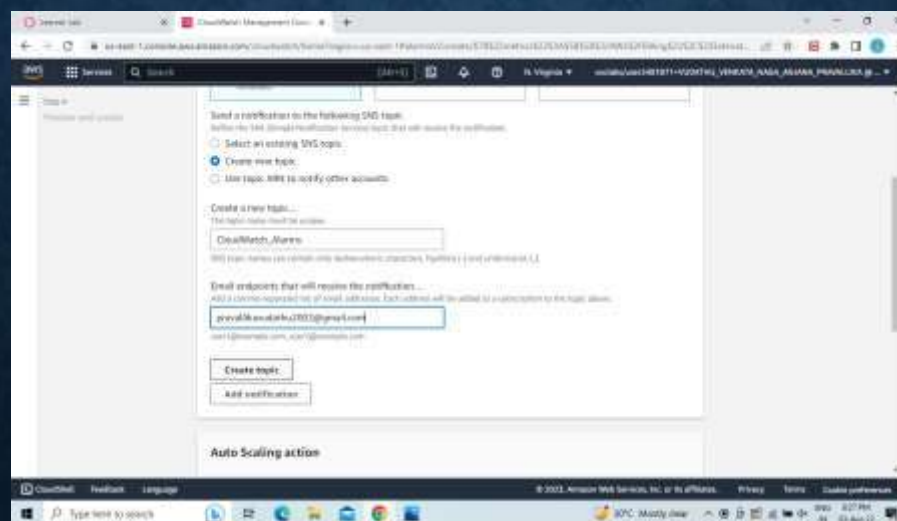
In the first step it will ask us to Specify metric and conditions.Click on Select Metric.

Change the Currency to Rupee.

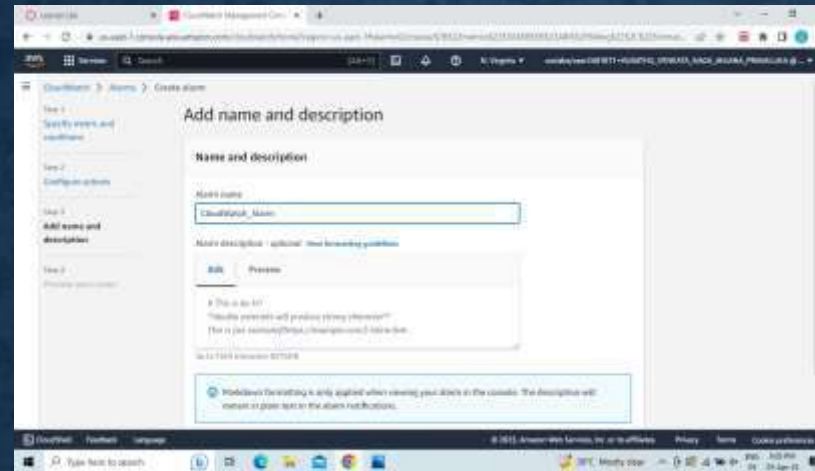
4.In the Conditions, section choose the EstimatedCharges like Greater/GreaterEqual/Lowerequal/Lower and also define the threshold value.



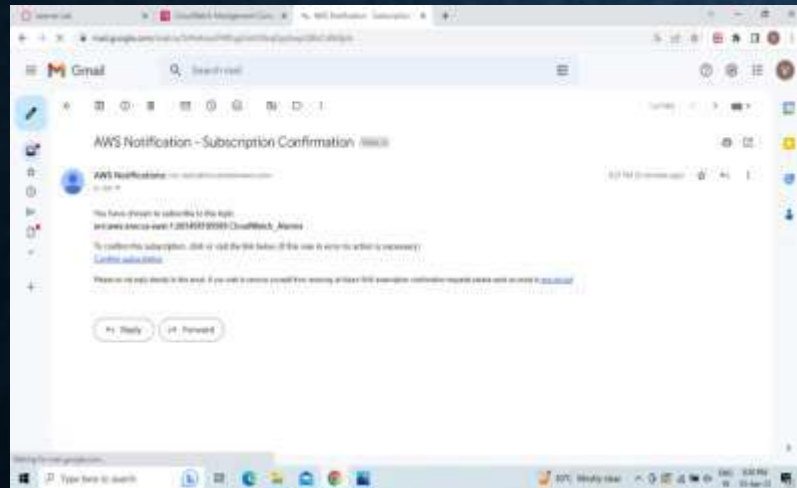
5. Now for Configure Actions choose to Create new topic. Give a name to the topic and enter your email to receive a notification. Click on Create Topic, then Next



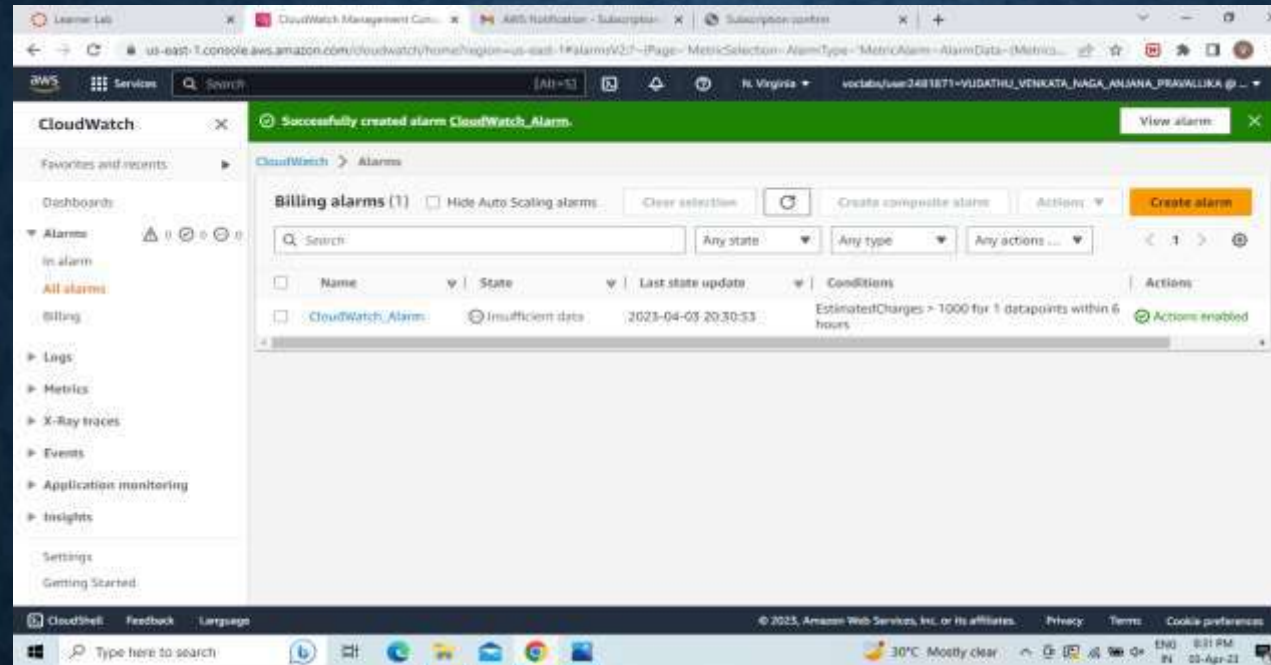
6. Give a name to your Alarm and Click on next.



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.

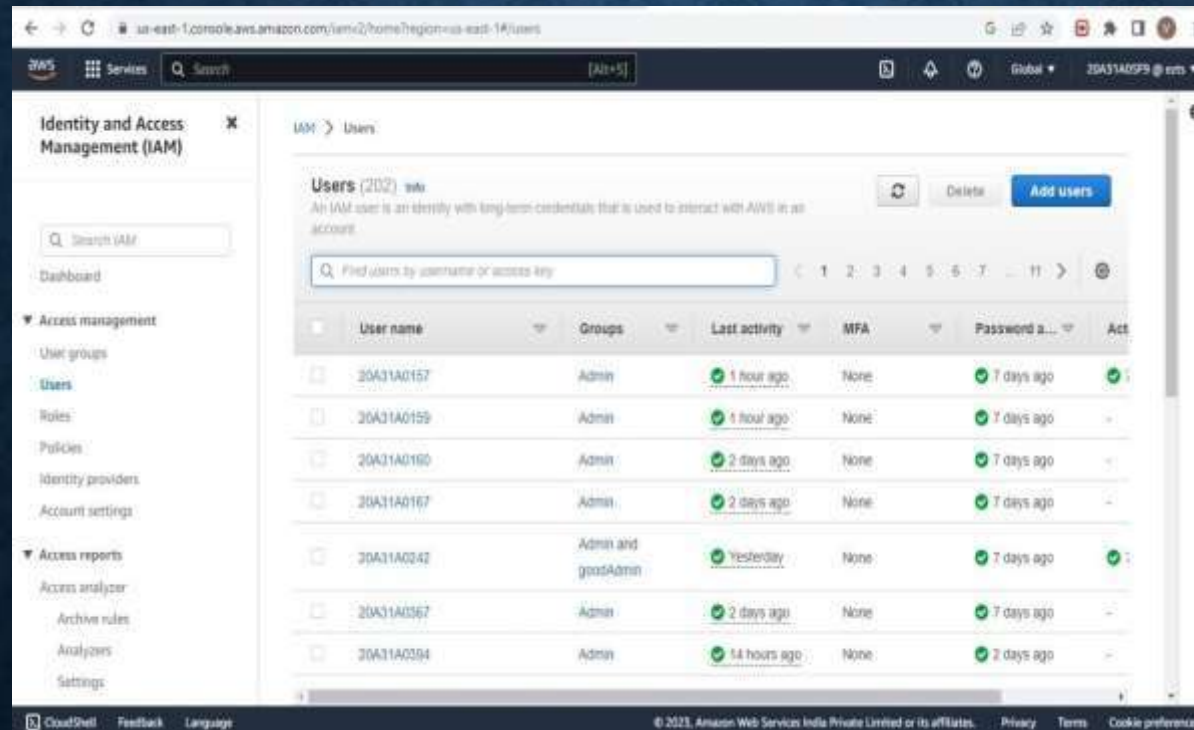


8. Preview the details you have entered .
9. Click on Create alarm. This will Create your Alarm.



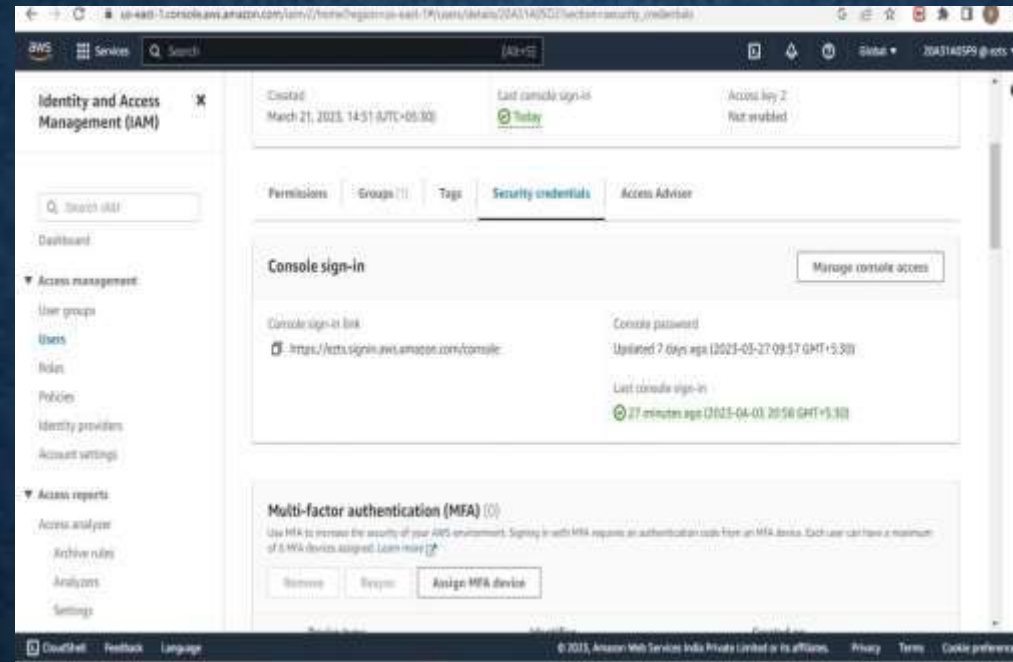
AWS COMMAND LINE INTERFACE

- 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 the Security Credentials tab.



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

us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/users/details/20A31A05D2/brief-security-credentials

Identity and Access Management (IAM)

Access keys (7)

Create access key

Description	Status	Created	Last used	Last used region	Last used service
AKIATR4OXV3QNPAMUQ8M	Active	7 days ago	7 days ago	us-east-1	sam

SSH public keys for AWS CodeCommit (0)

Use SSH public keys to authenticate access to AWS CodeCommit repositories. You can have a maximum of five (5) public keys (active or inactive) at a time. Learn more.

us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/users/details/20A31A05D2/brief-access-key

Access key best practices & alternatives

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

- Command Line Interface (CLI)**
You plan to use this access key to enable the AWS CLI to access your AWS account.
- Local code**
You plan to use this access key to enable application code running on a local development environment to access your AWS account.
- Application running on an AWS compute service**
You plan to use this access key to enable application code running on any AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
- Third-party service**
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
- Application running outside AWS**

us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/users/details/20A31A05D2/create-access-key

IAM > Users > 20A31A05D2 > Create access key

Set description tag - optional

The description for this access key will be attached to this user as a tag and shown alongside the access key.

Description tag value

Describe the purpose of this access key and where it will be used. A good description will help you obtain this access key successfully later.

Maximum 256 characters. Allowed characters are letters, numbers, spaces, hyphens, and underscores.

Cancel Previous Create access key

us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/users/details/20A31A05D2/create-access-key

Access key created

This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

IAM > Users > 20A31A05D2 > Create access key

Retrieve access keys

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIATR4OXV3QDSD6M2Z Show

Access key best practices:

- Never share your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

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)

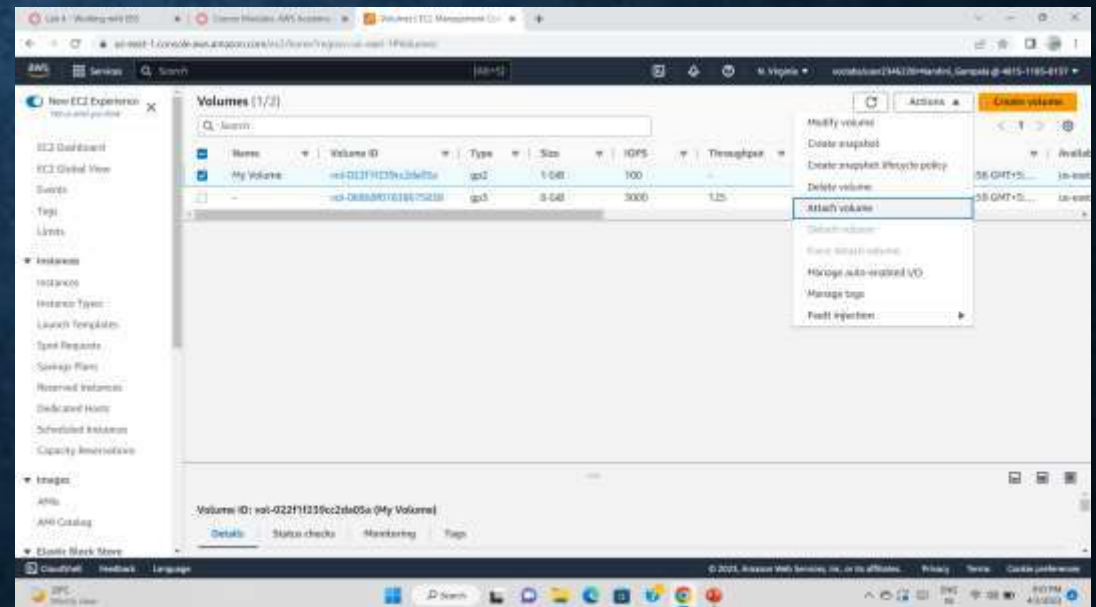
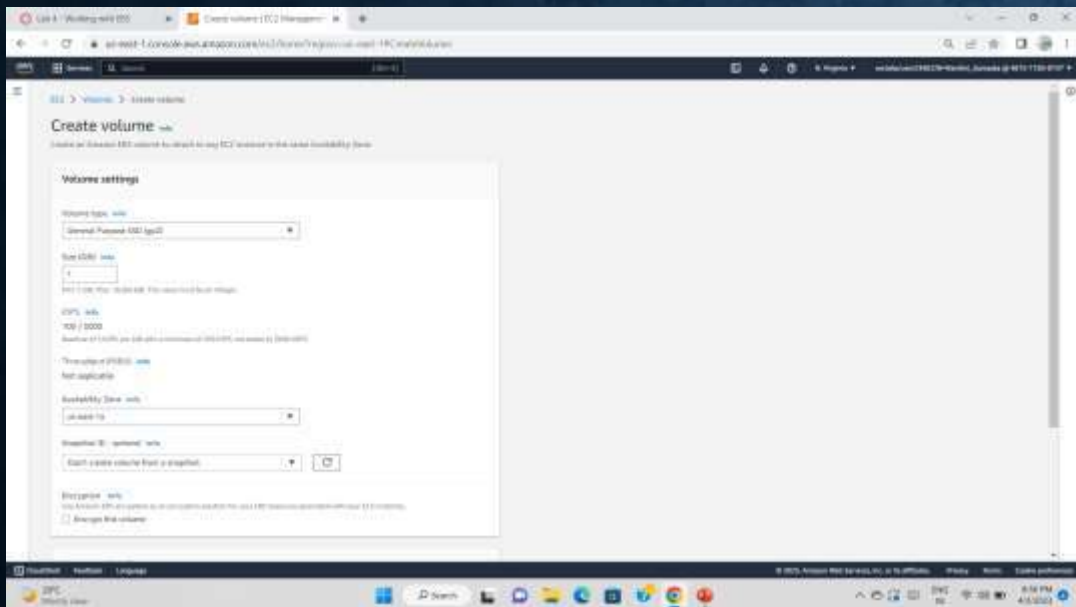
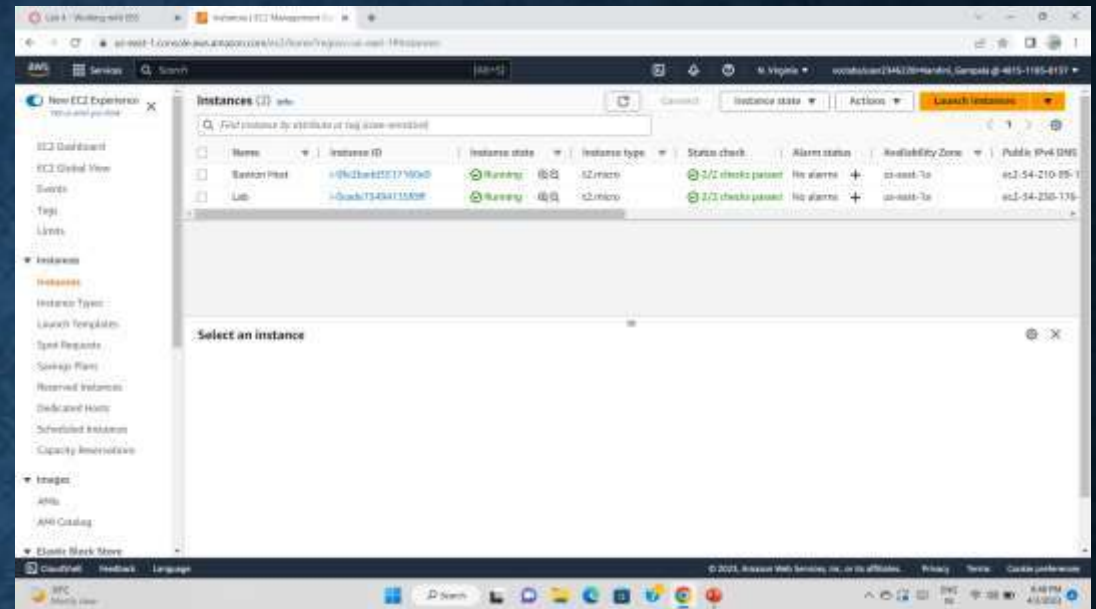
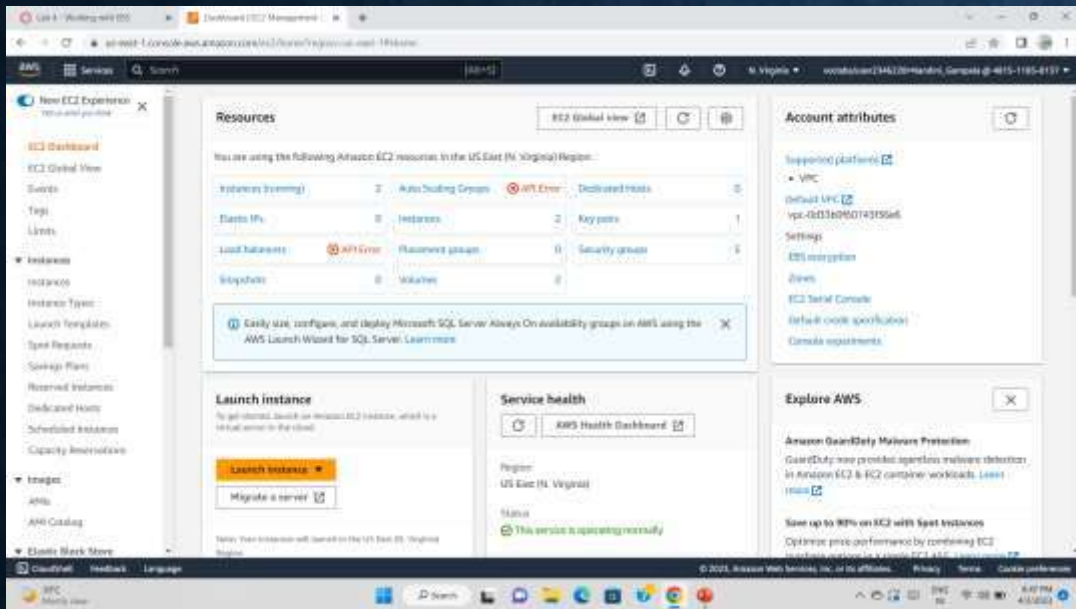
```
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

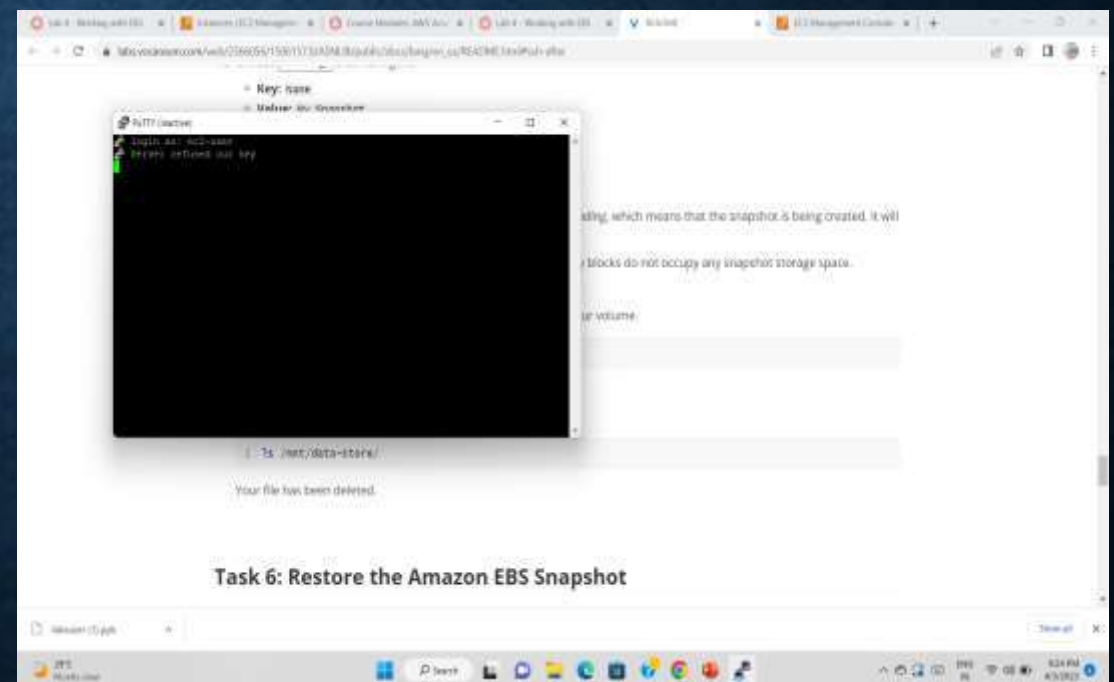
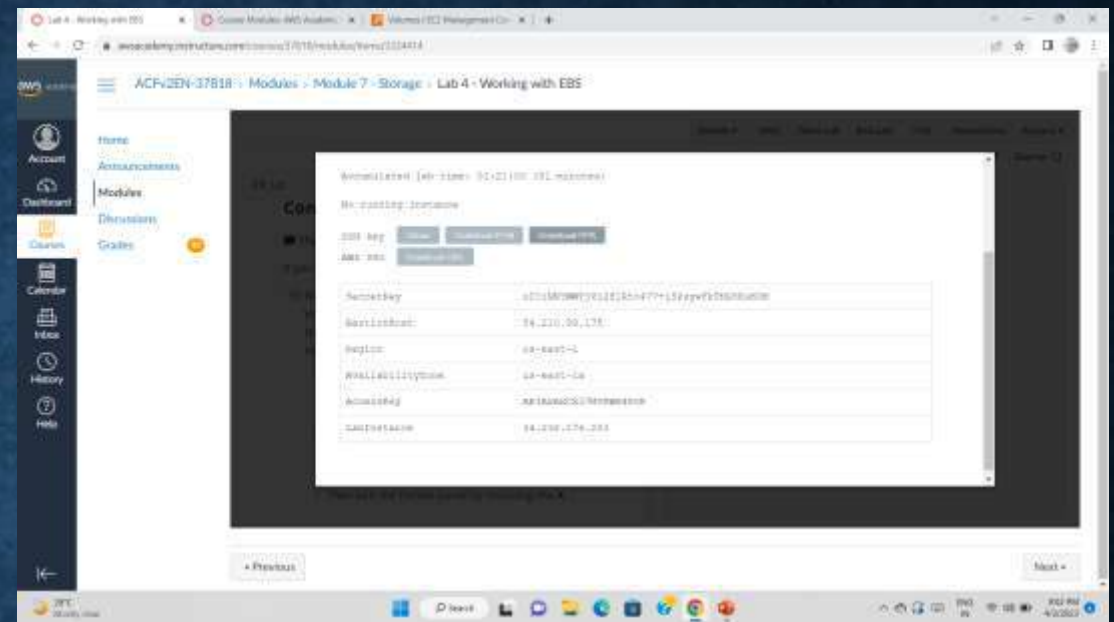
C:\Users\sivas>aws configure
AWS Access Key ID [None]: AKIATR40XV3QD5GD6MZZ
AWS Secret Access Key [None]: vMQP4GL99CbDSxsPWSgiTkkozMiRsUUZ0i+hDdNT
Default region name [None]: us-east-1
Default output format [None]: json
```

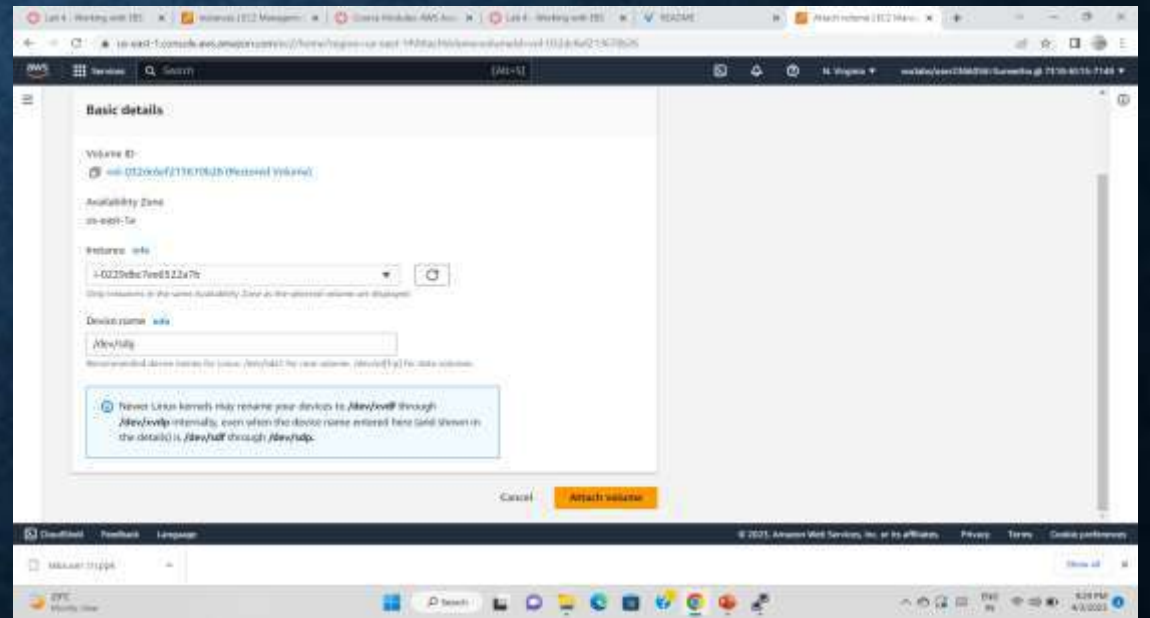
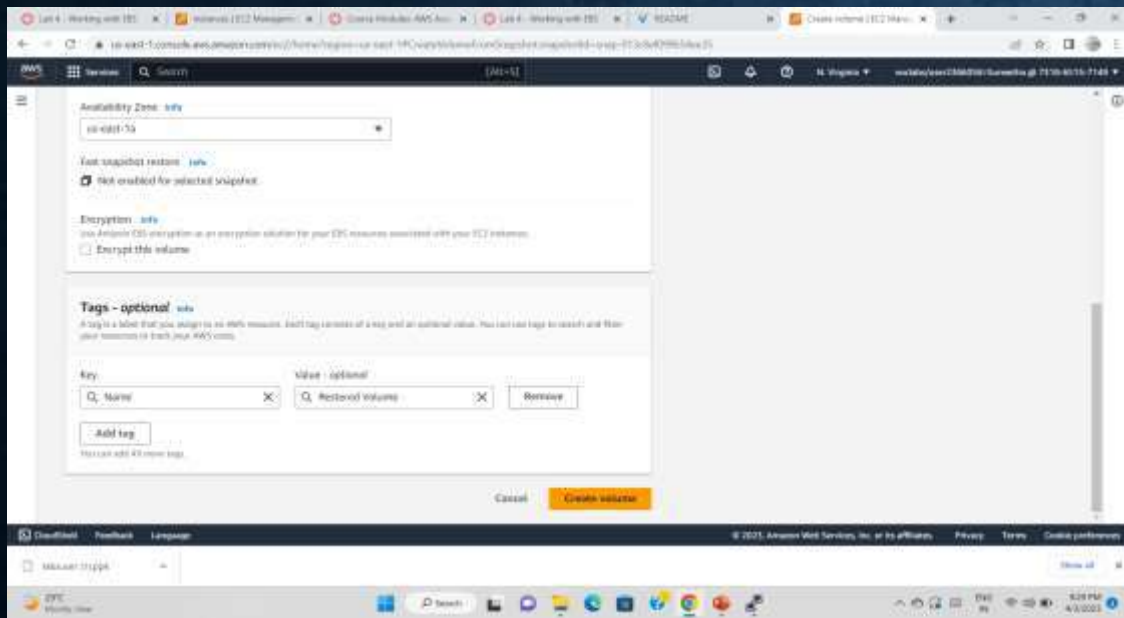
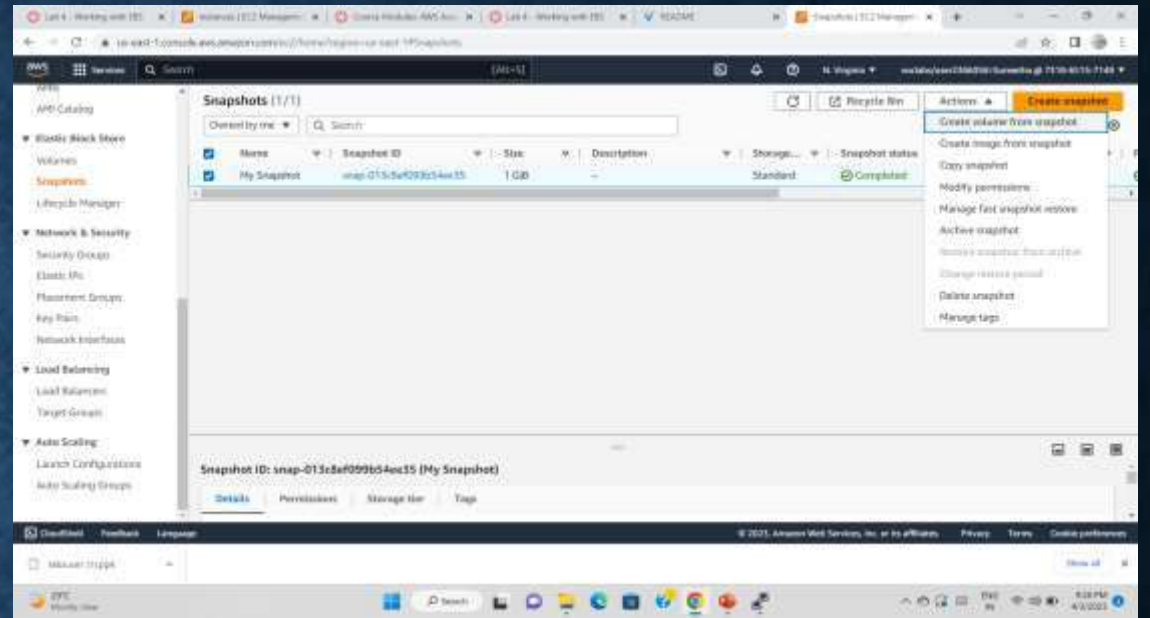
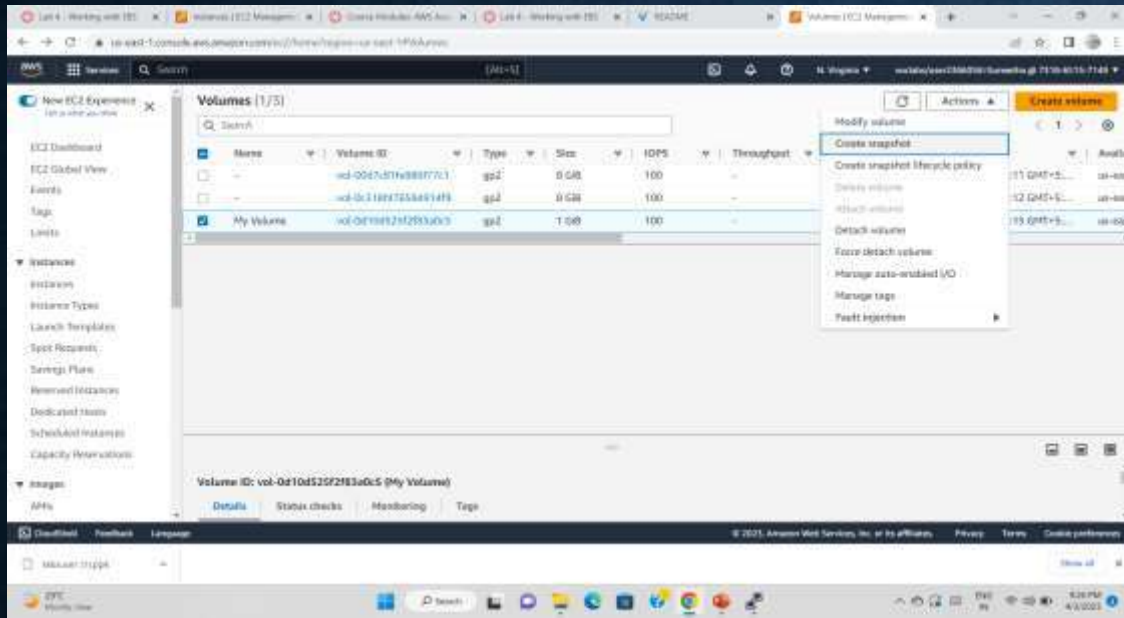
Finally we get Javascript Object Notation of all the users as output

CREATING A EBS VOLUME

1. Open Management Console, on the services menu, open Ec2
2. In the left navigation pane choose instances and create an instance with a name
3. Next, In the left navigation pane choose Volumes
4. Click on Create Volume
5. Select volume type, size(Gib), and Availability Zone, and in Add tag section add key and value names.
6. Then click on create volume
7. Click on volumes on the left navigation pane select the created volume and attach a previously created instance to it.
8. Download the ppk file
9. Then, go to the "Details" drop-down, choose "show"
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 look like: `df -h`
`sudo mkfs -t ext3/dev/sdf` etc.,
14. Create an EBS snapshot by giving the necessary fields.
15. Create a volume using a snapshot.
16. Attach the volume to the created EC2 instance

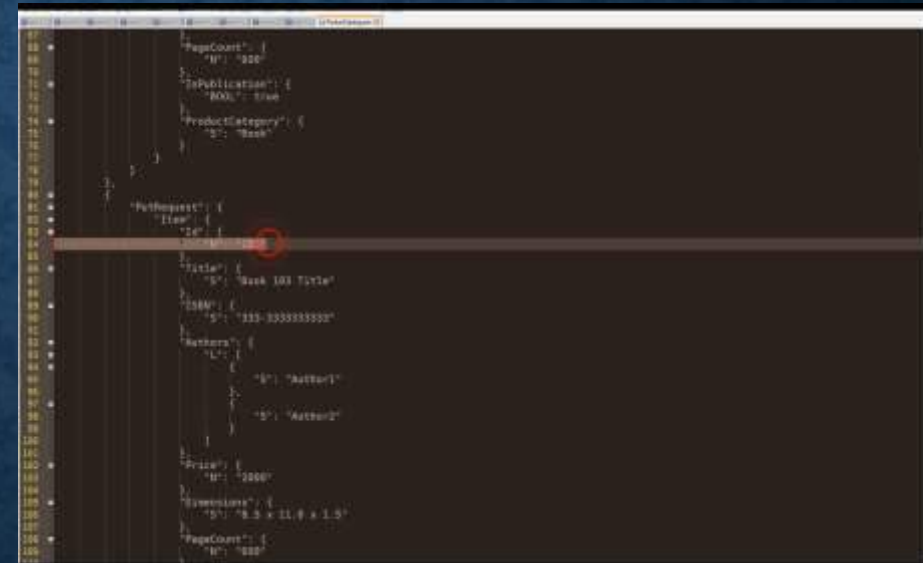
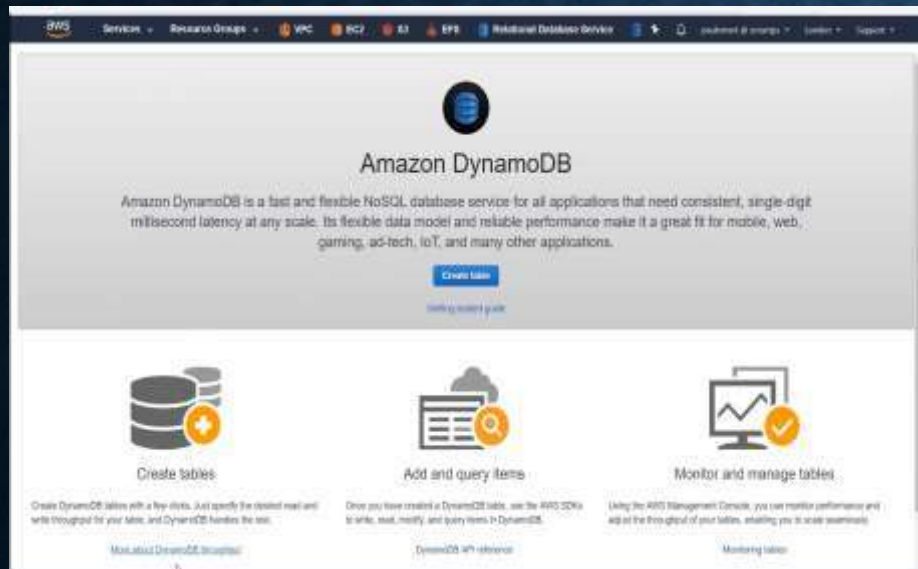




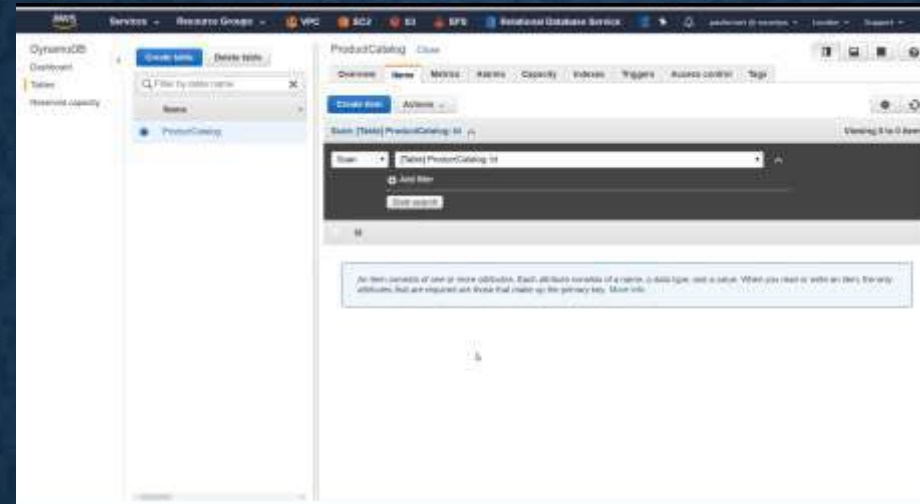


SETTING UP AMAZON DYNAMO DB

1. Setting up the Amazon DynamoDB
2. Here, we will be having a JSON file which is a product catalog
3. The products have a lot of different attributes and it is only common.
4. The interface looks like this:

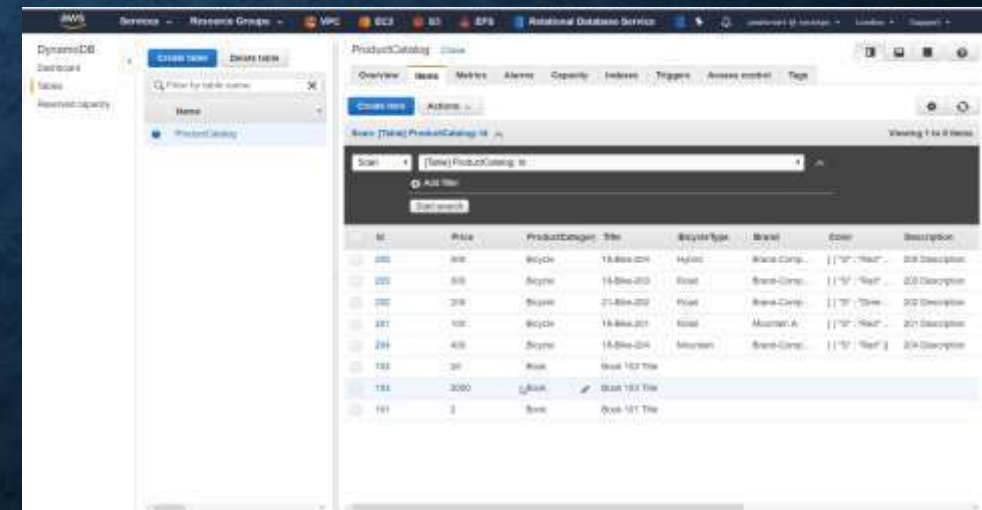


5. After creating the table, we can see that there are no items present.



6. So we will use the CLI to populate the table. Open PowerShell of AWS.

```
Windows PowerShell for AWS
C:\> aws dynamodb list-tables --region eu-west-2
{
  "TableNames": [
    "ProductCatalog"
  ]
}
C:\> aws dynamodb describe-table --table-name ProductCatalog --region eu-west-2
{
  "Table": {
    "TableName": "ProductCatalog",
    "TableArn": "arn:aws:dynamodb:eu-west-2:489284224315:table/ProductCatalog",
    "AttributeDefinitions": [
      {
        "AttributeName": "id",
        "AttributeType": "S"
      }
    ],
    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 5,
      "WriteCapacityUnits": 5
    },
    "TableSizeBytes": 0,
    "TableStatus": "ACTIVE",
    "KeySchema": [
      {
        "KeyType": "HASH",
        "AttributeName": "id"
      }
    ],
    "ItemCount": 0,
    "CreationDateTime": 1521926613.794
  }
}
C:\> aws dynamodb batch-write-item --request-items file://ProductCatalog.json --region eu-west-2
```



DynamoDB

Dashboard

Tables

Reserved capacity

Create table Delete table

Filter by table name

Name

ProductCatalog

ProductCatalog Close

Overview Items Metrics Alarms Capacity Instance Triggers Access control Tags

Create item Actions

Query: [Table] ProductCatalog: id

Partition key

Number

204

Add filter

Sort

Ascending Descending

Attributes

All Projected

Get results Cancel changes

ID	Price	ProductCategory	Title	BicycleType	Brand	Color	Description
208	600	Bicycle	16-Bike-204	Hybrid	BrandComp...	[[{"S": "Red"}]	208 Description
203	300	Bicycle	16-Bike-203	Road	BrandComp...	[[{"S": "Red"}]	203 Description
202	200	Bicycle	21-Bike-202	Road	BrandComp...	[[{"S": "Orange"}]	202 Description
201	100	Bicycle	16-Bike-201	Road	Mountain B...	[[{"S": "Red"}]	201 Description
204	400	Bicycle	16-Bike-204	Mountain	BrandComp...	[[{"S": "Red"}]	204 Description
102	30	Book	Book 102 Title				
103	2000	Book	Book 103 Title				
101	2	Book	Book 101 Title				

DynamoDB

Dashboard

Tables

Reserved capacity

Create table Delete table

Filter by table name

Name

ProductCatalog

ProductCatalog Close

Overview Items Metrics Alarms Capacity Instance Triggers Access control Tags

Create item Actions

Query: [Table] ProductCatalog: id

Partition key

Number

204

Add filter

Sort

Ascending Descending

Attributes

All Projected

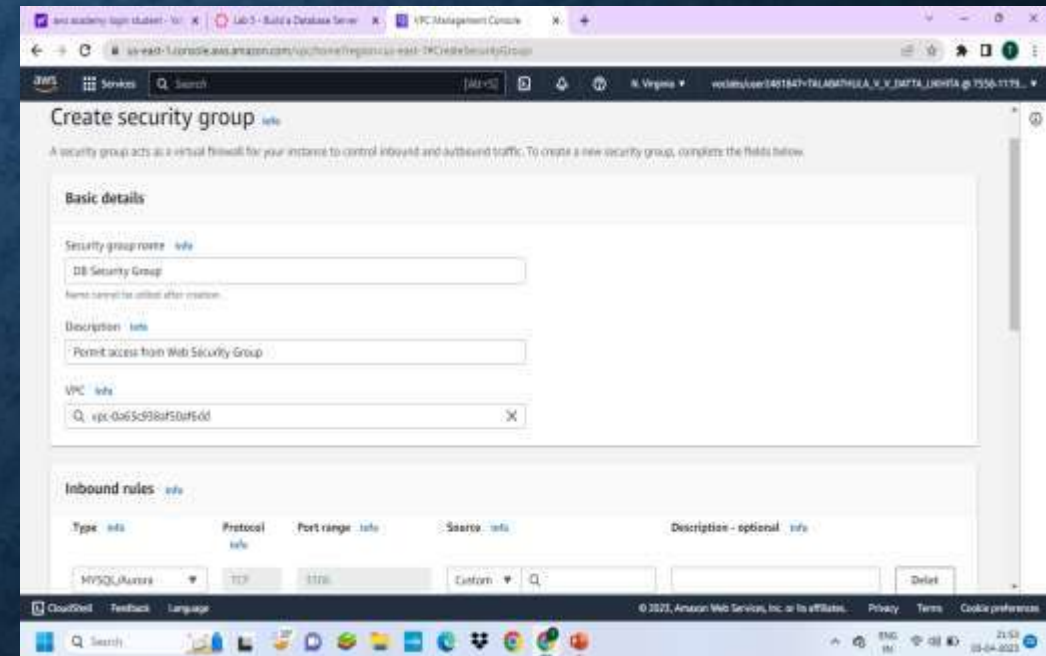
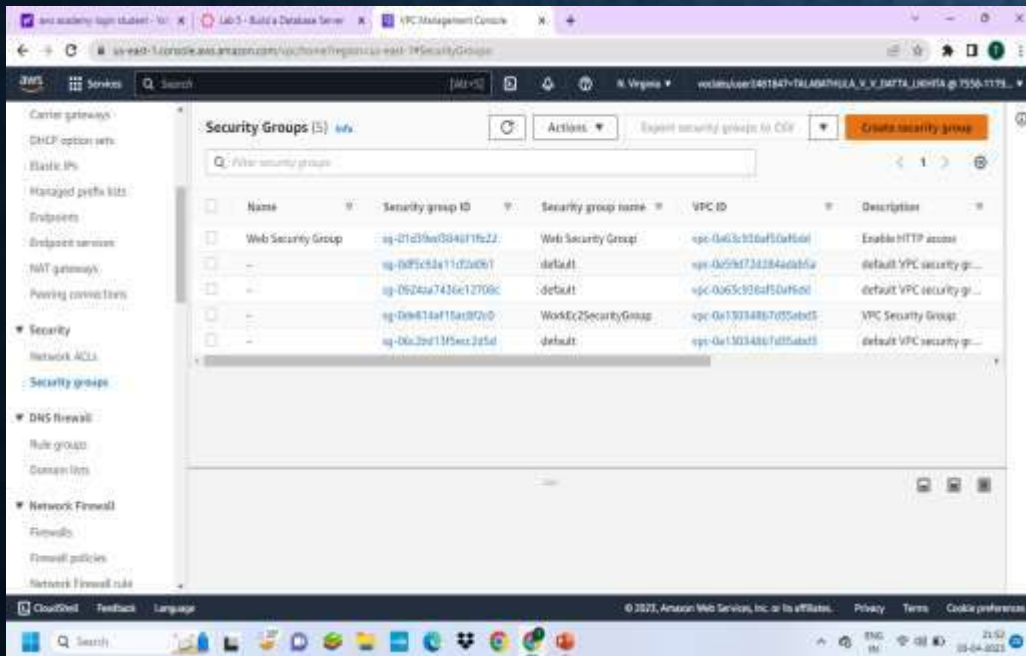
Get results

ID	Price	ProductCategory	Title	BicycleType	Brand	Color	Description
204	400	Bicycle	16-Bike-204	Mountain	BrandComp...	[[{"S": "Red"}]	204 Description

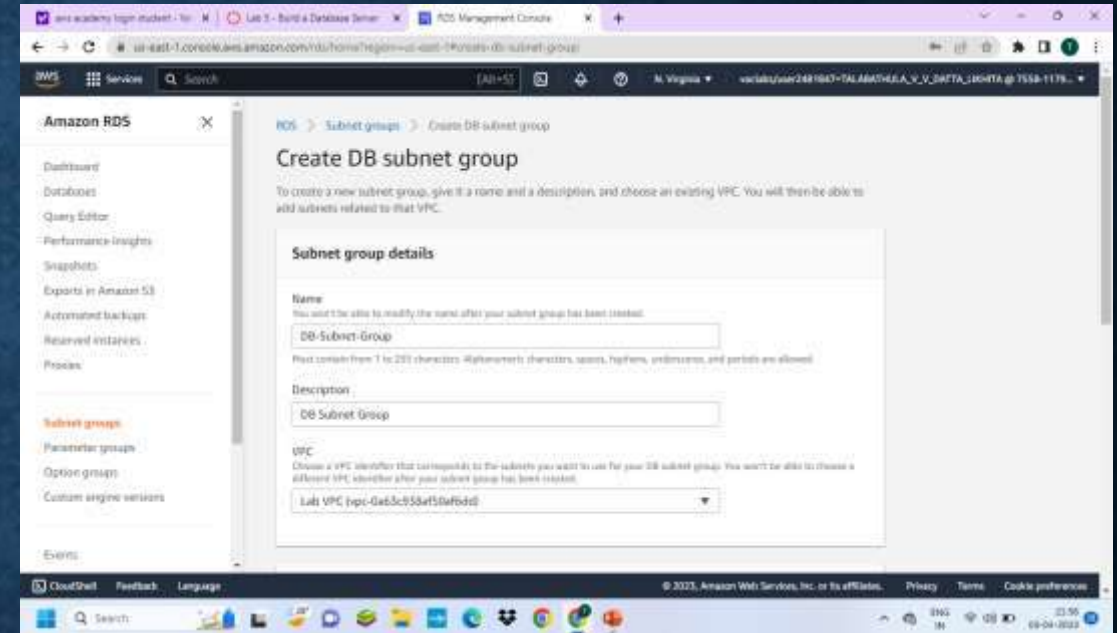
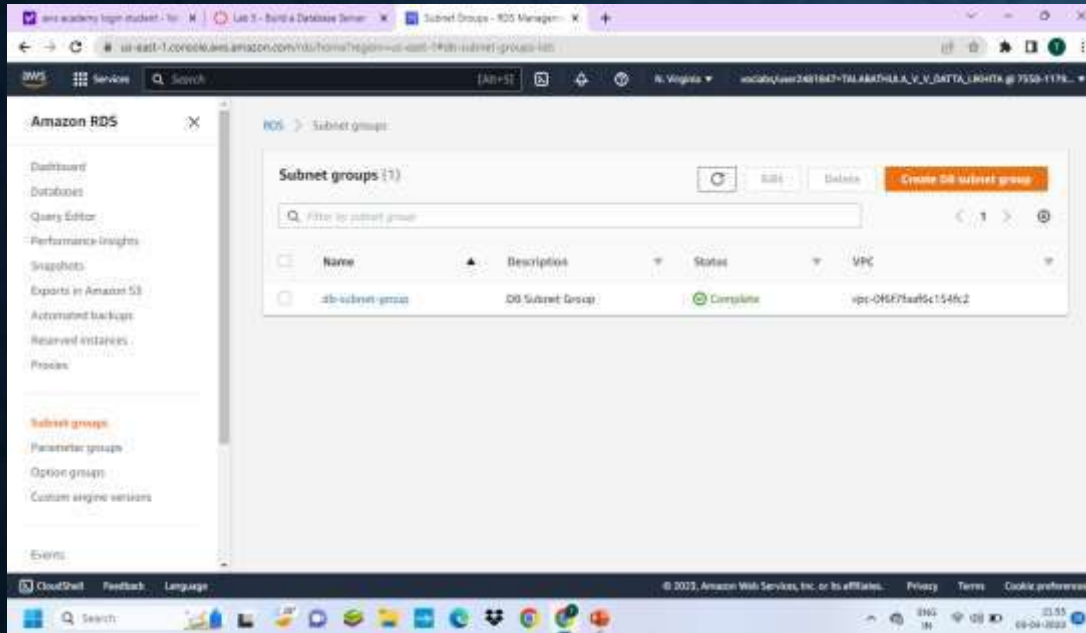
AWS RDS

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

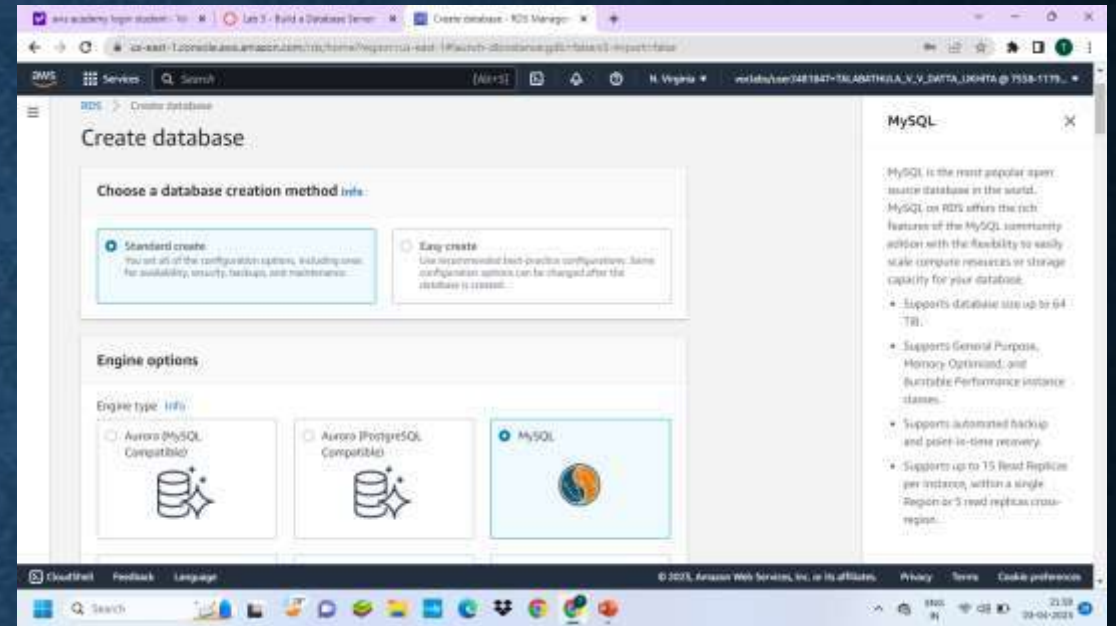
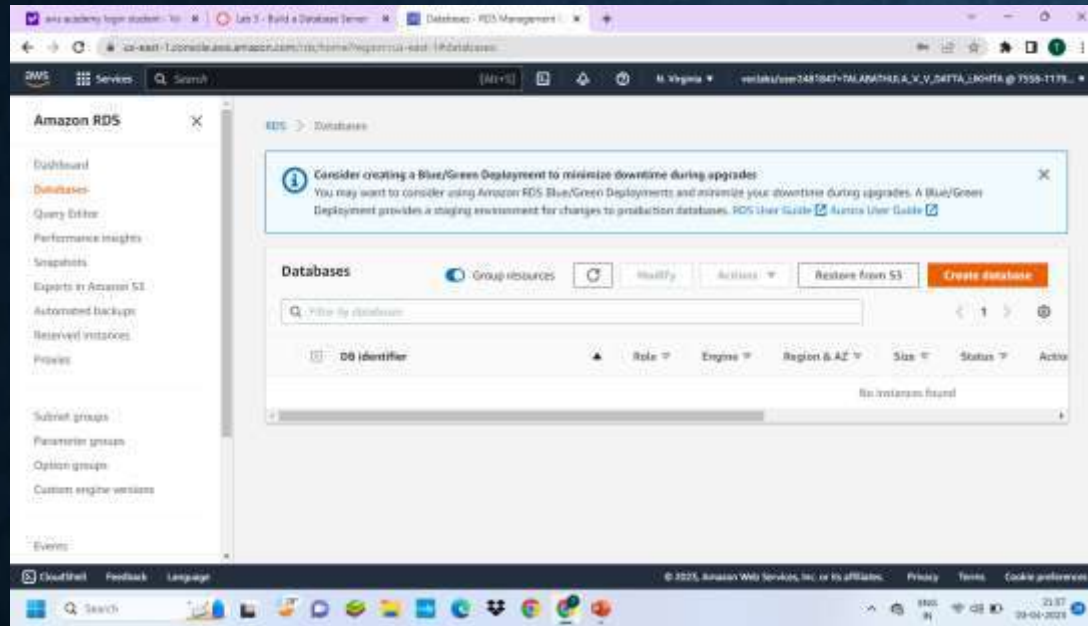
AWS management console → vpc → security groups → choose to create security group → and add an inbound rule → to create a security group.



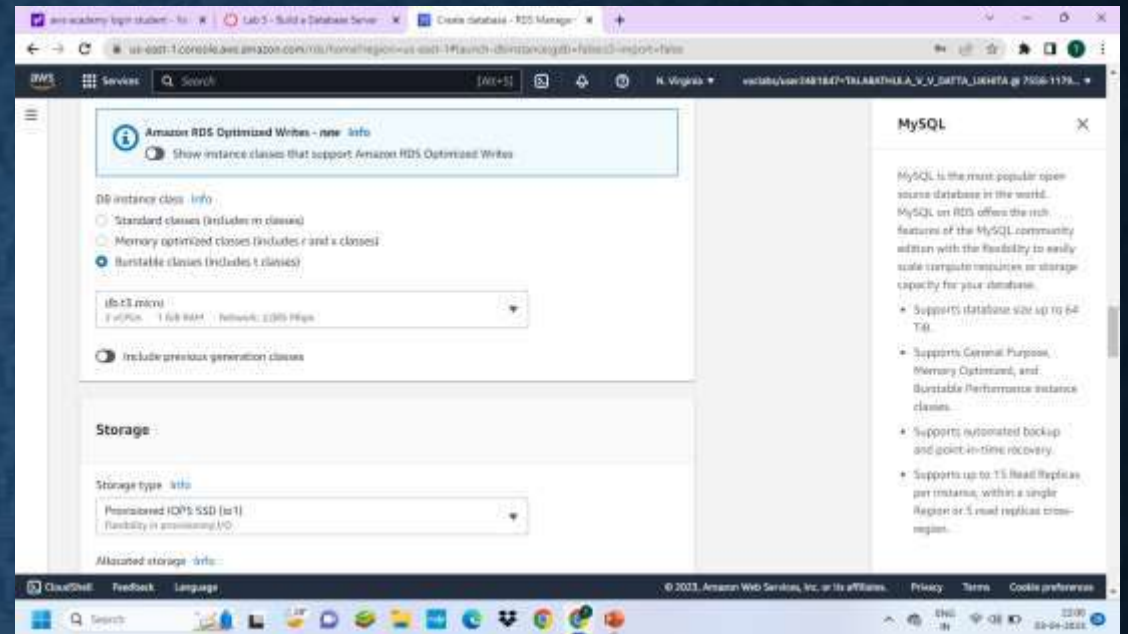
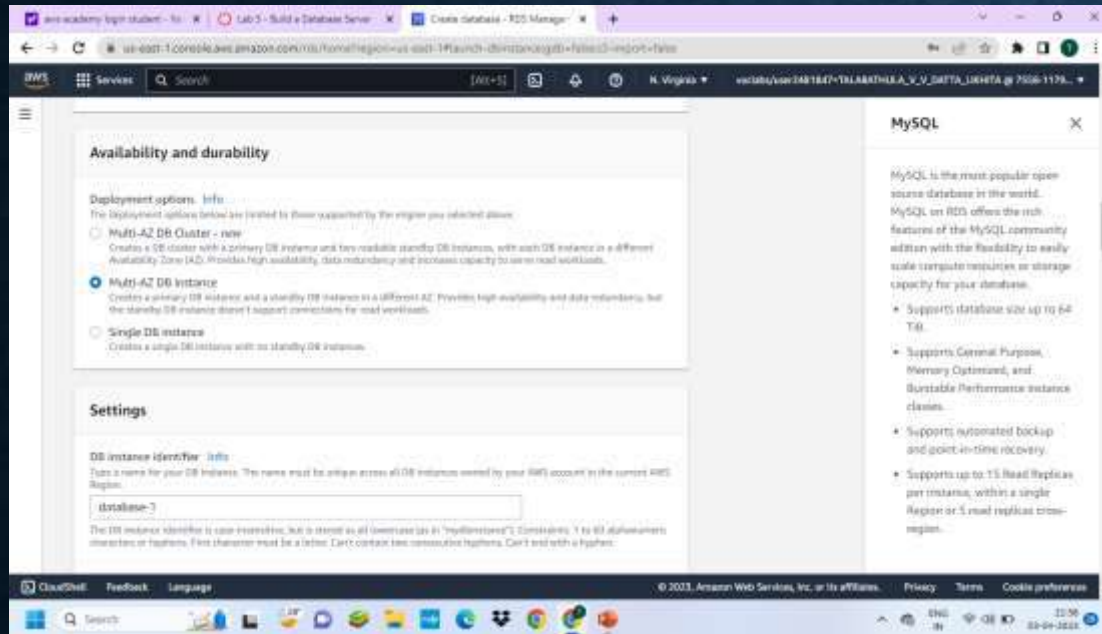
Step 2 : Create a DB Subnet Group.Rds → subnet groups → choose create DB subnet group → add subnets → create DB subnet group.



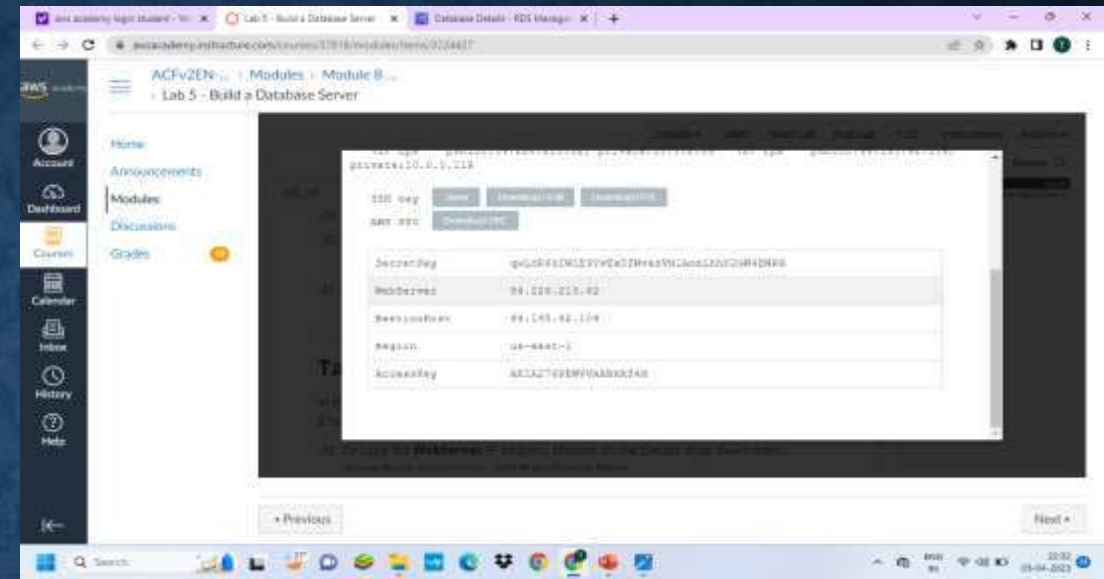
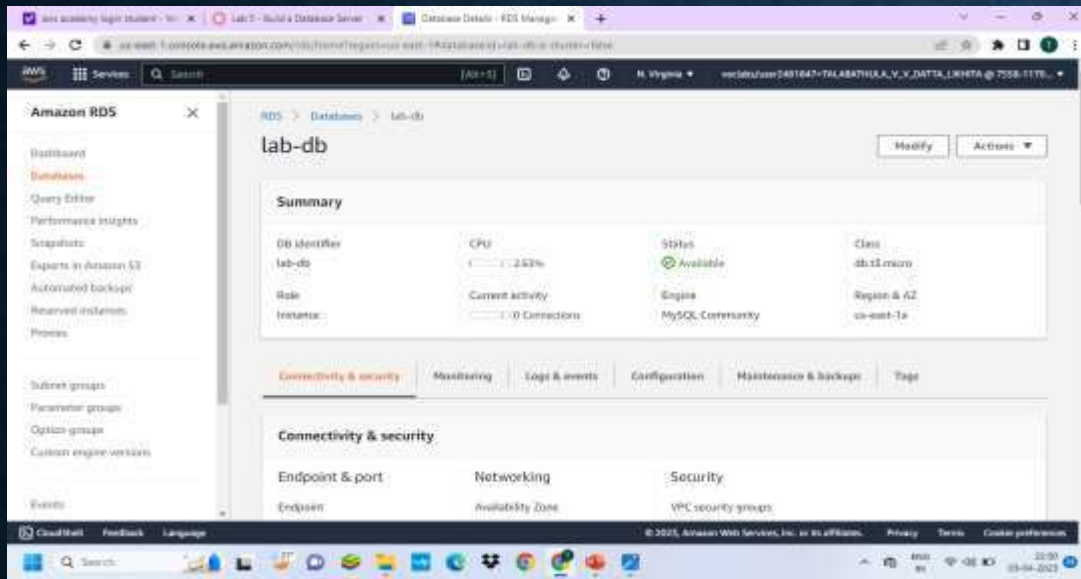
Step 3: In the left navigation pane, choose **Databases** → choose create database → MYSQL



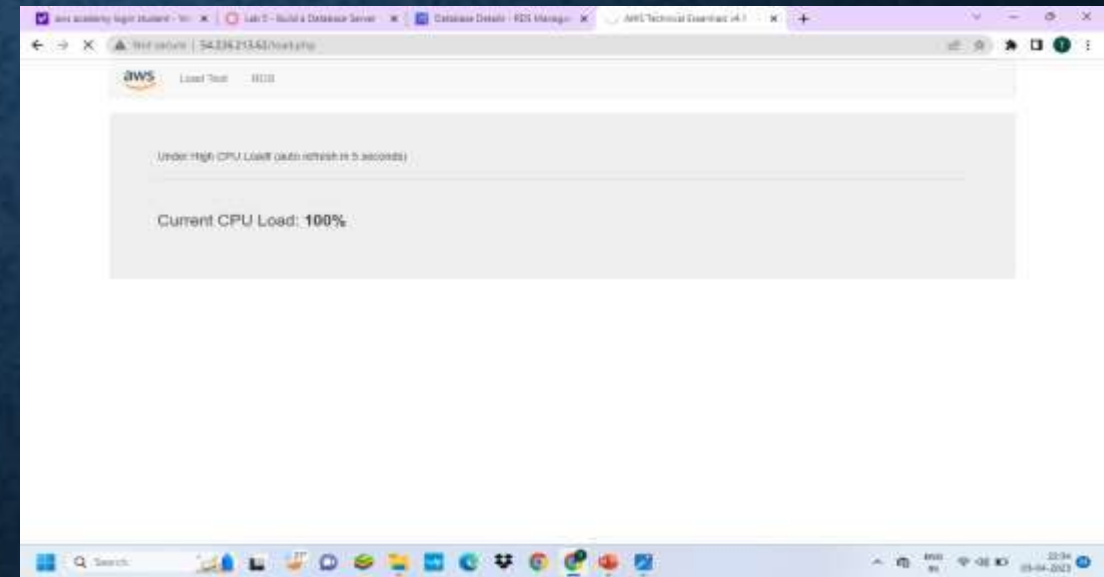
Step 4: In Availability and durability, choose Multi-AZ DB instance then configure settings, DB instance class, Storage, connectivity, choose existing vpc security group, and set up additional configuration.



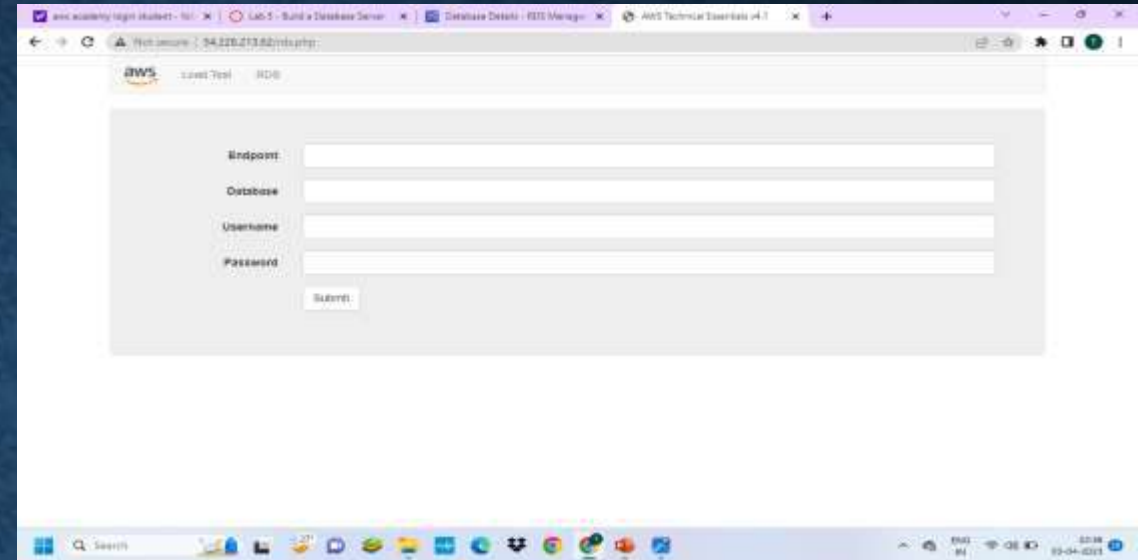
Step 5: Wait until Info changes to Modifying or Available.
Scroll down to the Connectivity & security section and copy the **Endpoint** field.



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.



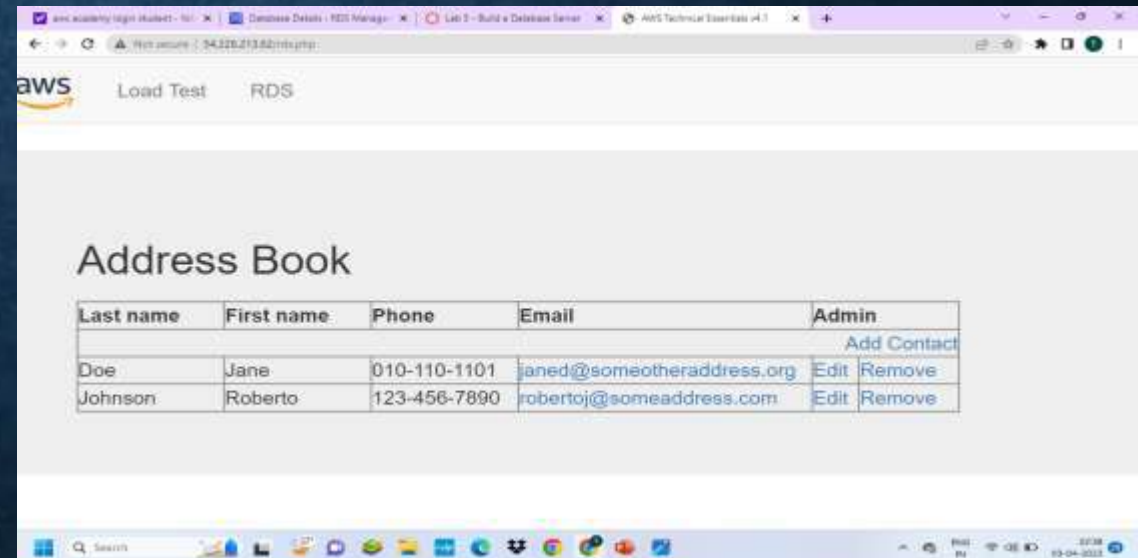
Step 7 : Choose the **RDS** link at the top of the page and configure the settings.



The screenshot shows the AWS console interface for configuring an RDS instance. At the top, there are tabs for 'Load Test' and 'RDS'. The 'RDS' tab is selected. Below the tabs, there is a form with the following fields:

- Endpoint:
- Database:
- Username:
- Password:
- Submit:

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.



The screenshot shows the AWS console interface for the 'Address Book' application. The application displays a table with the following columns: Last name, First name, Phone, Email, and Admin. The table contains two rows of data:

Last name	First name	Phone	Email	Admin
Doe	Jane	010-110-1101	janed@someotheraddress.org	Add Contact Edit Remove
Johnson	Roberto	123-456-7890	roberto@someaddress.com	Edit Remove

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)
8. Choose Add.

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`.

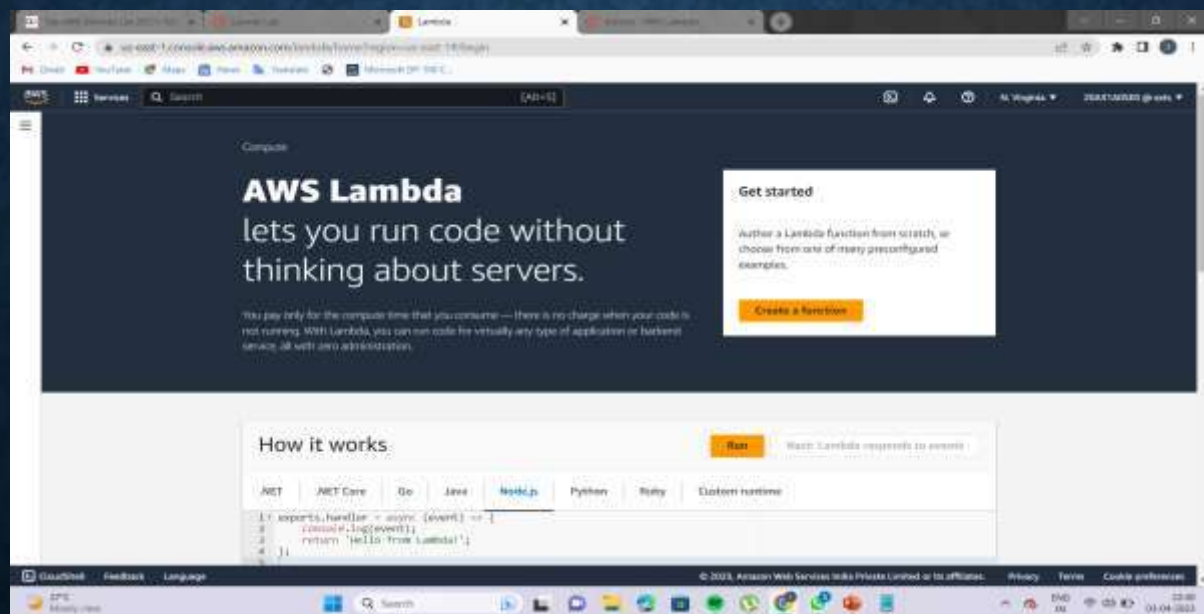
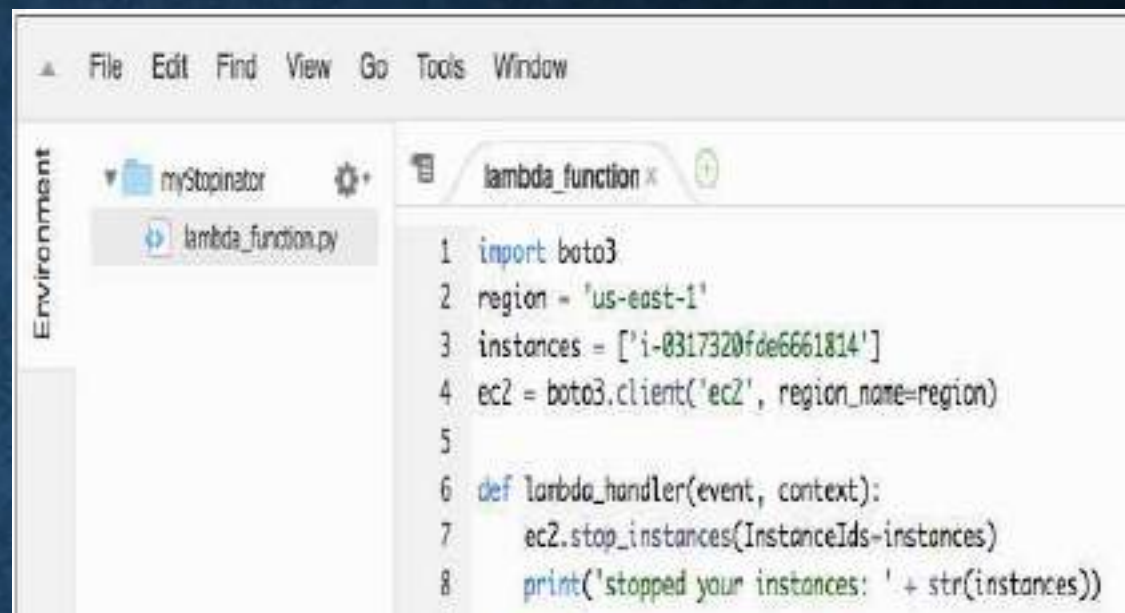
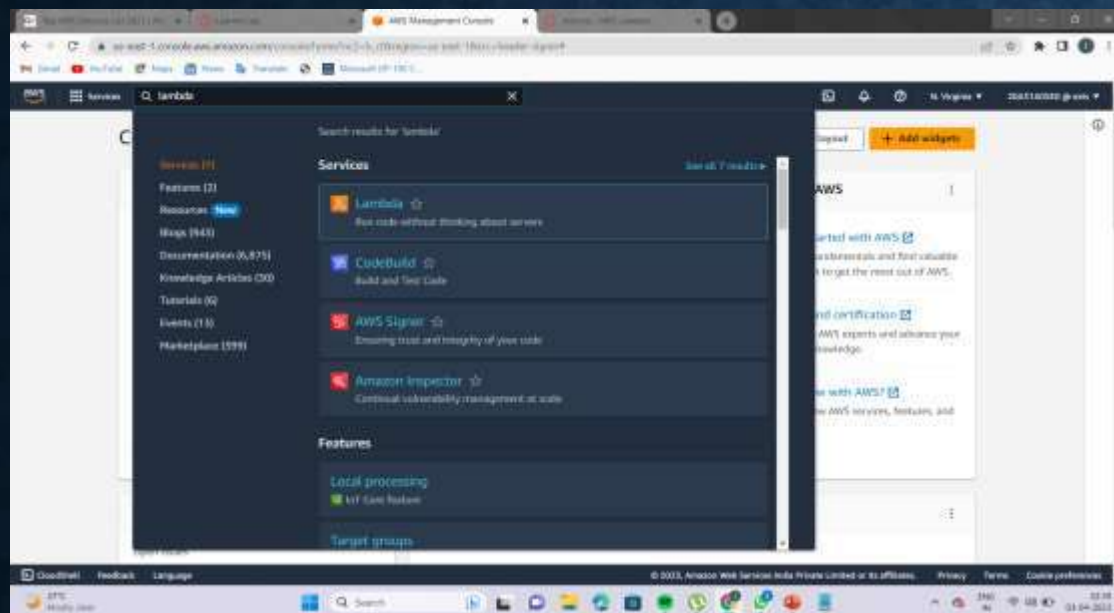
11. Verify that an EC2 instance named `instance1` is running in your account, and copy the `instance1` instance ID.

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.



ELASTIC LOAD BALANCER(ELB)

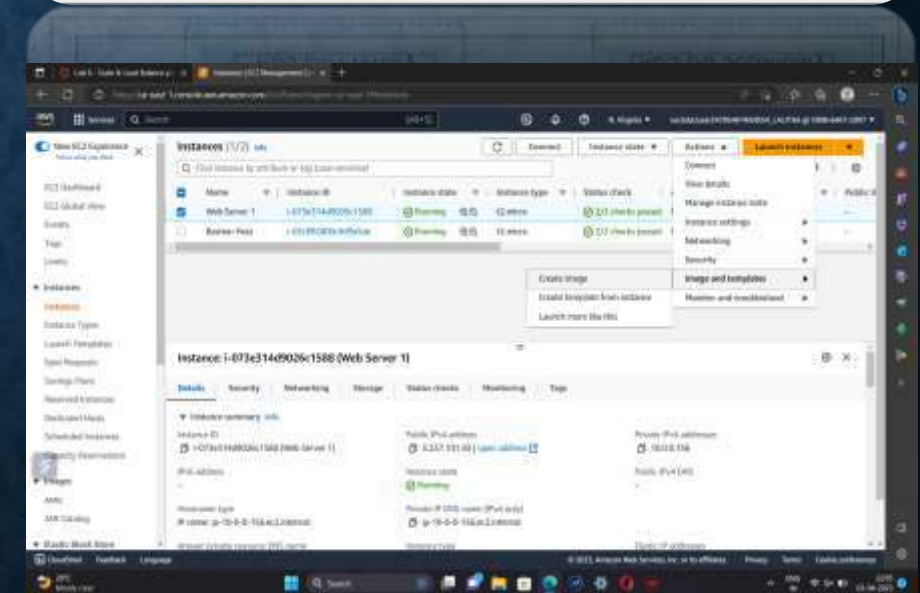
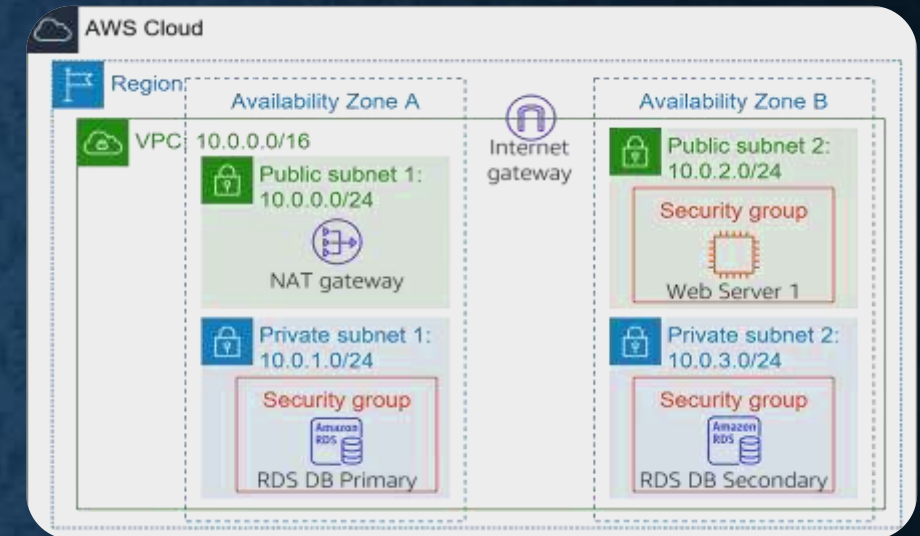
Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances

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

Procedure:

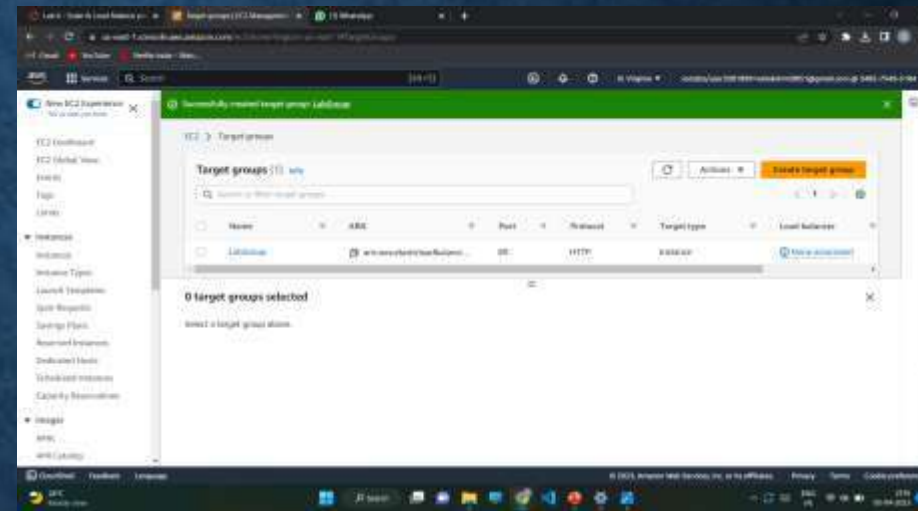
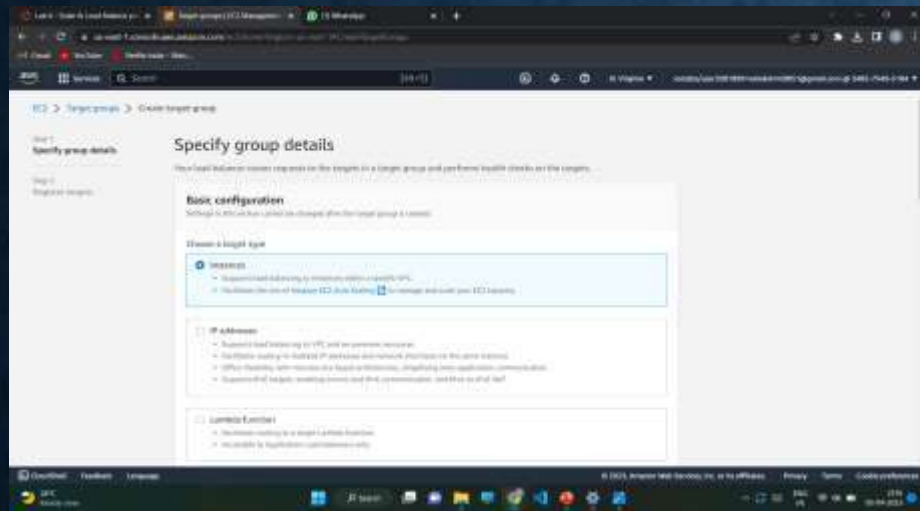
Task1: Creating an AMI for Auto Scaling

- ❖ Click start lab then click on AWS.
- ❖ You will navigate to the 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 an image. Name the image and give the description.
- ❖ Click create an 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**.

Load Balancers (ELB Management)

Application Load Balancer

Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Create

Network Load Balancer

Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

Create

Gateway Load Balancer

Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

Create

Load Balancers (ELB Management)

Security groups

A security group is a set of firewall rules that control the traffic to your load balancer.

Security groups

Select up to 5 security groups

Create new security group

Web Security Group: sg-8C5c8d7aTa2f8d83 X

Listeners and routing

A Listener is a protocol that checks for incoming requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

Listener: HTTP-80

Protocol: HTTP Port: 80

Default action: info

Forward to: LoadGroup: Target: Load Balancer: info

HTTP

Listener tags - optional

Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Load Balancers (ELB Management)

Add-on services

None

Tags

None

Activations

Content default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.

Cancel Create load balancer

Load Balancers (ELB Management)

Recommendation to not use launch configurations

Amazon EC2 Auto Scaling no longer adds support for new EC2 instances to launch configurations and will stop supporting new EC2 instance types after December 31, 2022. We recommend that customers using launch configurations migrate to launch templates. For more information, see the documentation.

Launch configurations

Search launch configurations

Name	AMI ID	Instance type	Spot price	Creation time
No launch configurations found in this region.				

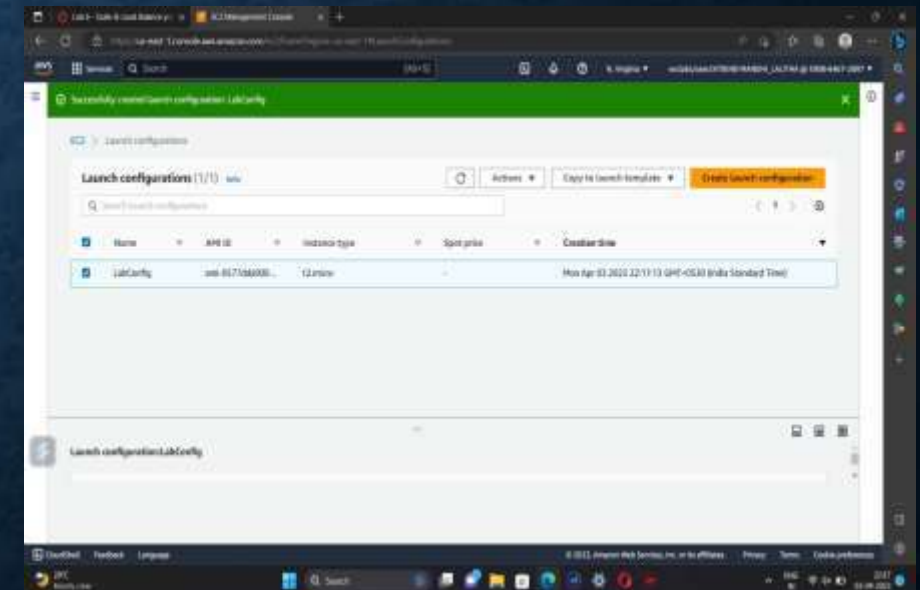
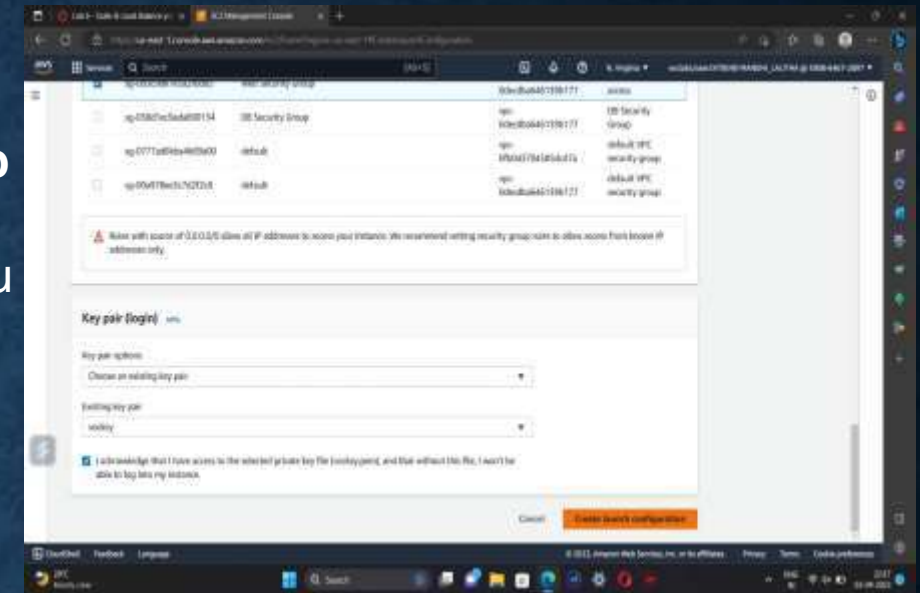
Create launch configuration

Select a launch configuration above

- ❖ Click create load balancer.

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

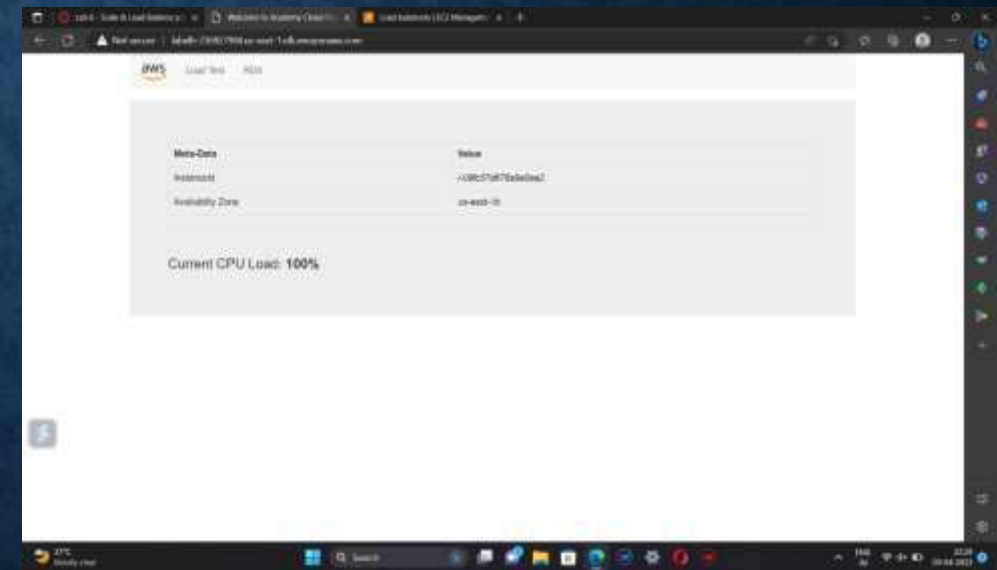
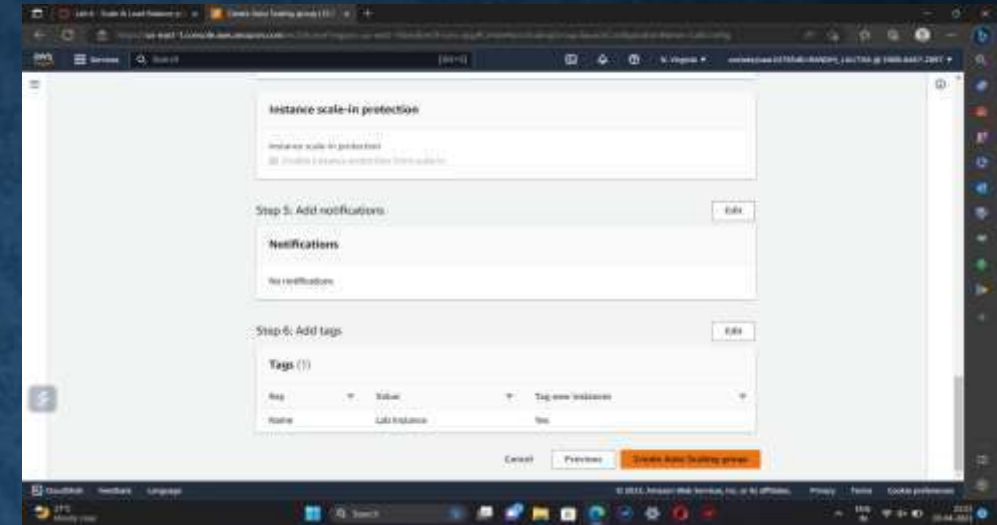
- ❖ In Launch Configurations, click create launch configuration.
- ❖ Name the configuration and for AMI choose web server AMI that you created in task 1.
- ❖ Select the instance type.
- ❖ Under Additional Configuration, for monitoring select Enable EC2 instance detailed monitoring within CloudWatch.
- ❖ 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.

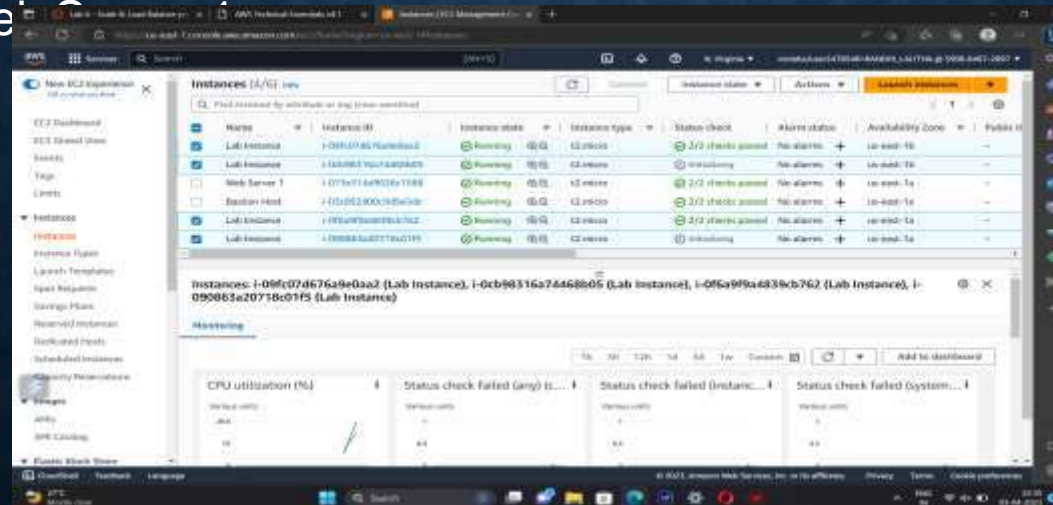
Task 4: Verify that Load Balancing is Working

- ❖ 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 We



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:

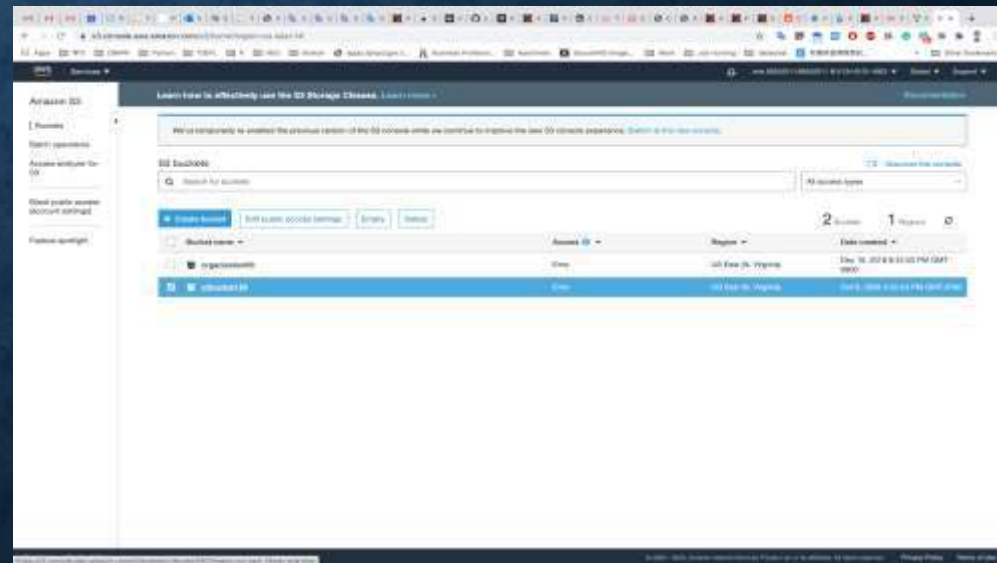
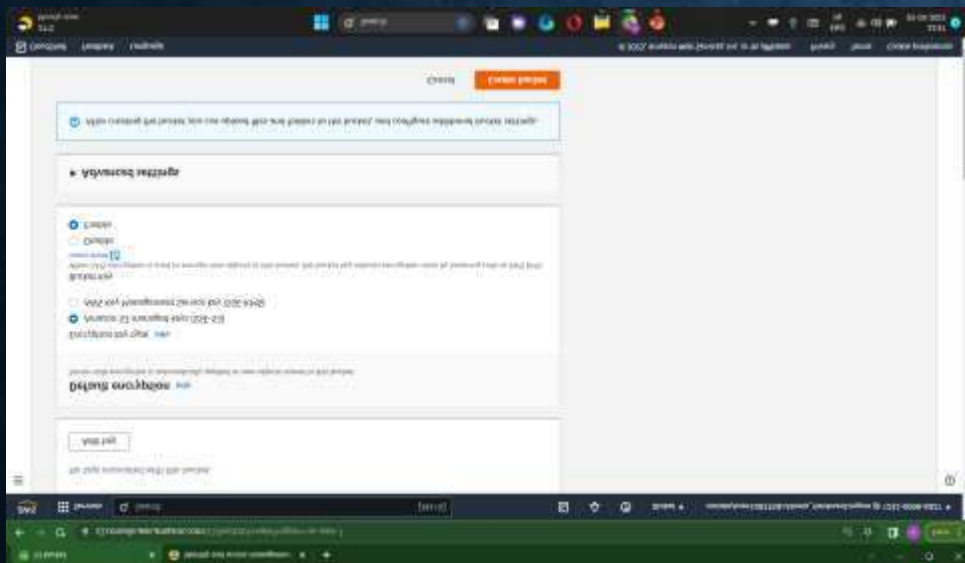
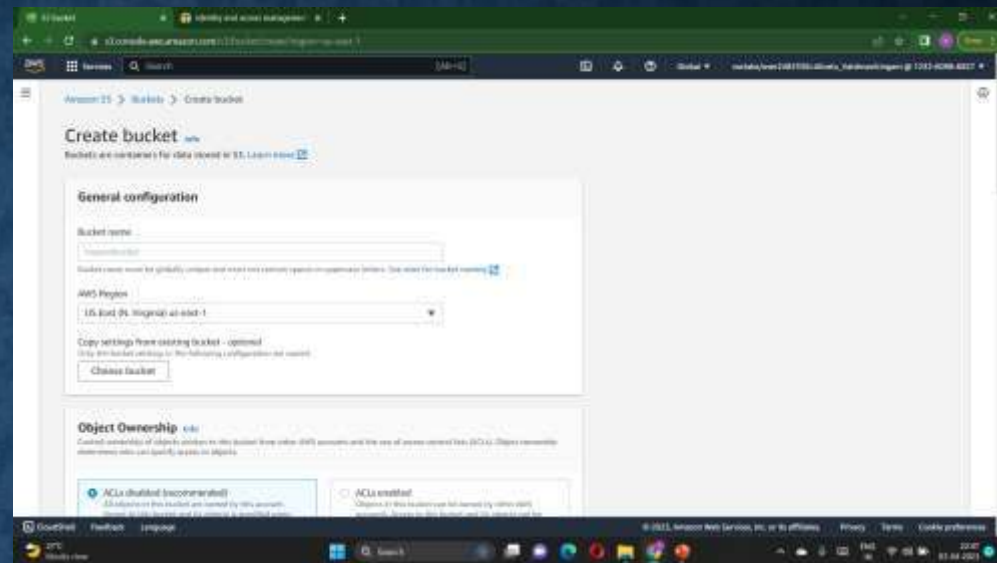
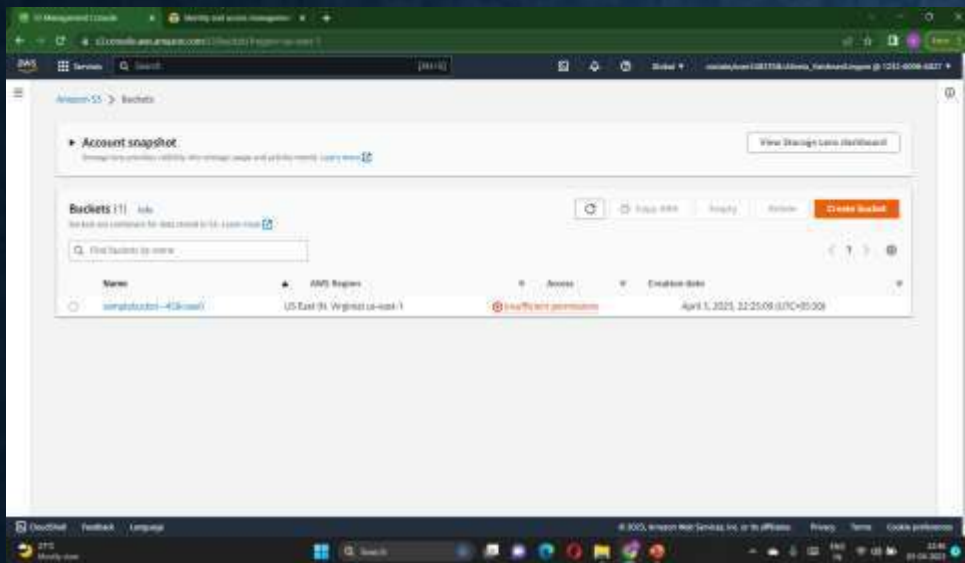
Step 10: Go back to your bucket and click on Permissions.

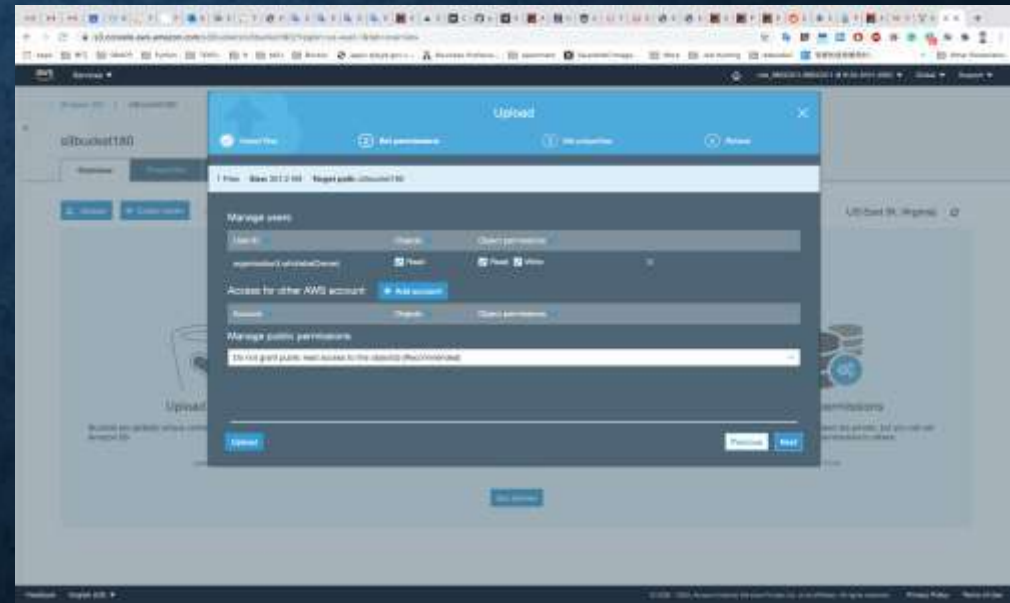
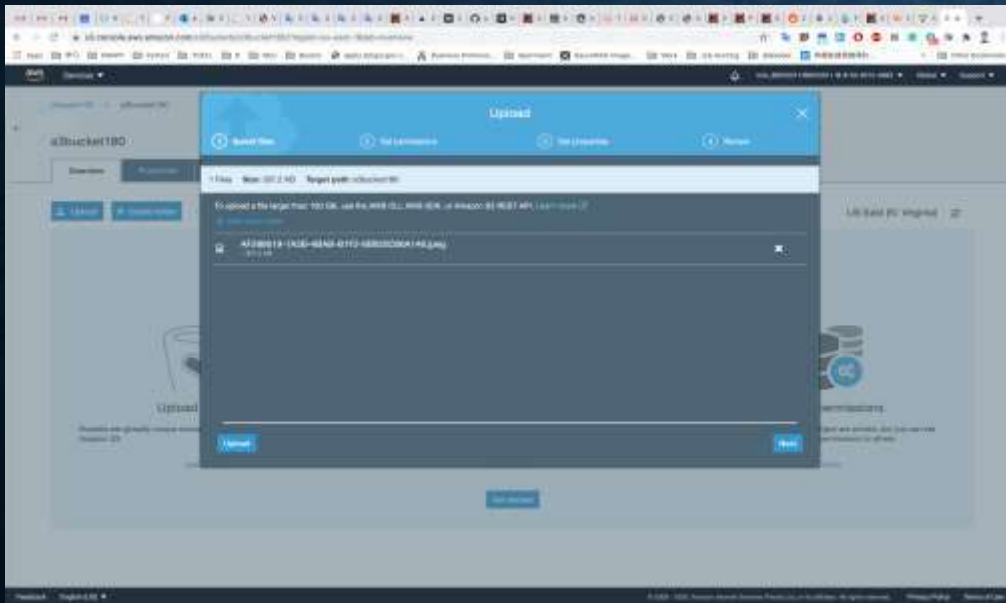
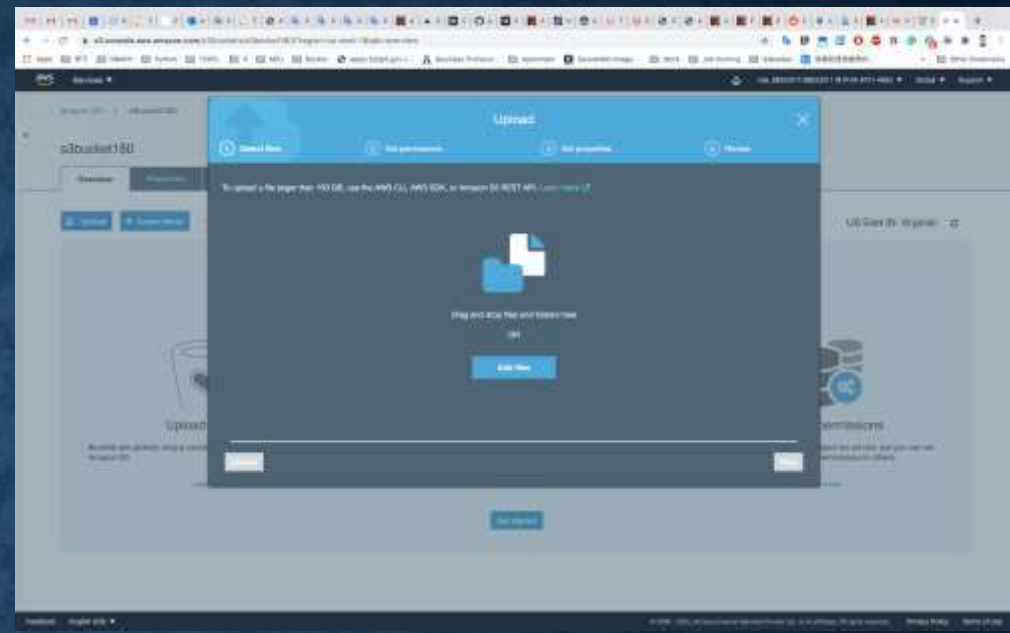
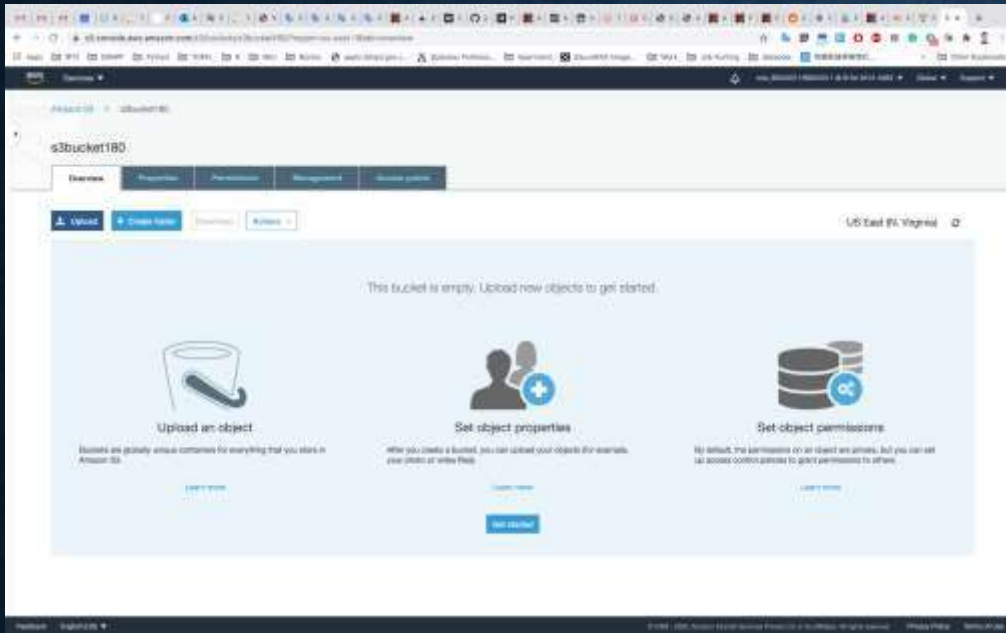
Step 11: Click on Everyone under the Public access, and click on Read object on the right of pop-up window. Then click on Save.

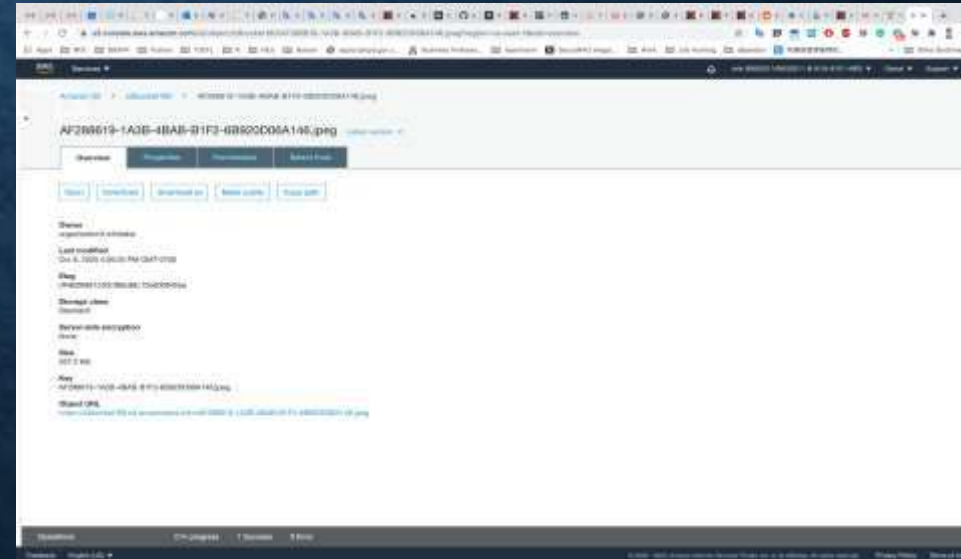
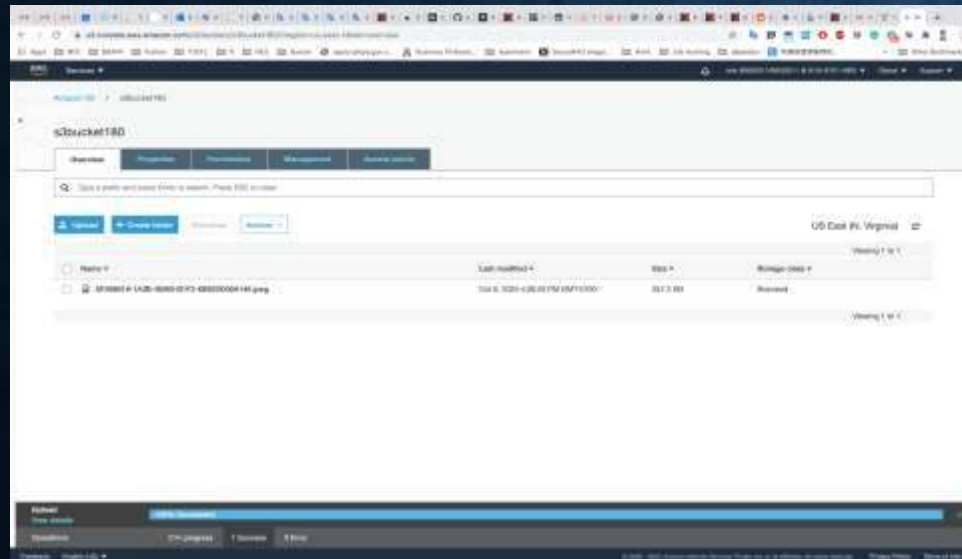
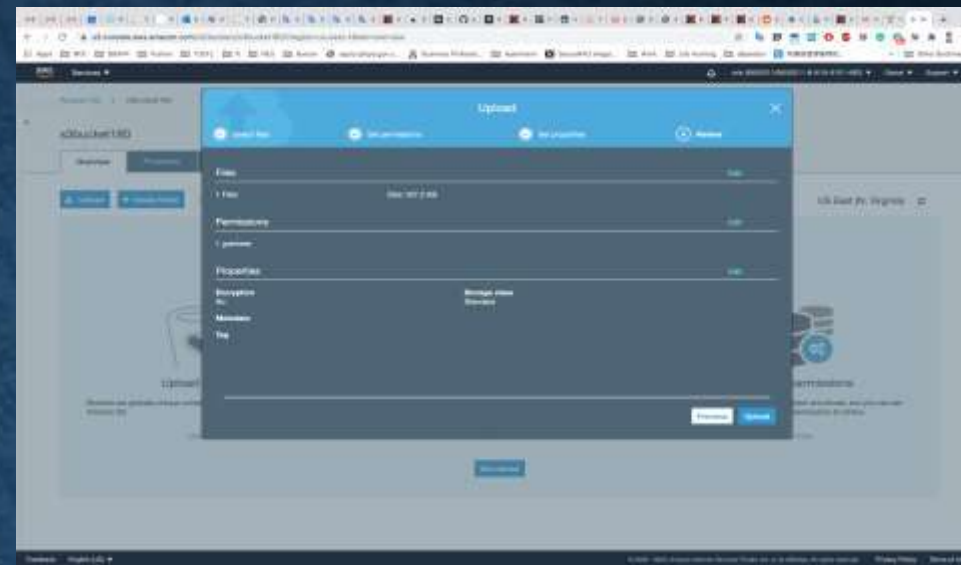
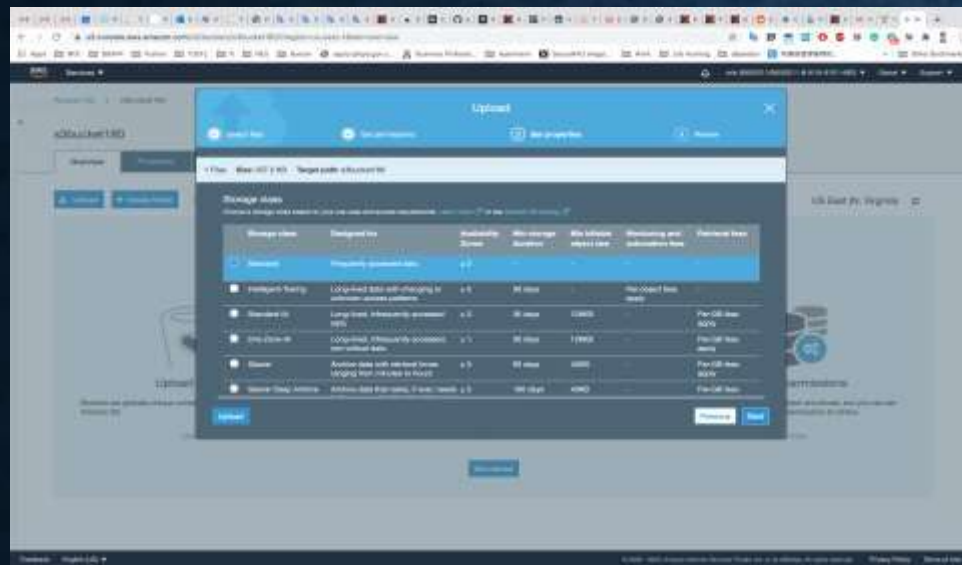
Step 12 : Now its state switches to Read Object - Yes

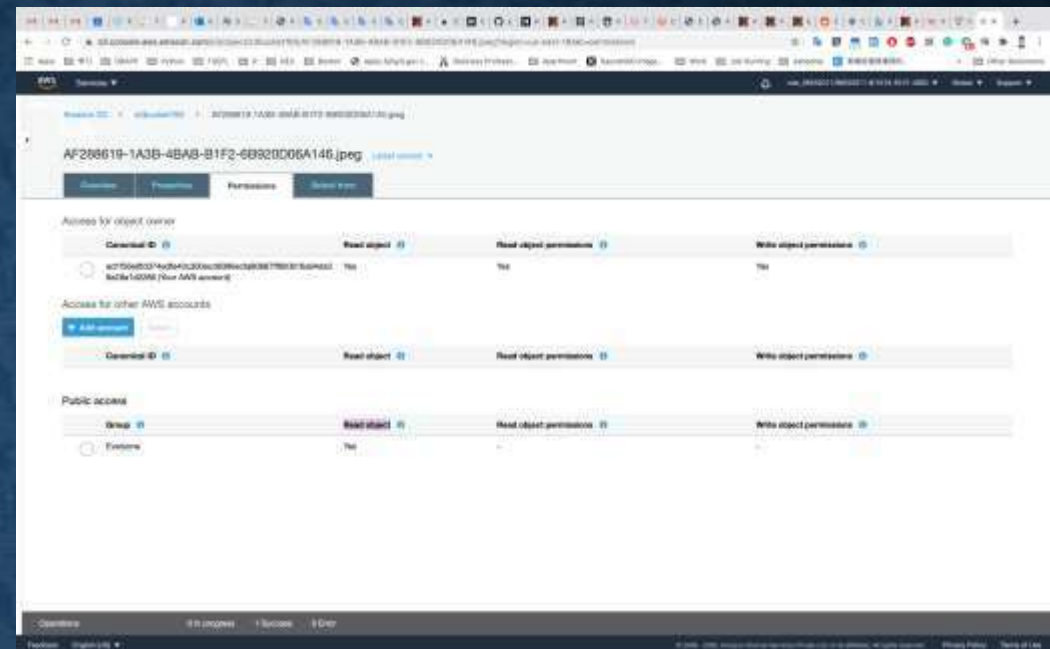
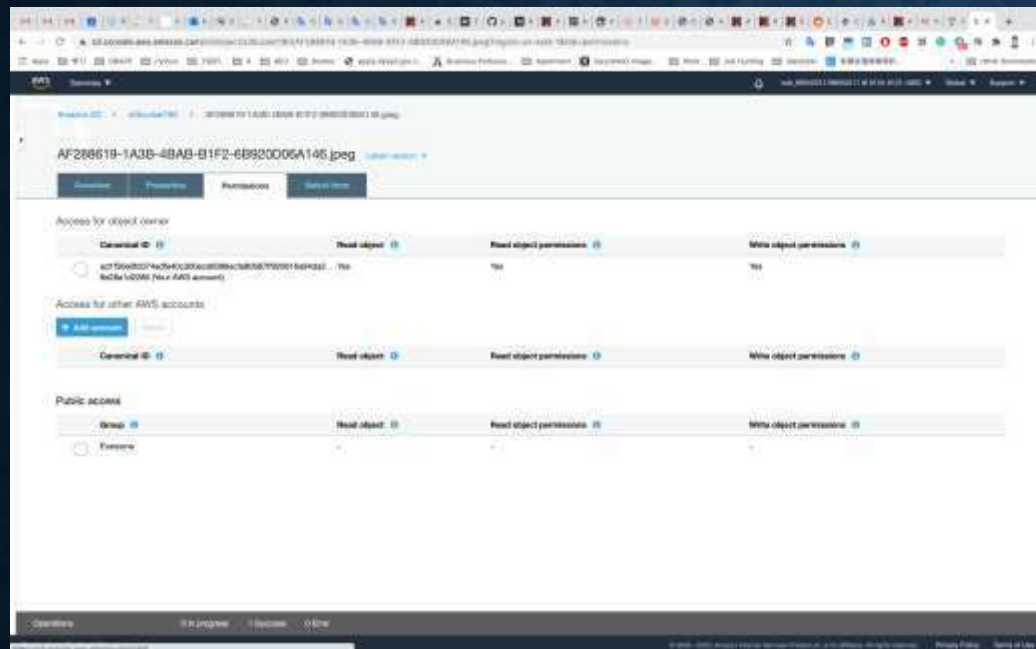
Step 13: Click on Overview, and click on your Object URL again .

Step 14: Notice the URL on your browser









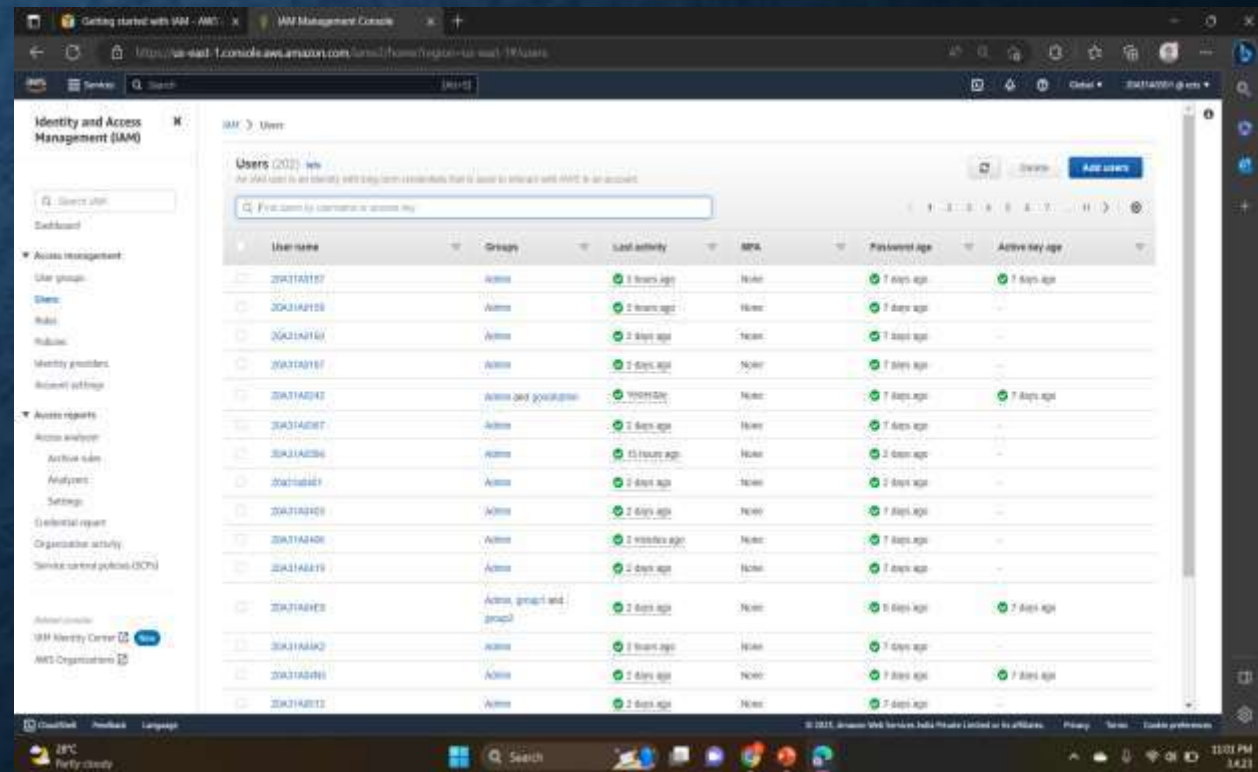
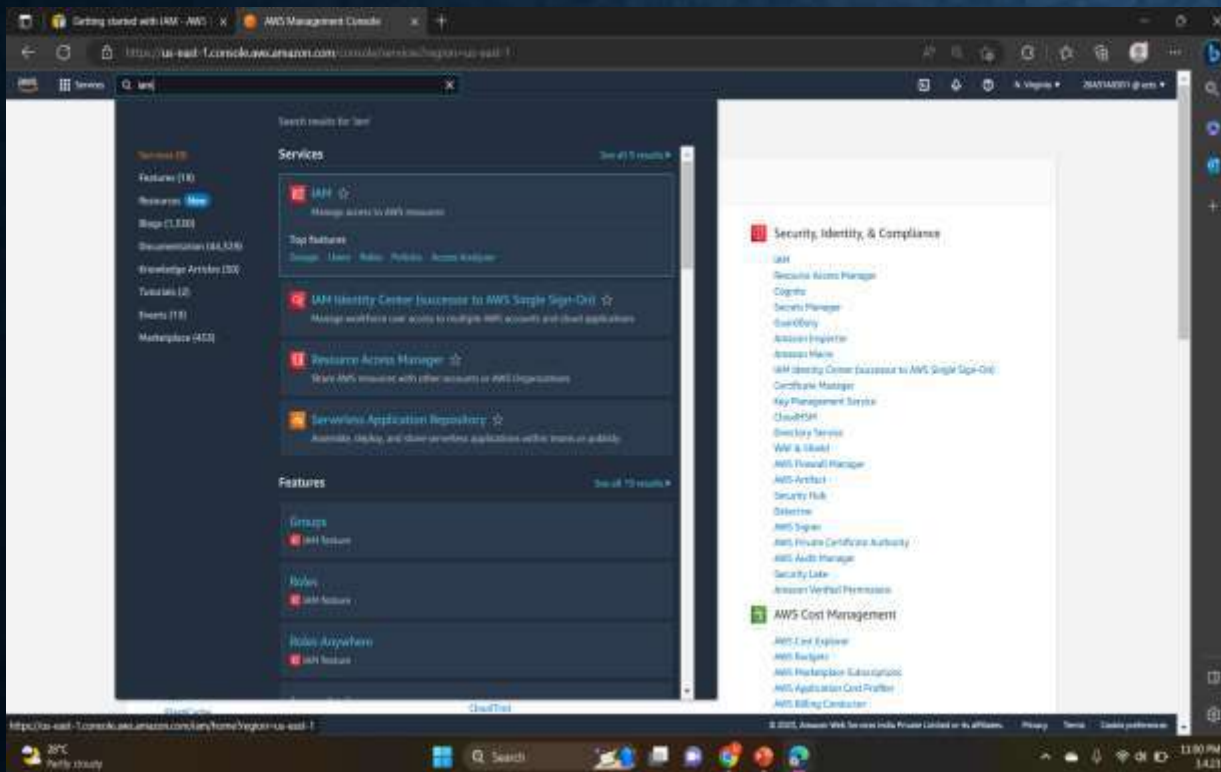
AWS Identity and Access Management (IAM)

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

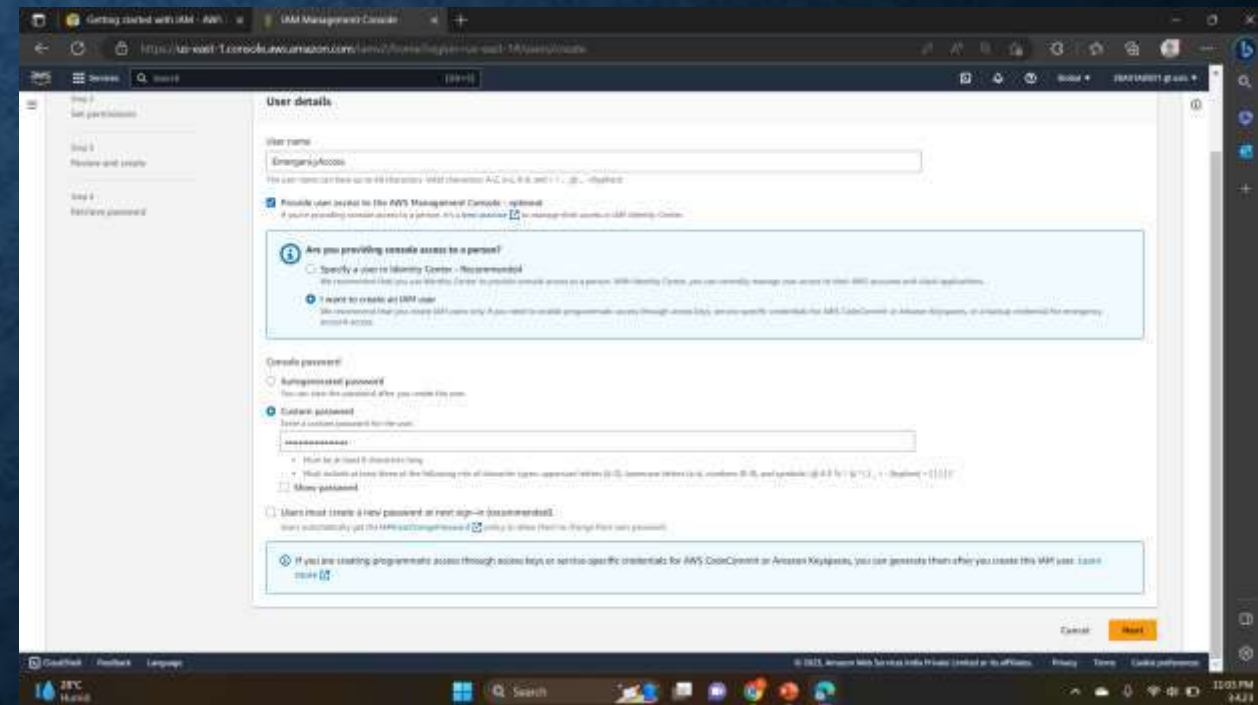
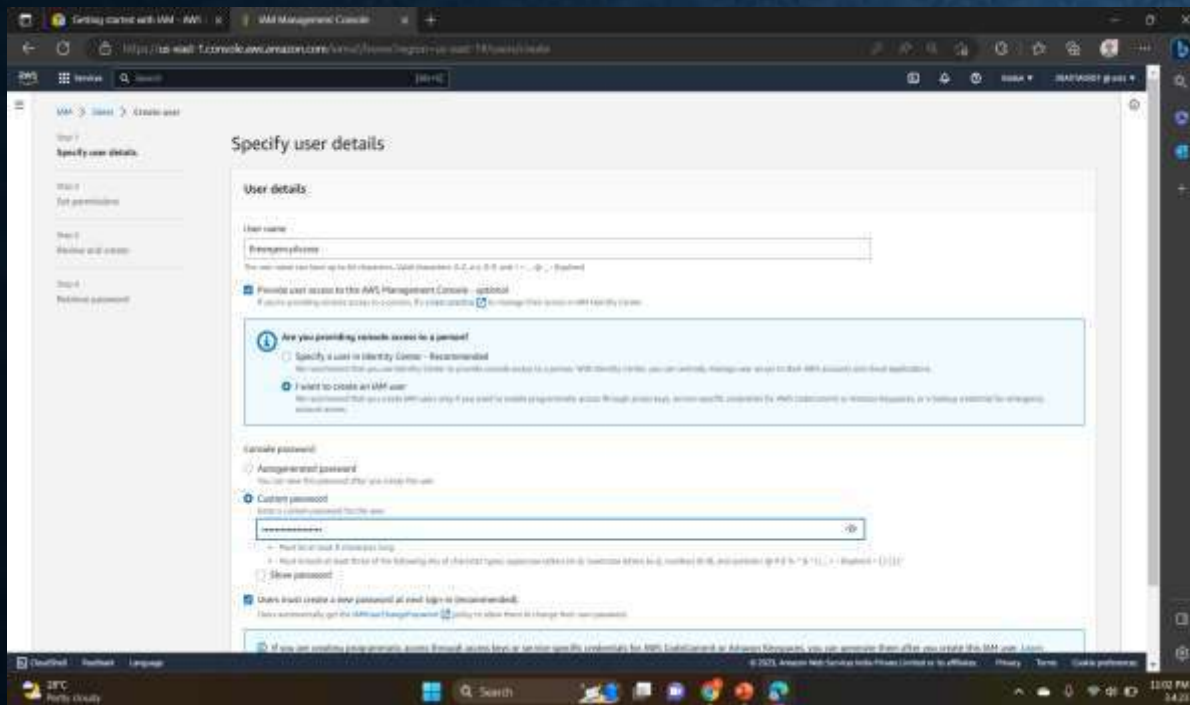
1. On the **Console Home** page, select the IAM service. 2. In the navigation pane, select **Users** and then select **Add users**.



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.**

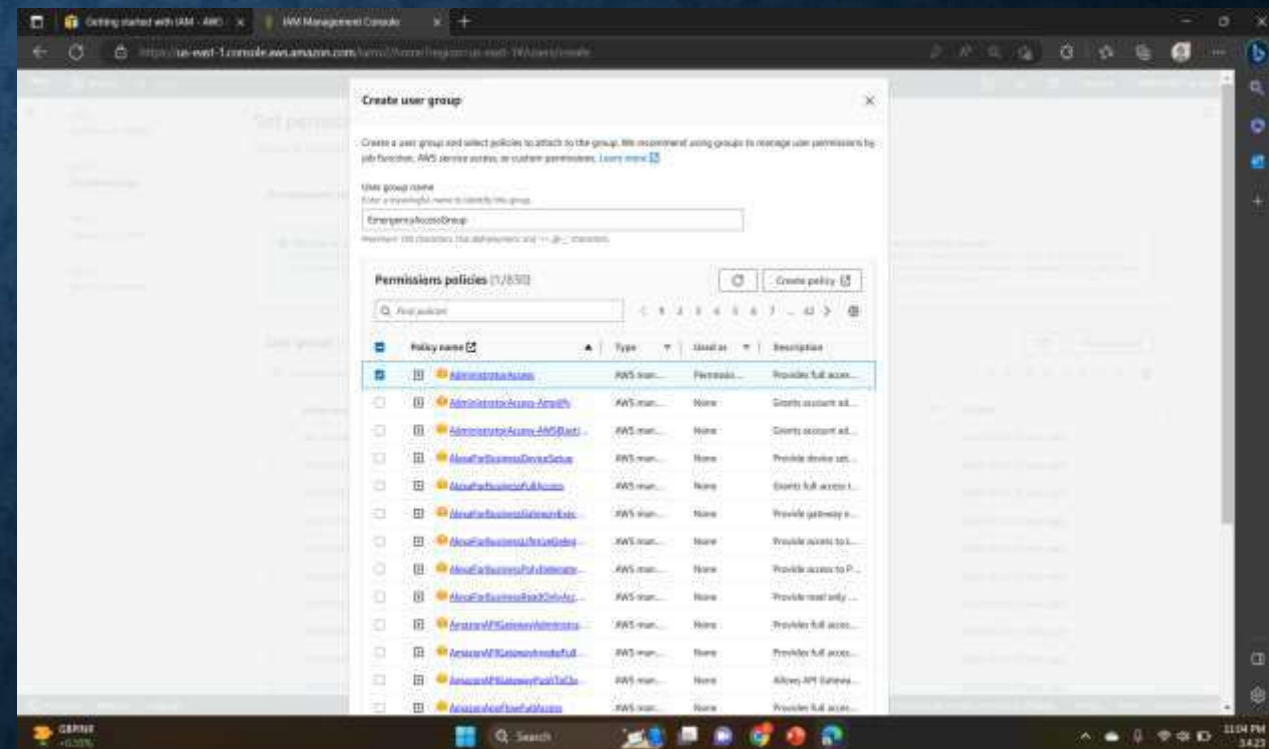
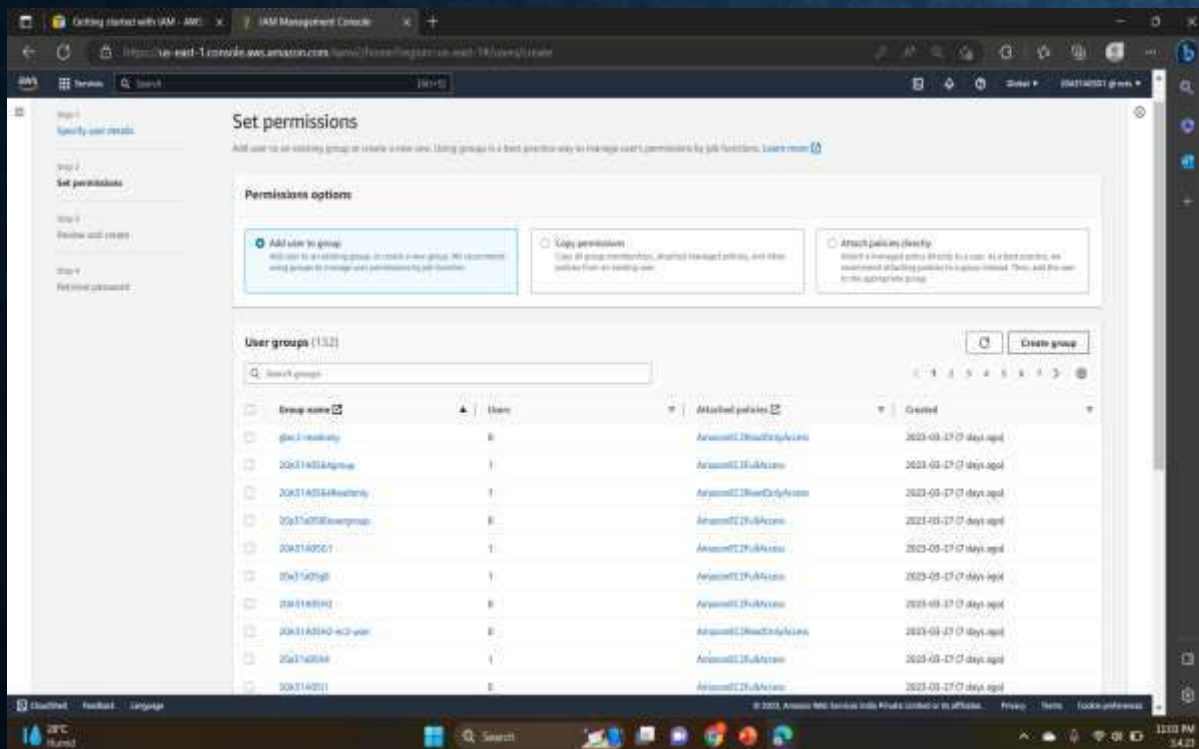
4. Under **Console password**, select **Custom Password** and create your own password.

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



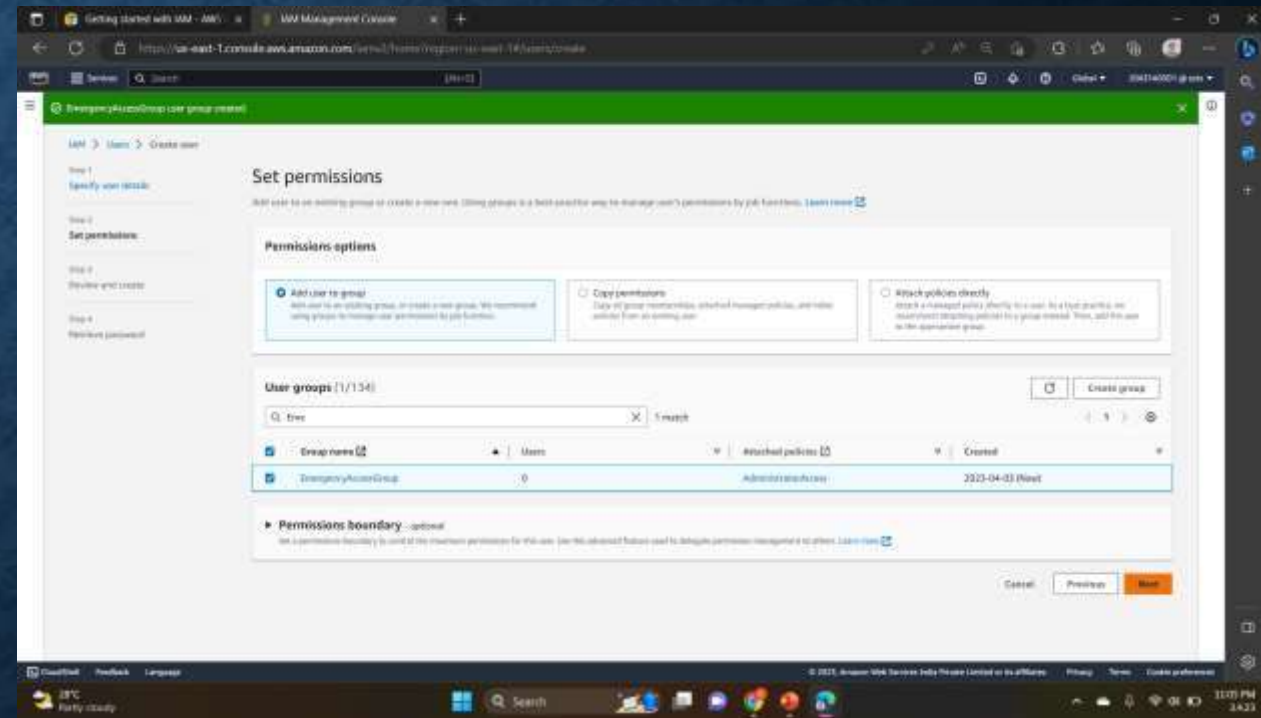
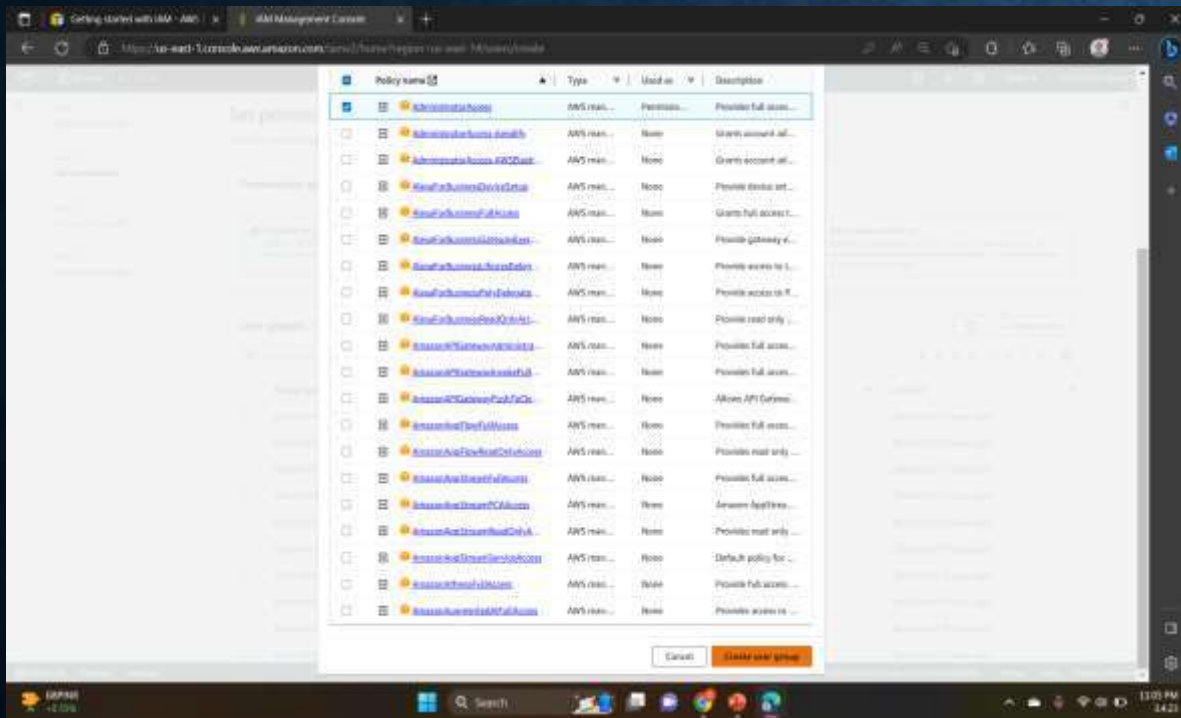
5. On the **Set permissions** page, under **Permissions options**, select **Add user to group**. Then, under **User groups**, select **Create group**.

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

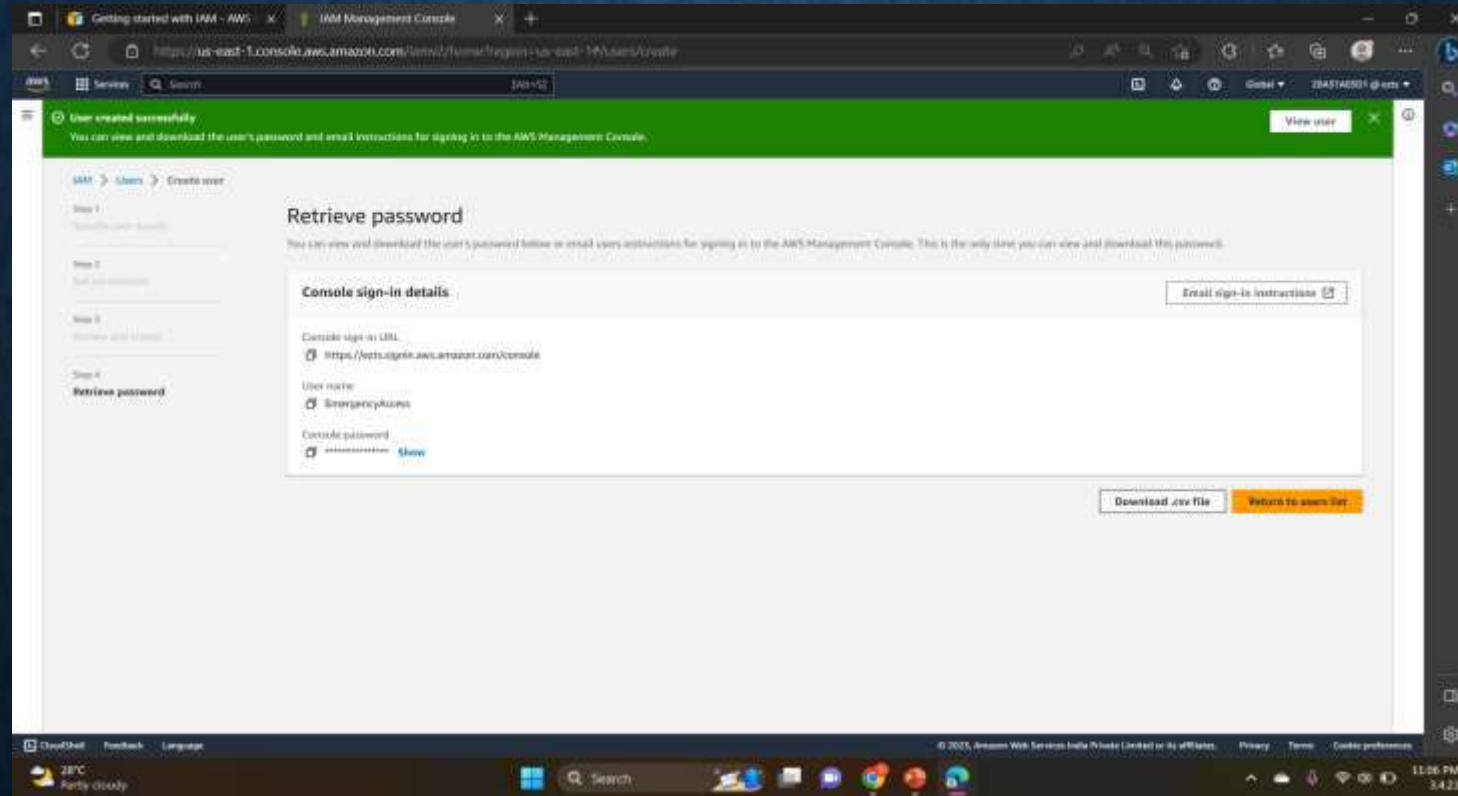


7. Select **Create user group** to return to the **Set permissions** page.

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

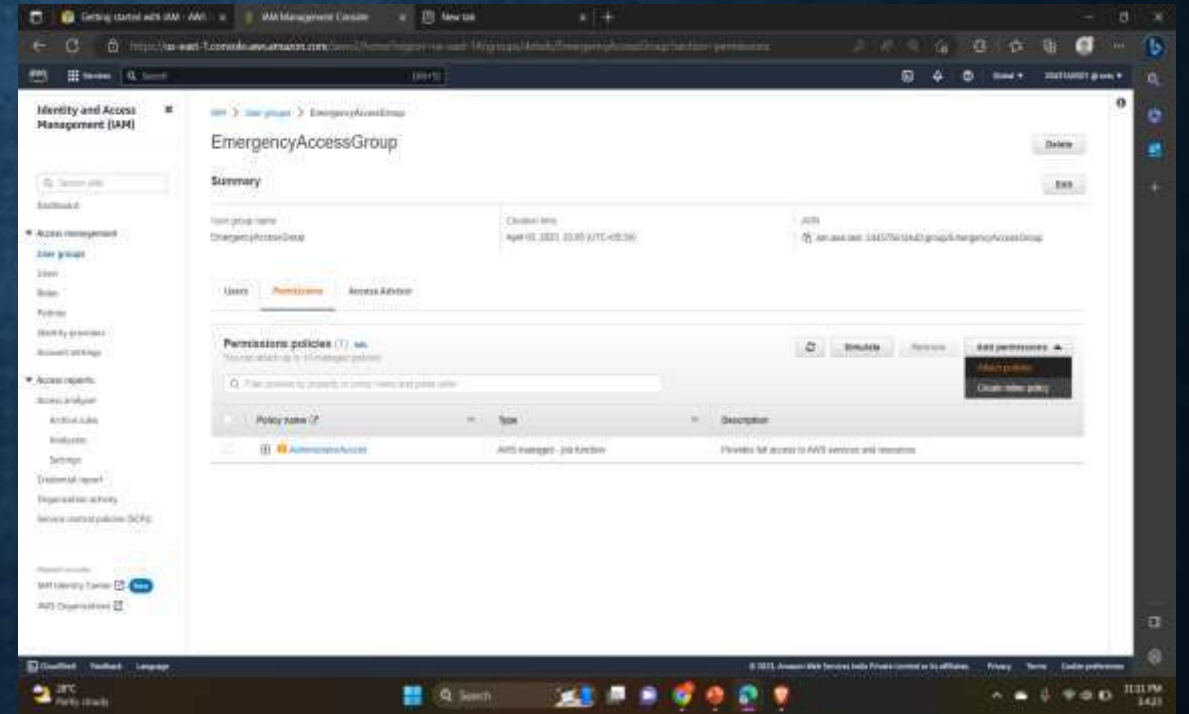
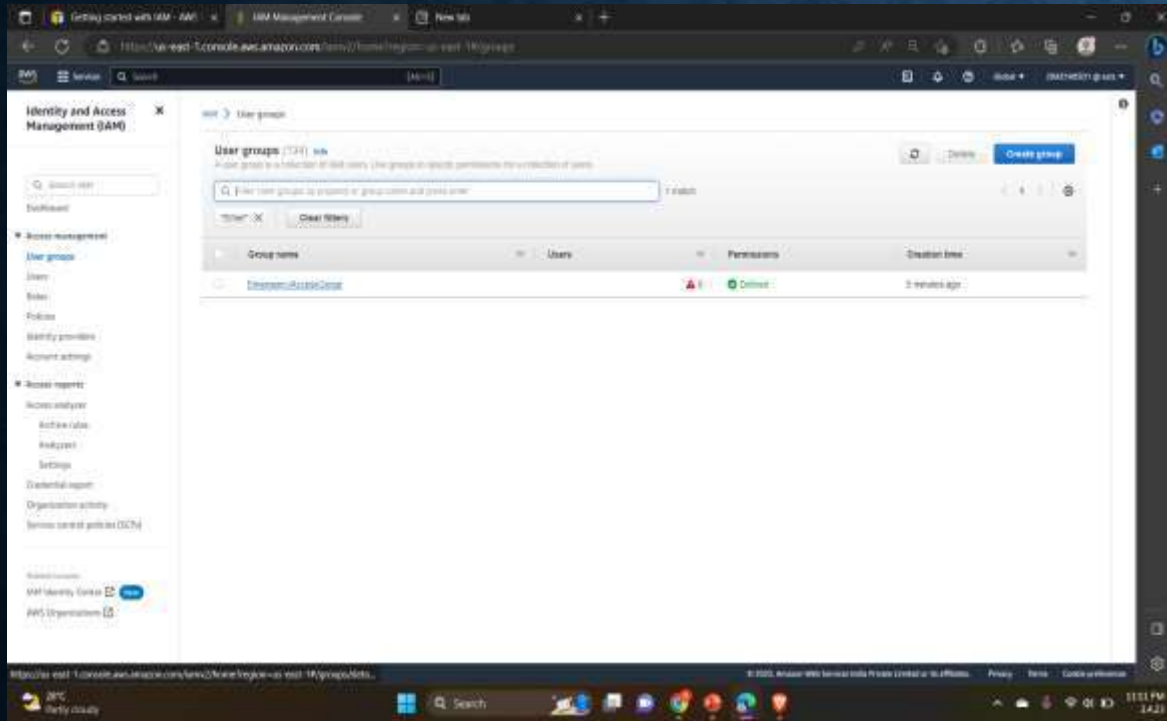


10. On the **Review and create** page, review the list of user group memberships to be added to the new user. When you are ready to proceed, select **Create user**.
11. On the **Retrieve password** page, select **Download .csv file** to save a .csv file with the user credential information (Connection 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.

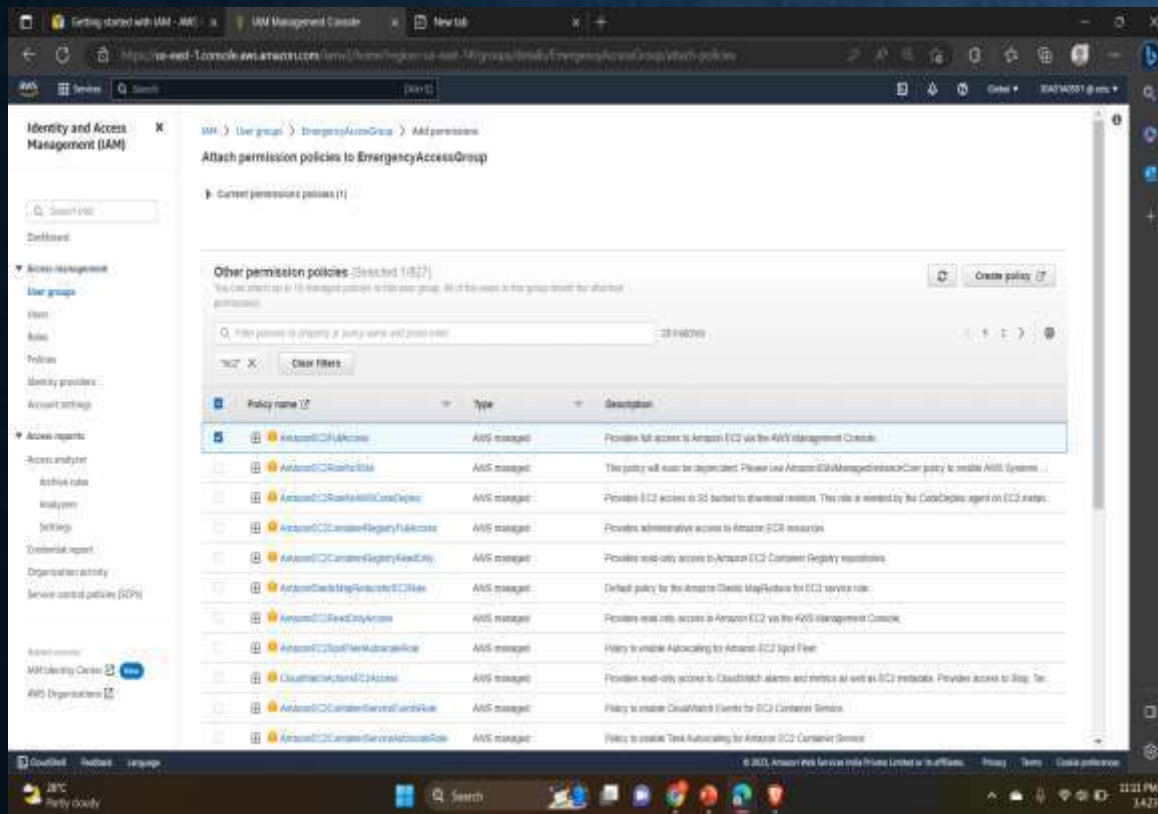


13. If you want to add permissions to the user group, then go to **User groups** and click on the respective **User group**.

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

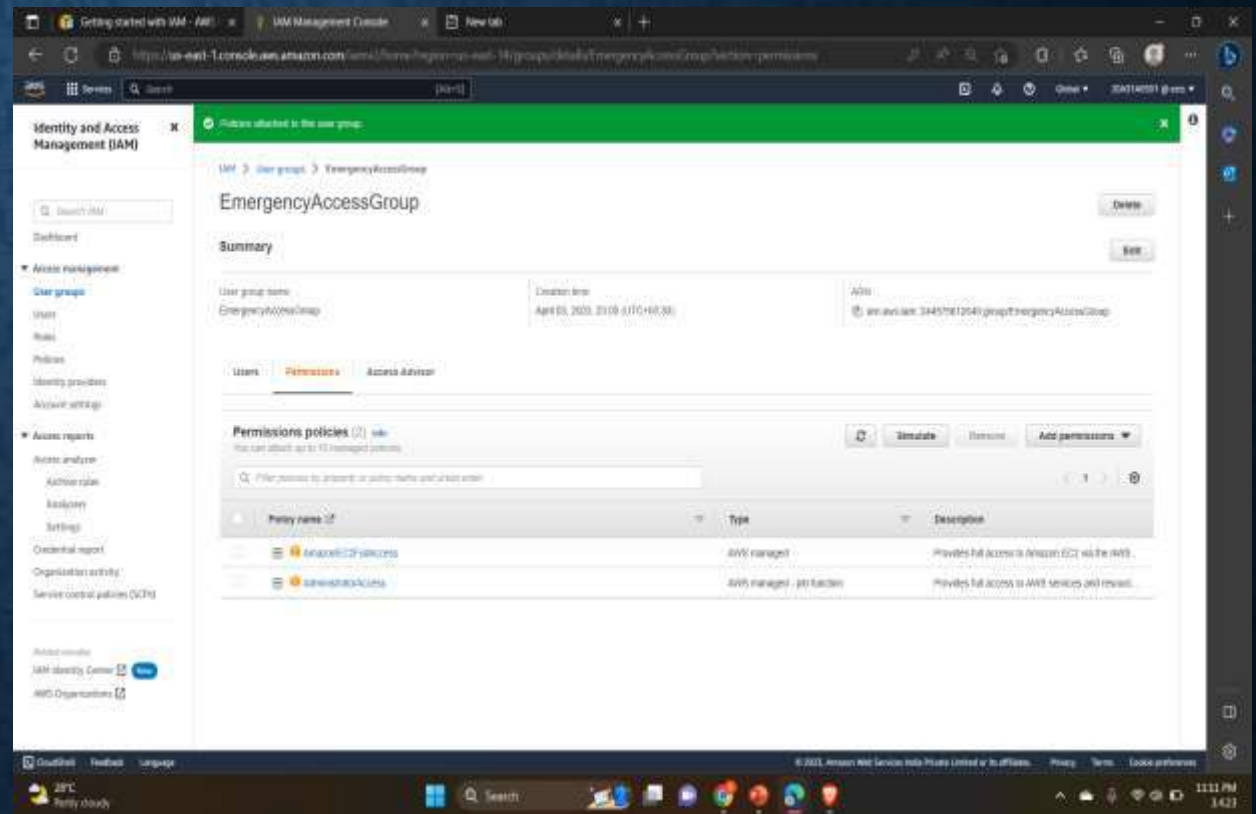


15. Add the permission policy and the policy is attached to the **User group**.



The screenshot shows the AWS IAM console interface. The left sidebar contains navigation links for Identity and Access Management (IAM), Access Management, Access reports, and IAM identity Center. The main content area is titled 'Attach permission policies to EmergencyAccessGroup'. It shows a list of 'Other permission policies' that can be attached to the user group. The policies are listed in a table with columns for Policy name, Type, and Description.

Policy name	Type	Description
AmazonEC2FullAccess	AWS managed	Provides full access to Amazon EC2 via the AWS Management Console.
AmazonEC2ReadOnlyAccess	AWS managed	This policy will soon be deprecated. Please use AmazonEC2ReadOnlyAccess policy to enable AWS Systems Manager to connect to your Amazon EC2 instances.
AmazonEC2ReadOnlyAccess	AWS managed	Provides EC2 access to EC2 instances to download content. This role is needed by the CodeDeploy agent on EC2 instances.
AmazonEC2ContainerRegistryFullAccess	AWS managed	Provides administrative access to Amazon ECR resources.
AmazonEC2ContainerRegistryReadOnly	AWS managed	Provides read-only access to Amazon ECR Container Registry resources.
AmazonElasticMapReduceEC2Role	AWS managed	Default policy for the Amazon Elastic MapReduce for EC2 service role.
AmazonEC2ReadOnlyAccess	AWS managed	Provides read-only access to Amazon EC2 via the AWS Management Console.
AmazonEC2SpotFleetManagedRole	AWS managed	Policy to enable AutoScaling for Amazon EC2 Spot Fleet.
AmazonEC2SpotFleetManagedRole	AWS managed	Provides read-only access to CloudWatch alarms and metrics as well as EC2 metadata. Provides access to Stop Terminate and Reboot actions for Amazon EC2 Spot Fleet.
AmazonEC2SpotFleetManagedRole	AWS managed	Policy to enable CloudWatch Events for EC2 Container Service.
AmazonEC2SpotFleetManagedRole	AWS managed	Policy to enable Task Automation for Amazon EC2 Container Service.



The screenshot shows the AWS IAM console interface. The left sidebar contains navigation links for Identity and Access Management (IAM), Access Management, Access reports, and IAM identity Center. The main content area is titled 'EmergencyAccessGroup'. It shows a summary of the group and a list of attached permission policies. The policies are listed in a table with columns for Policy name, Type, and Description.

Policy name	Type	Description
AmazonEC2FullAccess	AWS managed	Provides full access to Amazon EC2 via the AWS Management Console.
AmazonEC2ReadOnlyAccess	AWS managed - job function	Provides full access to AWS services and resources.