

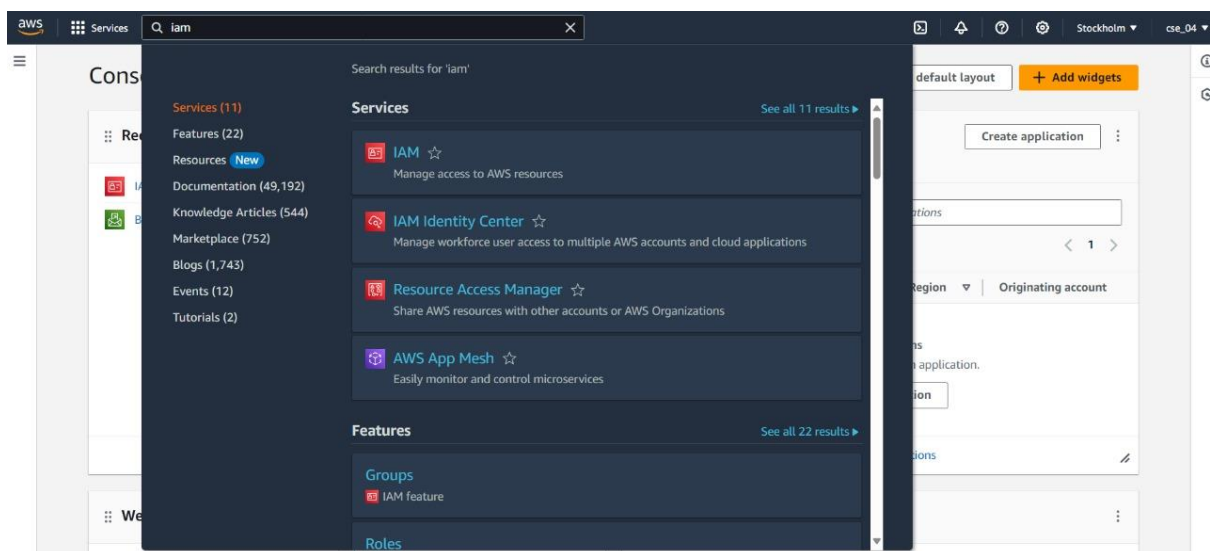
Assignment 3:

Problem Statement: Create IAM user and give full access to S3.

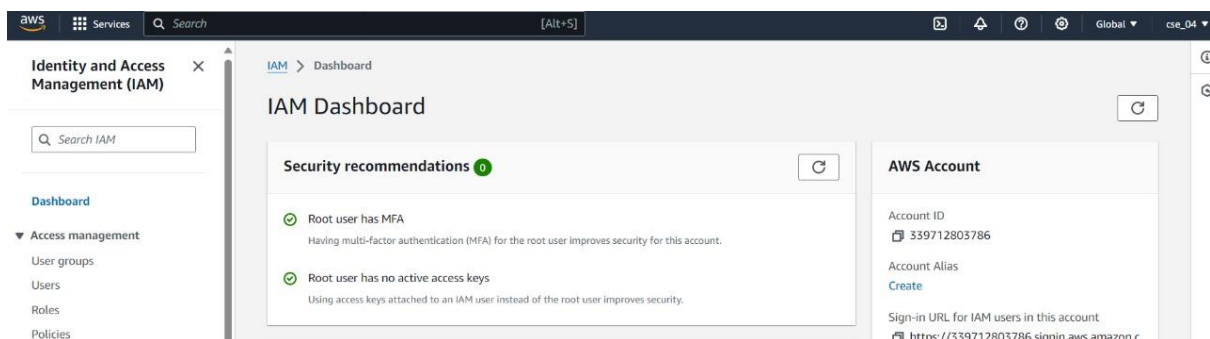
Steps:

Still now we were using Root user which has full control. It is like project manager and IAM (Identity and Access Management) users are like team members. They have to assigned with one or more than one particular tasks. If IAM is given full access with S3(Simple Storage Services) then it can't access EC2(Elastic Compute Cloud) or RDS (Relational Database Service). So the steps of this assignment are -

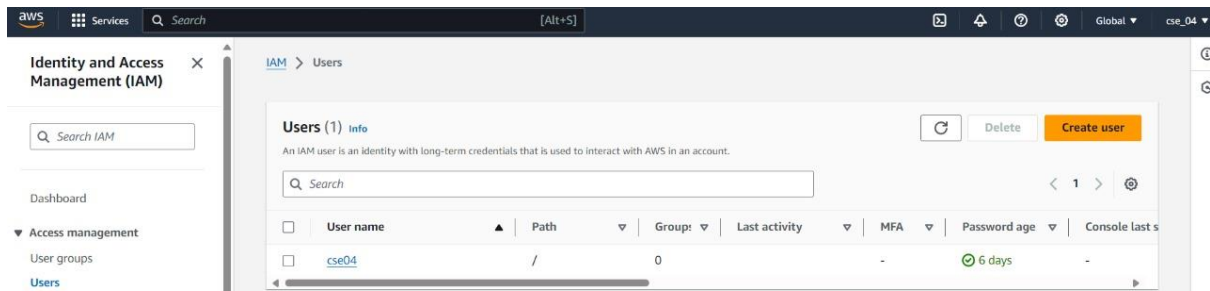
a) At first search **IAM** and click on **IAM** option.



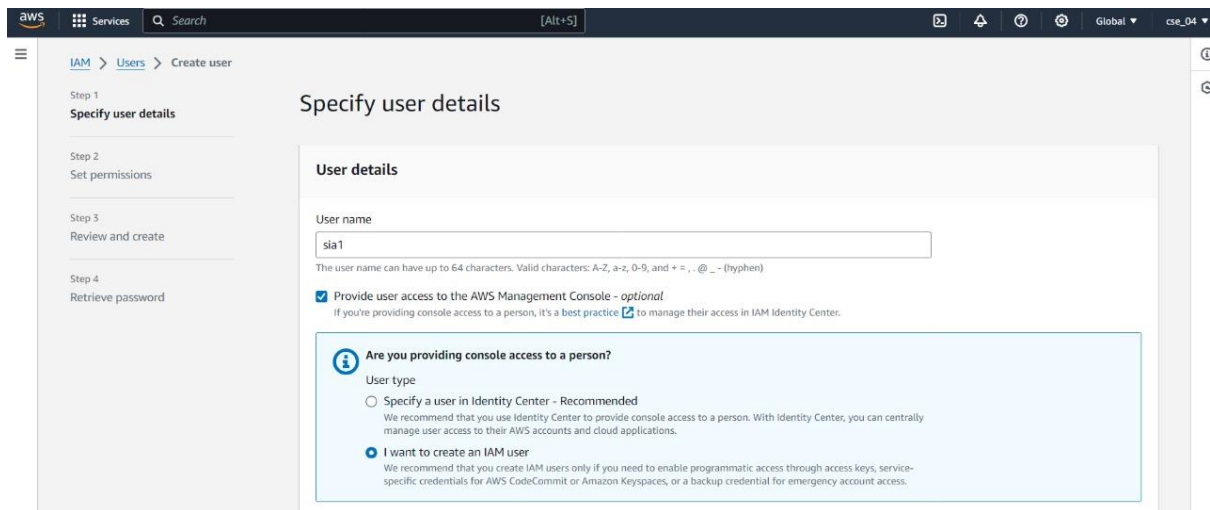
b) Now go to **Access management** and click on **Users**.



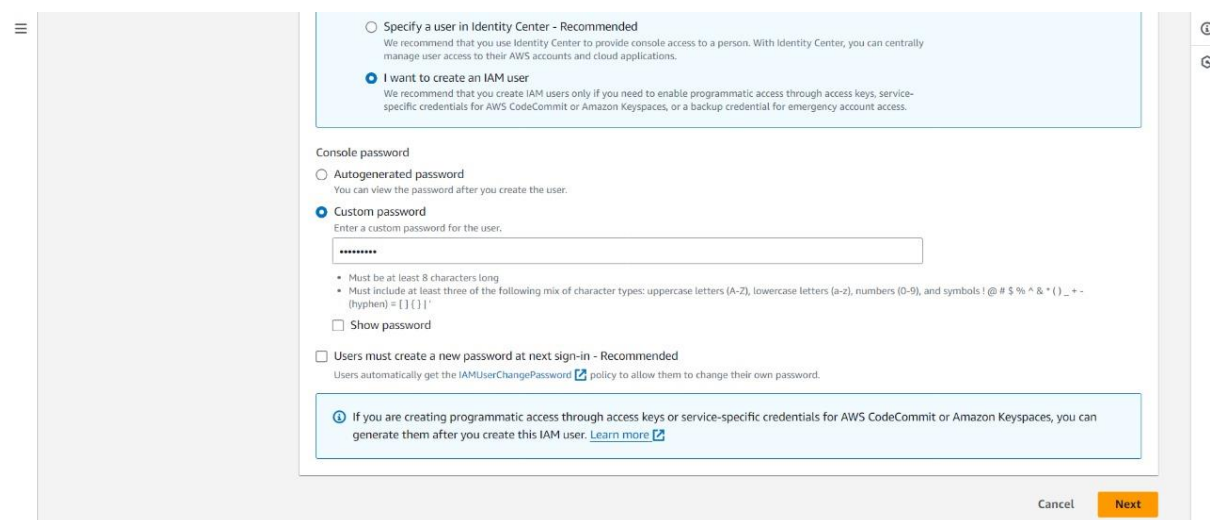
c) Now click on **Create user**.



d) Now give **username** and click on check box stating ‘**Provide user access to the AWS Management Console – optional**’. After that click on **I want to create an IAM user**.



e) Now click on **Custom password** and give password following the rules mentioned bellow and now uncheck the option stating **Users must create a new password at next sign-in – Recommended**. Now click on **next** option.



f) Now click on **Create group**.

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

- ☒ **Add user to group**
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- ☐ **Copy permissions**
Copy all group memberships, attached managed policies, and inline policies from an existing user.
- ☐ **Attach policies directly**
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Get started with groups
Create a group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. [Learn more](#)

[Create group](#)

Set permissions boundary - optional

[Cancel](#) [Previous](#) [Next](#)

g) Now give **username** for user group. After this in **Permissions policies** search **s3** and search **AmazonS3FullAccess** in Policy name not in Description. Click on searched checkbox and now click on **Create user group**.

Create user group

Create a user group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. [Learn more](#)

User group name
Enter a meaningful name to identify this group.
group1
Maximum 128 characters. Use alphanumeric and '+', '@', '-' characters.

Permissions policies (1/912)

Filter by Type: All types 9 matches

Policy name	Type	Use...	Description
<input type="checkbox"/> AmazonDMSRedsh...	AWS managed	None	Provides access to manage S3 sett
<input checked="" type="checkbox"/> AmazonS3FullAccess	AWS managed	None	Provides full access to all buckets v
<input type="checkbox"/> AmazonS3ObjectL...	AWS managed	None	Provides AWS Lambda functions pe
<input type="checkbox"/> AmazonS3Outpost...	AWS managed	None	Provides full access to Amazon S3
<input type="checkbox"/> AmazonS3Outpost...	AWS managed	None	Provides read only access to Amaz

[Cancel](#) [Create user group](#)

h) After it our first user group will be created. Now click on **Group name's checkbox** and then click on **Next**.

The screenshot shows the 'Set permissions' step in the AWS IAM console. The left sidebar indicates the progress: Step 2 (Set permissions), Step 3 (Review and create), and Step 4 (Retrieve password). The main content area is titled 'Permissions options' and contains three radio button options: 'Add user to group' (selected), 'Copy permissions', and 'Attach policies directly'. Below these options is a table titled 'User groups (1/1)' with a search bar and a 'Create group' button. The table has columns for 'Group name', 'Users', 'Attached policies', and 'Created'. One group, 'group1', is listed with 0 users and the 'AmazonS3FullAccess' policy, created on 2024-02-11. A link to 'Set permissions boundary - optional' is also visible. At the bottom right are 'Cancel', 'Previous', and 'Next' buttons.

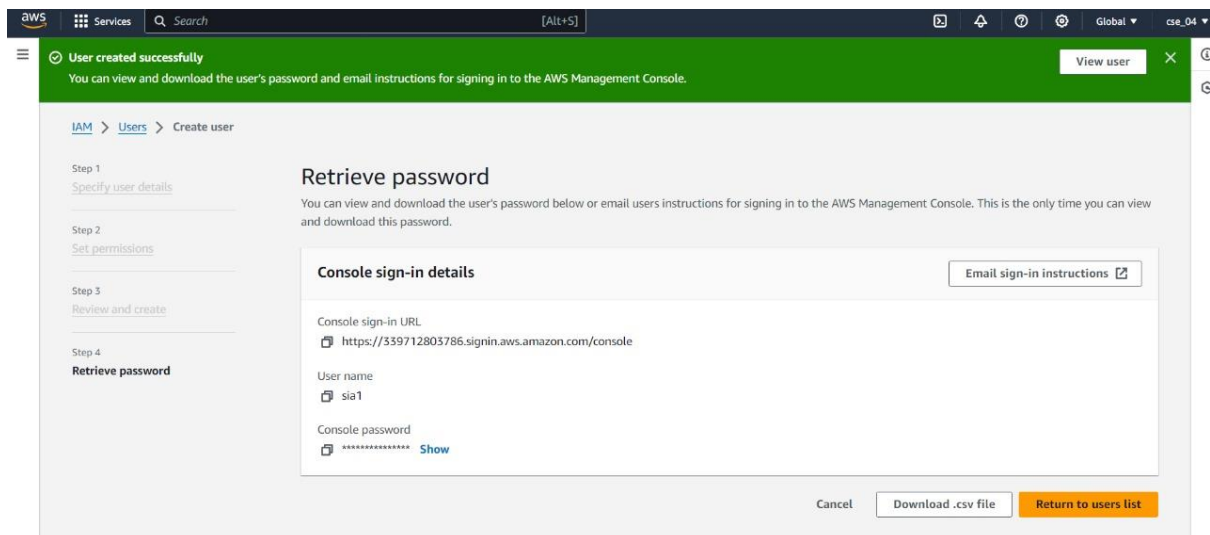
Group name	Users	Attached policies	Created
group1	0	AmazonS3FullAccess	2024-02-11 (Now)

i) Now again click on **Create user**.

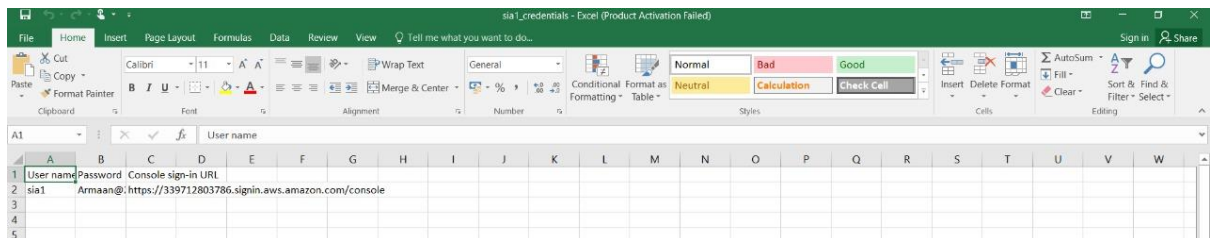
The screenshot shows the 'Review and create' step in the AWS IAM console. The left sidebar indicates the progress: Step 3 (Review and create) and Step 4 (Retrieve password). The main content area is titled 'User details' and contains three sections: 'User name' (sia1), 'Console password type' (Custom password), and 'Require password reset' (No). Below these is a 'Permissions summary' table with columns for 'Name', 'Type', and 'Used as'. One entry, 'group1', is listed as a 'Group' used as a 'Permissions group'. At the bottom is a 'Tags - optional' section with an 'Add new tag' button. At the bottom right are 'Cancel', 'Previous', and 'Create user' buttons.

Name	Type	Used as
group1	Group	Permissions group

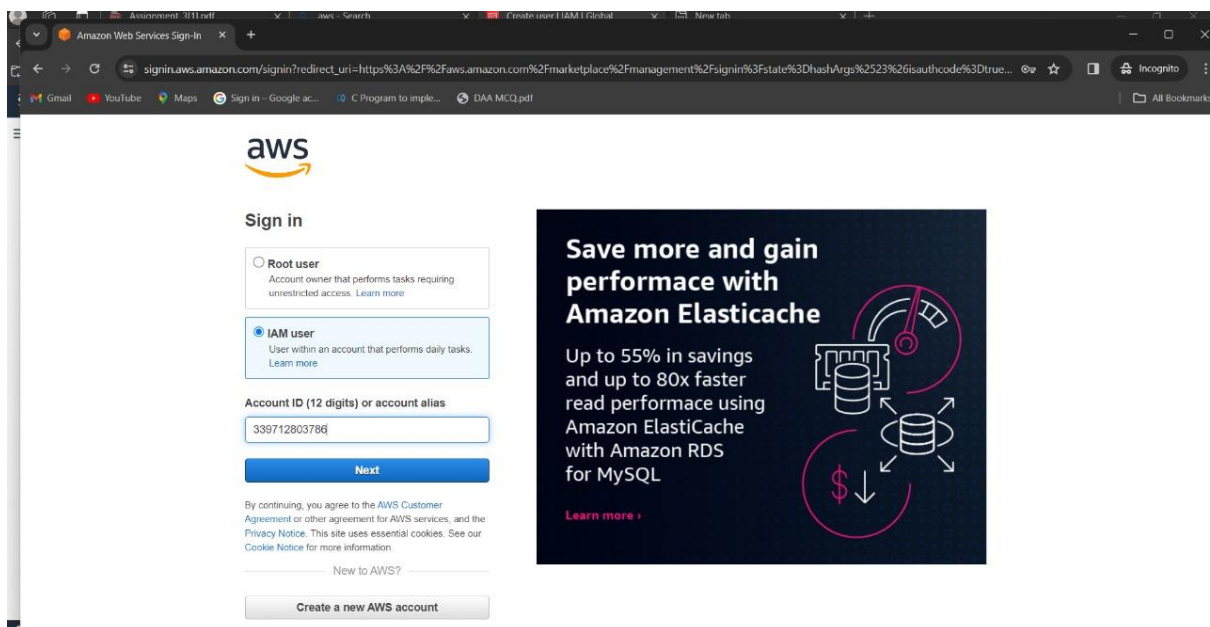
j) Now user will be created successfully. Now **download .csv file** and click on **Return on user list**.



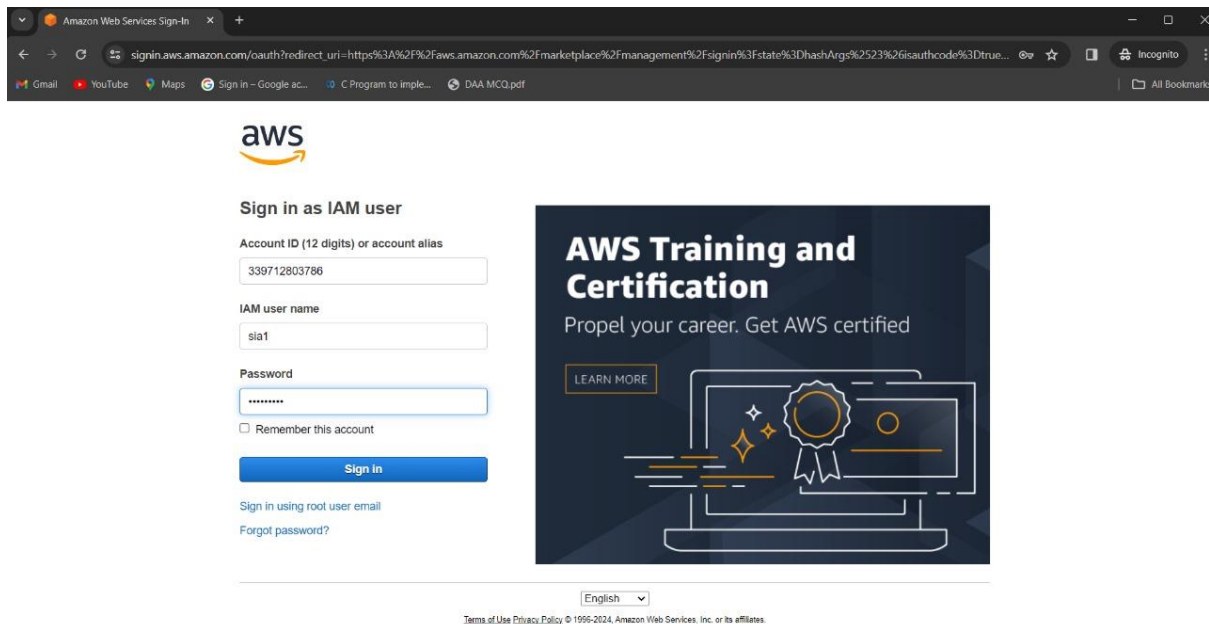
This is .csv file content:



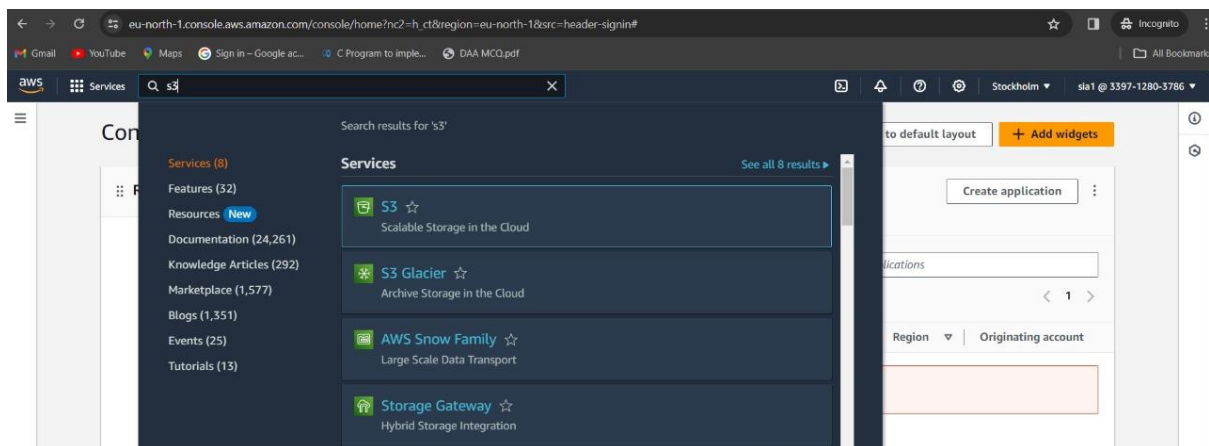
k) Now go to **incognito mode** and search **amazon console login**. Click on **IAM user** and give **12 digit Account ID** from that .csv file. Click on **Next**.



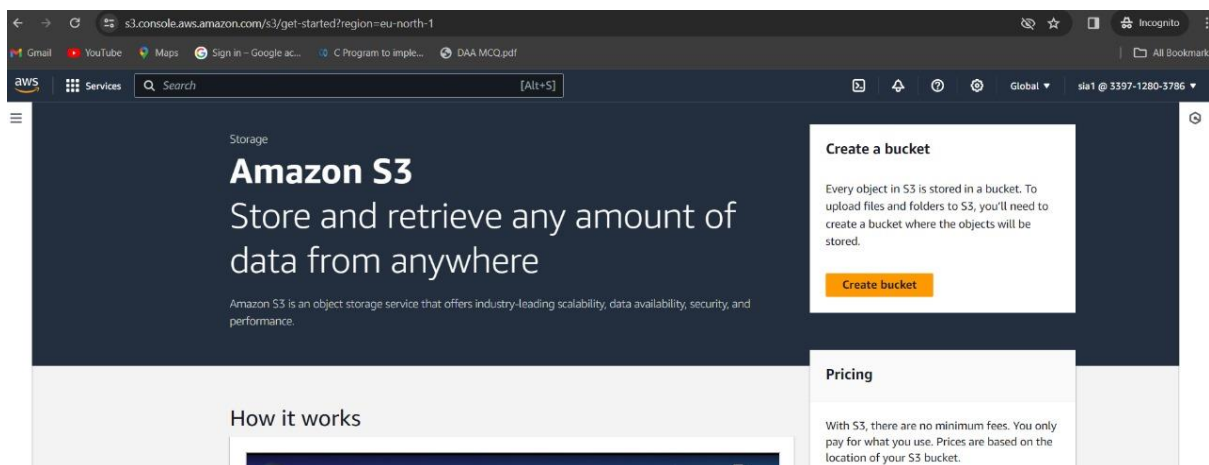
l) Now give **IAM user name** and **Password** from that .csv file. Click on **sign in**.



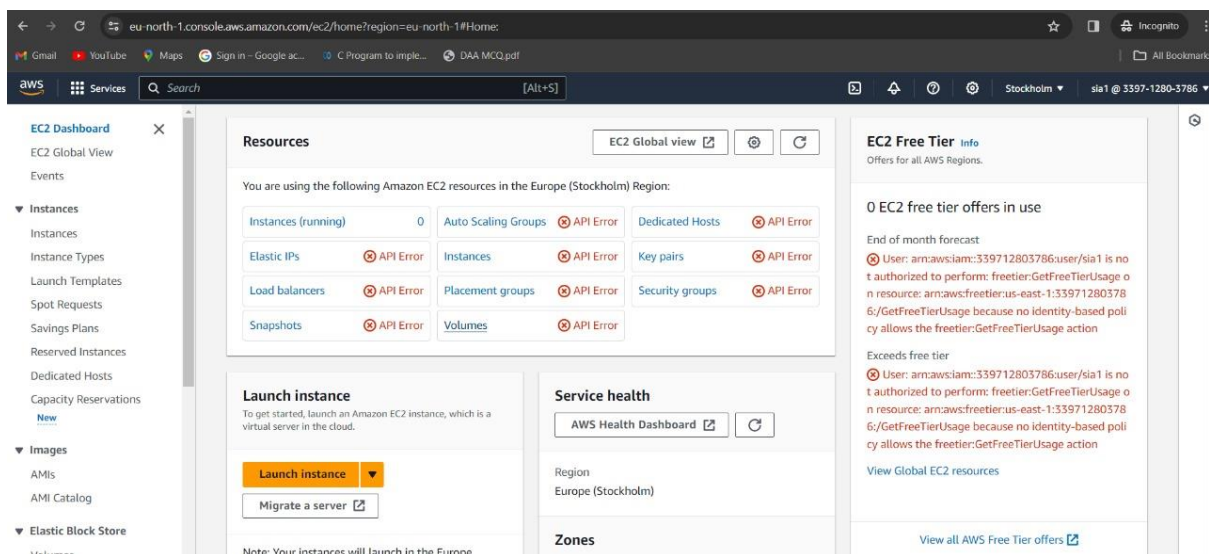
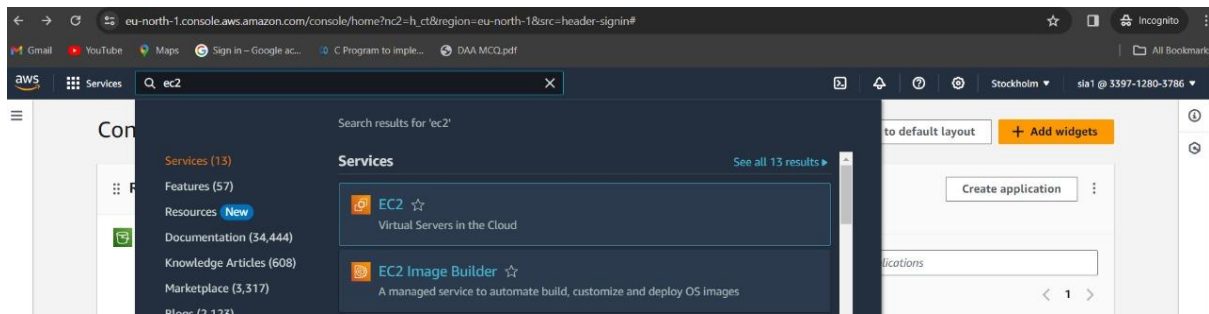
m) Now search **S3** in AWS console and click in **S3**.



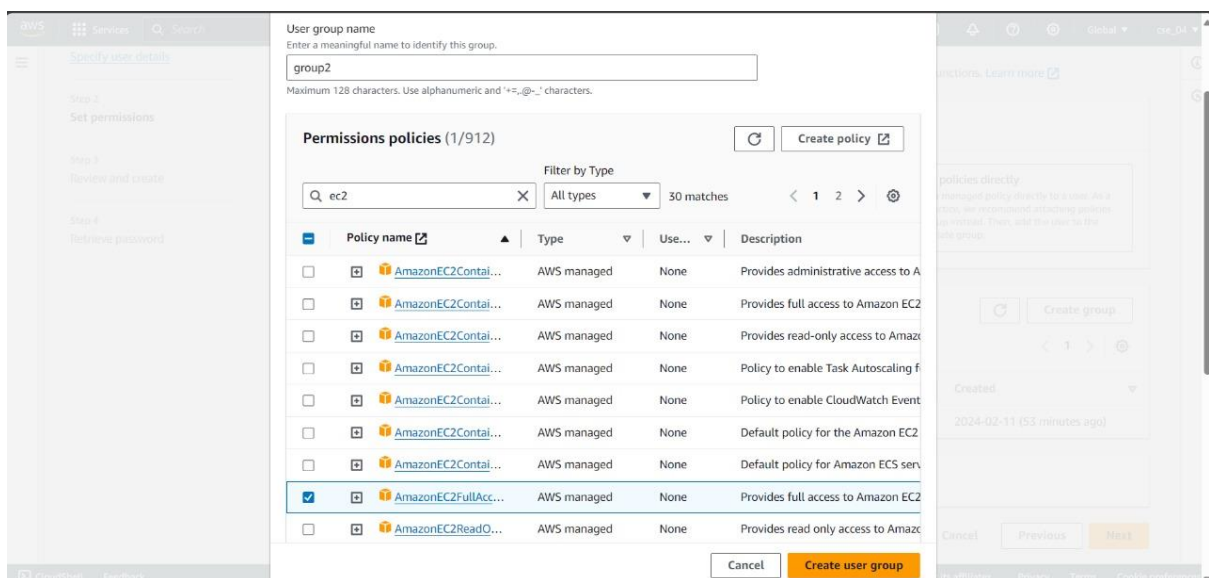
n) S3 window will be opened and there is a option of **Create bucket**. In this bucket we can apply static website, file, folder.



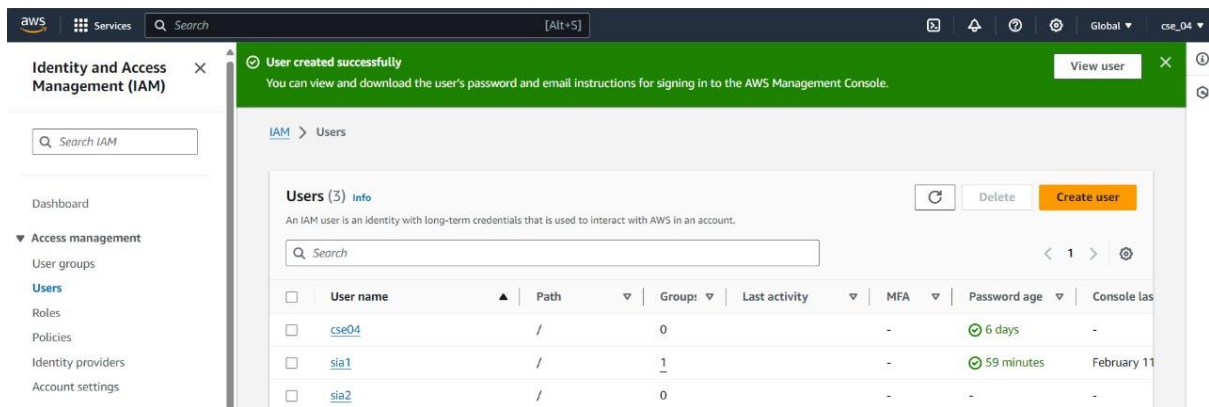
o) Now back to console and search **EC2** like S3 and press on **EC2**. You can see API errors are occurring as only s3 access was given to this user not EC2.



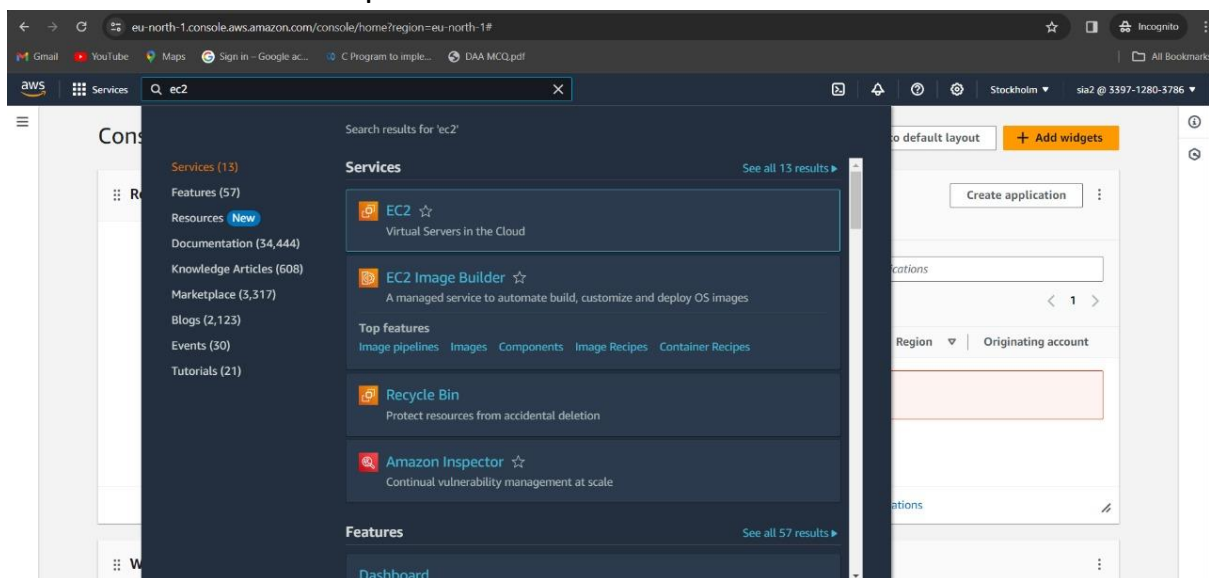
p) Now do sign out and come out from incognito mode. Now make another user. Follow same steps and also make a new user group. And now in Permission policies search **EC2** in search bar and click on **AmazonEC2FullAccess** and press on **Create user group**.



q) In same way **download .csv file** and return to **users list**. Now two separate IAM users will be created and both assigned with different access one is for s3 and another is for EC2.



r) Now in same way return back to incognito mode and sign in to AWS console and select IAM root and give 12 digit id , username and password by copying those from .csv file like previous. Now search ec2 and click on EC2.



s) Now when we go to **EC2** then we can see no API Error is occurring like previous as this user has given access of EC2.

The screenshot displays the AWS Management Console interface for the EC2 service. The top navigation bar includes the AWS logo, a 'Services' menu, a search bar, and a keyboard shortcut '[Alt+S]'. The left-hand navigation pane is expanded to show the 'EC2 Dashboard' with a sub-menu containing 'EC2 Global View', 'Events', and a list of EC2-related services under 'Instances' and 'Images'. The main content area is titled 'Resources' and shows a summary of EC2 resources in the 'Europe (Stockholm) Region'. A table lists various resource types and their counts. Below this, there are sections for 'Launch instance' and 'Service health'. The 'Launch instance' section includes a description and a prominent orange 'Launch instance' button. The 'Service health' section provides a link to the 'AWS Health Dashboard' and a 'Region' dropdown menu.

Resources					
You are using the following Amazon EC2 resources in the Europe (Stockholm) Region:					
Instances (running)	0	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	0	Key pairs	0
Load balancers	0	Placement groups	0	Security groups	1
Snapshots	0	Volumes	0		

Launch instance
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Service health
AWS Health Dashboard

Region