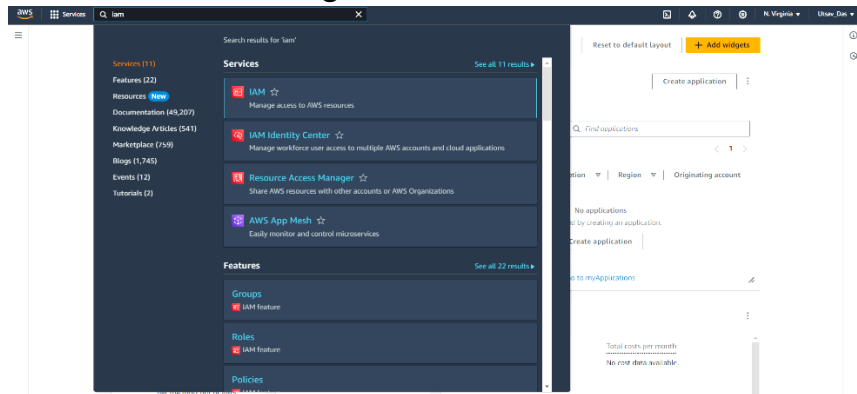


# Assignment – 3

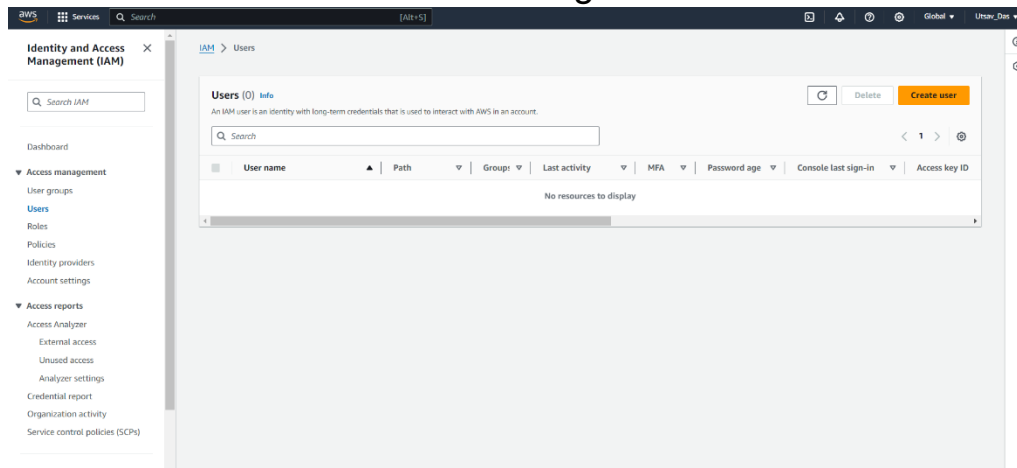
## Problem Statement:

Create IAM user and give full access to S3.

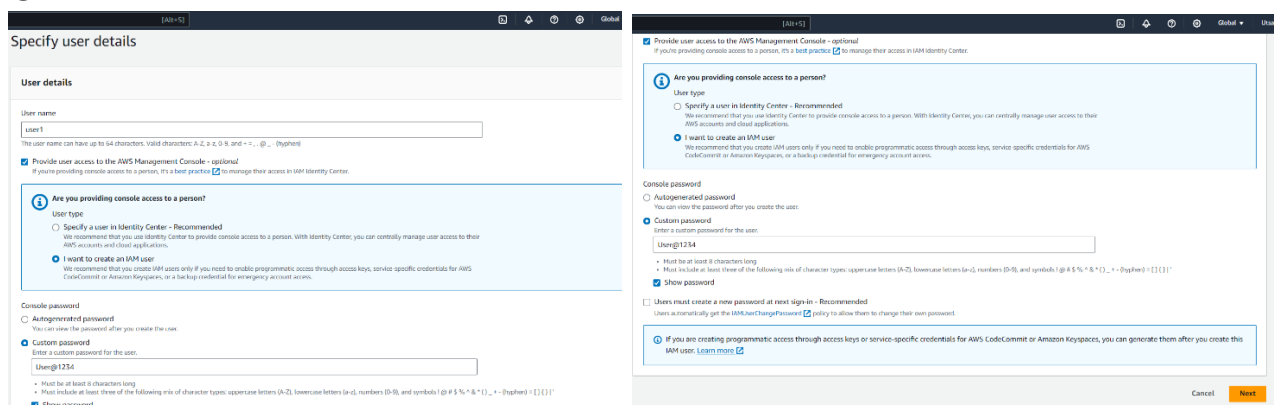
- 1) Go to 'https://console.aws.amazon.com' & sign in to AWS account.
- 2) Search for 'IAM' & go to 'IAM'.



- 3) Go to 'Users' under 'Access management' & click 'Create user'.



- 4) Give user name.  
Select 'I want to create an IAM user'.  
Select 'Custom password'.  
Give a password according to the constraints.  
Uncheck 'Users must create a new password at next sign-in'.  
Click 'Next'.



- 5) Select 'Add user to group'.
- Click '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**

- 6) Give 'User group name'.
- Search for S3 permission.
- Check the row with policy name 'AmazonS3FullAccess'.
- Click '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.  
s3fullaccess  
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 setti
<input checked="" type="checkbox"/>	AmazonS3FullAccess	AWS managed	None	Provides full access to all buckets v
<input type="checkbox"/>	AmazonS3Object...	AWS managed	None	Provides AWS Lambda functions pe
<input type="checkbox"/>	AmazonS3Outpost...	AWS managed	None	Provides full access to Amazon S3 r
<input type="checkbox"/>	AmazonS3Outpost...	AWS managed	None	Provides read only access to Amaz
<input type="checkbox"/>	AmazonS3ReadOn...	AWS managed	None	Provides read only access to all buc
<input type="checkbox"/>	AWSBackupService...	AWS managed	None	Policy containing permissions need
<input type="checkbox"/>	AWSBackupService...	AWS managed	None	Policy containing permissions need

Cancel **Create user group**

- 7) Check the user group 's3fullaccess'.
- Click 'Next'.

**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.

**User groups (1/1)**

	Group name	Users	Attached policies	Created
<input checked="" type="checkbox"/>	s3fullaccess	0	AmazonS3FullAccess	2024-02-17 (Now)

**Set permissions boundary - optional**

Cancel Previous **Next**

## 8) Review all details & click 'Create user'.

The screenshot shows the 'Review and create' step of the AWS IAM user creation process. The left sidebar indicates the progress: Step 1 (Specify user details), Step 2 (Set permissions), Step 3 (Review and create), and Step 4 (Retrieve password). The main content area is titled 'Review and create' and includes a sub-header 'Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.'

**User details**

User name	Console password type	Require password reset
user1	Custom password	No

**Permissions summary**

Name	Type	Used as
s3FullAccess	Group	Permissions group

**Tags - optional**

Tag are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tags.

Buttons at the bottom: Cancel, Previous, Create user.

## 9) IAM user is created. Download the csv file for future use. Click 'Return to users list'.

The first screenshot shows the 'Retrieve password' page. It provides the console sign-in URL: <https://21125759233.signin.aws.amazon.com/console>. It also displays the user name 'user1' and a masked console password with a 'Show' button. Buttons at the bottom include 'Cancel', 'Download .csv file', and 'Return to users list'.

The second screenshot shows the 'User created successfully' page. It includes a 'View user' button and a table of users. The table has columns for User name, Path, Group, Last activity, MFA, Password age, Console last sign-in, and Access key ID. The user 'user1' is listed with a path of '/' and a group of 's3FullAccess'.

## 10) Signout from the root user.

Go to login page of AWS console.

Select 'IAM user'.

Give 12 digit account ID from the url in downloaded csv file.(Ex-632288837993)

Click 'Next'.

Give 'IAM user name' (Ex-user1) & 'Password' (Ex-User1@1234) from the downloaded csv file & click 'Sign in'.

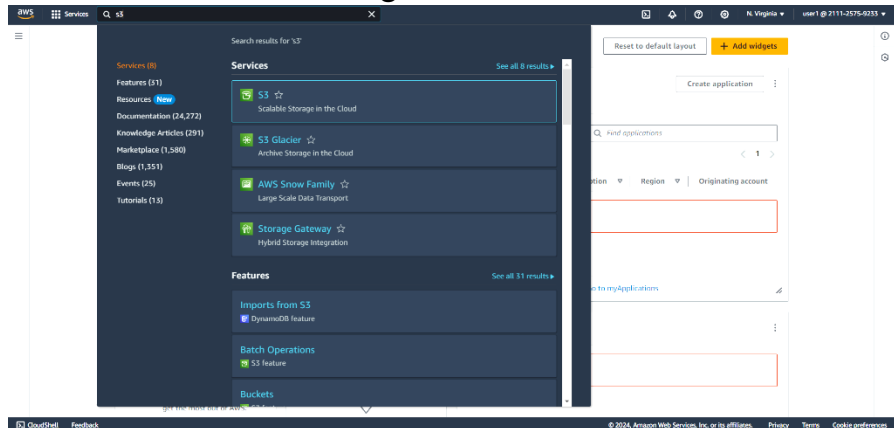
Now IAM user login successfully completed.

The screenshot shows the AWS 'Sign in' page. It has two options: 'Root user' (Account owner that performs tasks requiring unrestricted access) and 'IAM user' (User within an account that performs daily tasks). The 'IAM user' option is selected. Below the options, there is a field for 'Account ID (12 digits) or account alias' with the value '211125759233'. A 'Next' button is at the bottom.

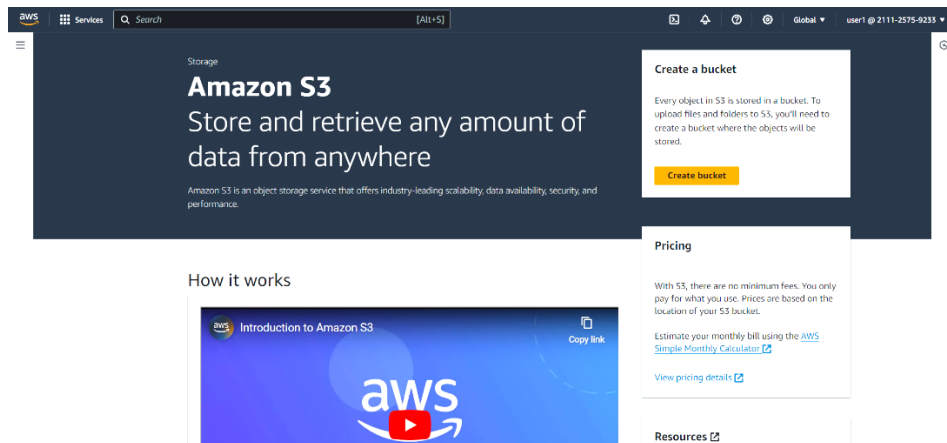
The screenshot shows the 'Sign in as IAM user' page. It has a field for 'Account ID (12 digits) or account alias' with the value '211125759233'. Below it is a field for 'IAM user name' with the value 'user1'. There is a 'Password' field with a masked password. A 'Remember this account' checkbox is present. A 'Sign in' button is at the bottom. Links for 'Sign in using root user email' and 'Forgot password?' are also visible.

The screenshot shows the AWS 'Console Home' page. It features a 'Recently visited' section with links to 'AWS Health Dashboard', 'S3', 'Billing and Cost Management', and 'IAM'. There is a 'View all services' button. The 'Applications' section shows a table with columns for Name, Description, Region, and Originating account. The 'Access denied' message is visible. The 'Cost and usage' section also shows an 'Access denied' message. The bottom of the page includes a footer with copyright information and links for 'Privacy', 'Terms', and 'Cookie preferences'.

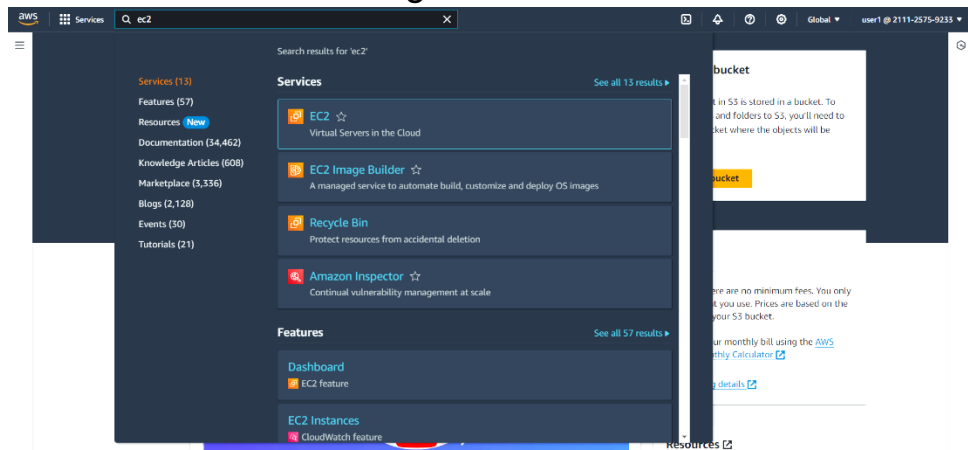
11) Now search for 'S3' & go to 'S3'.



12) We can use the features of 'S3' as 'user1' has full access to 'S3'.



13) Now search for 'EC2' & go to 'EC2'.



14) We cannot use the features of 'EC2' as 'user1' has no access to 'EC2'.

