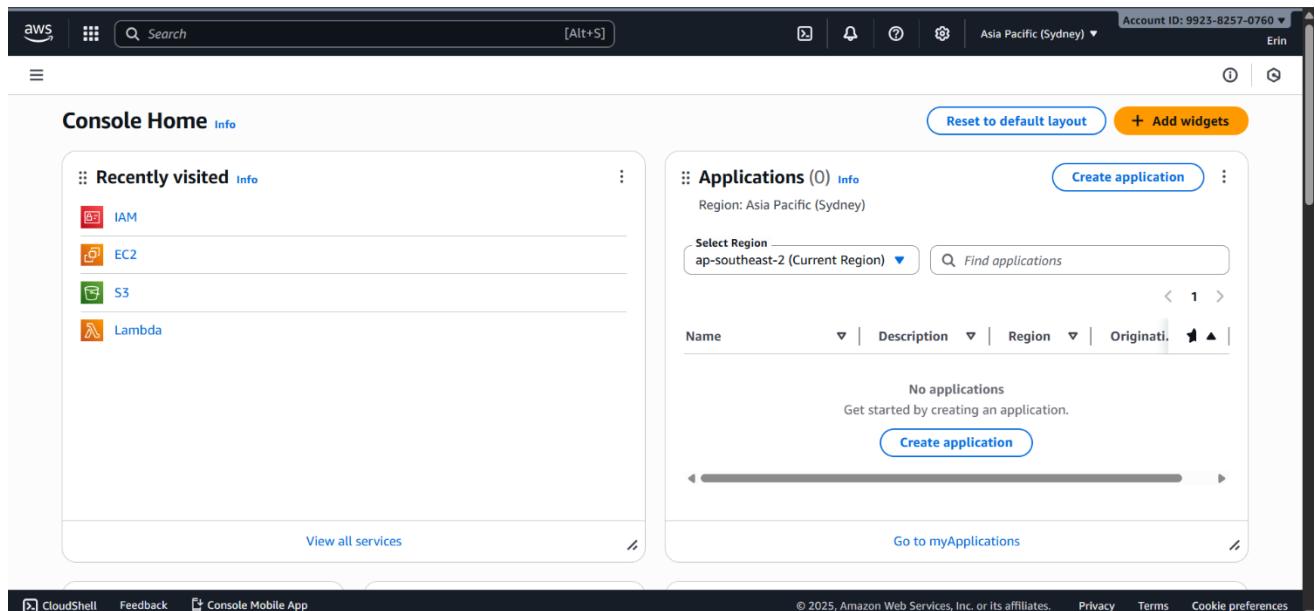


Practical:-5

Objective:- Creating an IAM (Identity and Access Management) user in AWS is to provide secure and controlled access to AWS resources for individuals or applications without using the root account.

Step 1: Sign in to AWS Console

Log in to your **AWS Management Console** using your **root account** or an **IAM user** who has **Administrator privileges**.



Step 2: Open IAM Service

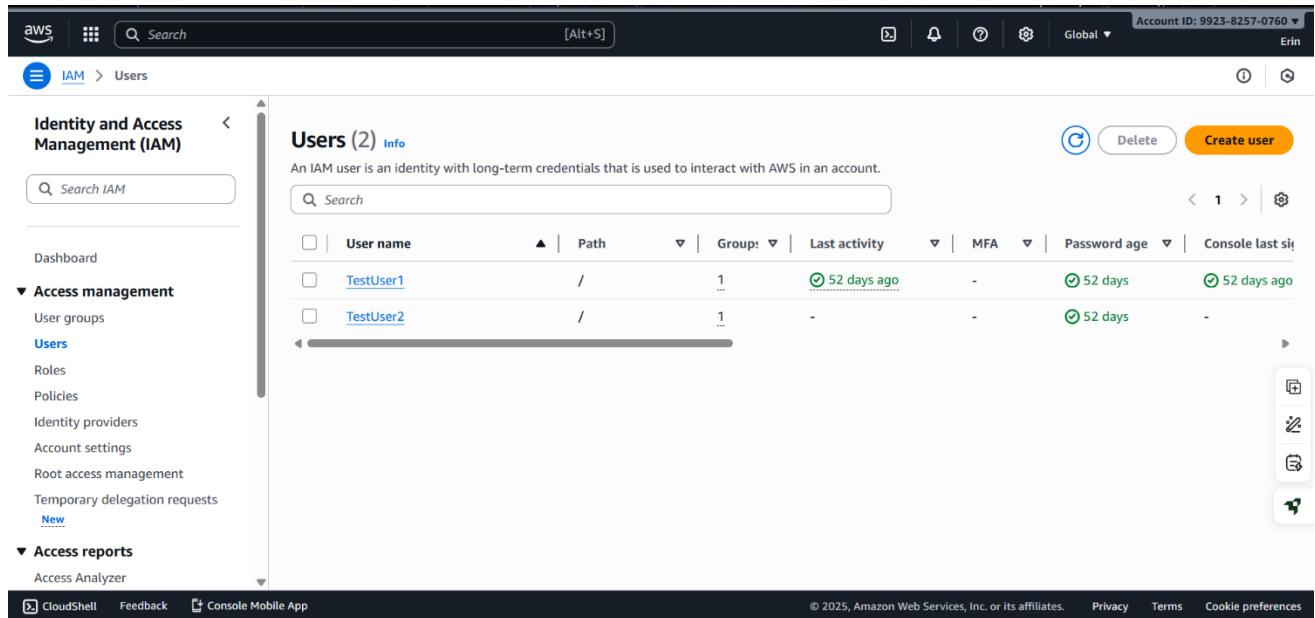
In the search bar at the top, type **IAM**. Select **IAM (Identity and Access Management)**.

The screenshot shows the AWS search interface. In the search bar at the top, the text "iam" is typed. Below the search bar, the "Services" section is expanded, showing three items: "IAM", "IAM Identity Center", and "Resource Access Manager". The "IAM" item is highlighted with a blue border. To the right of the services, there is a sidebar with options like "Create application" and "Region". At the bottom of the search results, there is a feedback section asking "Were these results helpful?" with "Yes" and "No" buttons.

The screenshot shows the IAM Dashboard. On the left, there is a navigation sidebar with sections for "Identity and Access Management (IAM)", "Dashboard", "Access management", "Access reports", and "Access analyzer". The main area is titled "IAM Dashboard" and contains two main sections: "Security recommendations" and "IAM resources". The "Security recommendations" section lists two items: "Root user has MFA" and "Root user has no active access keys". The "IAM resources" section shows the following counts: User groups (1), Users (2), Roles (5), Policies (3), and Identity providers (0). To the right of the dashboard, there is a "AWS Account" summary box showing the Account ID (992382570760) and a "Quick Links" box with a "My security credentials" link.

Step 3: Go to Users Section

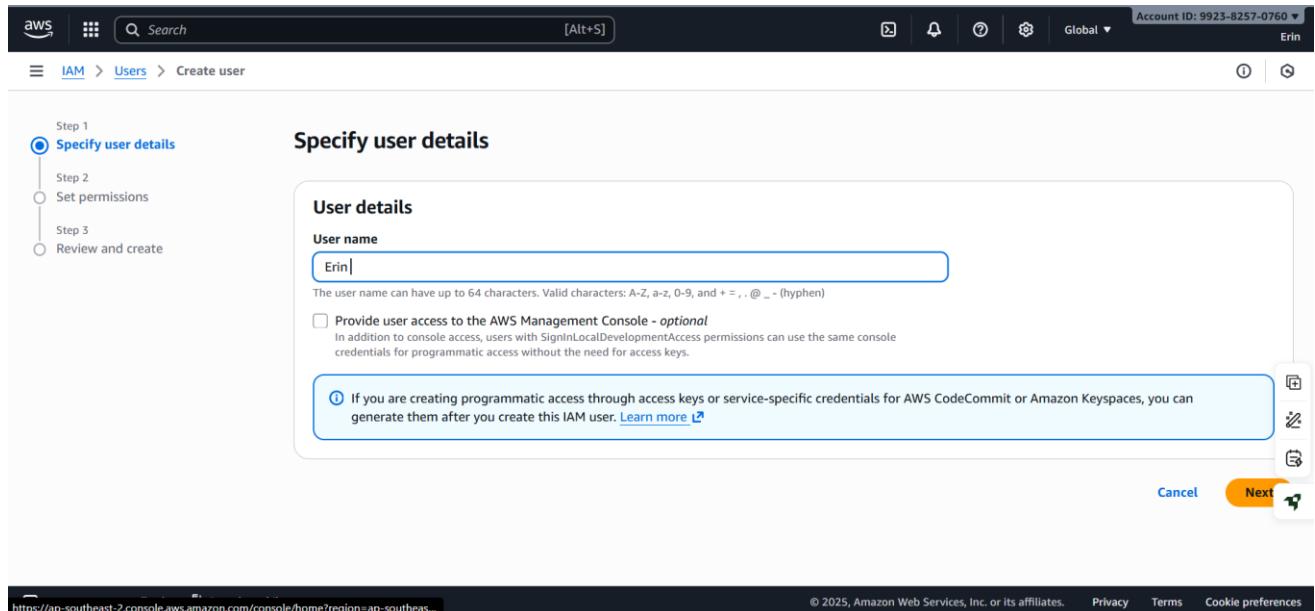
In the left sidebar, click on **Users**. You will see a list of all existing users. Click “**Create user**” to add a new one.



The screenshot shows the AWS IAM console with the 'Users' section selected. The left sidebar includes options like 'Dashboard', 'Access management' (with 'Users' selected), 'Policies', 'Identity providers', 'Account settings', 'Root access management', and 'Temporary delegation requests'. The main area displays a table of users with columns for 'User name', 'Path', 'Groups', 'Last activity', 'MFA', 'Password age', and 'Console last sign-in'. Two users are listed: 'TestUser1' and 'TestUser2'. A 'Create user' button is visible at the top right of the user list.

Step 4: Enter User Details

Enter a **User name**. Choose the type of access: **Password access** → if the user needs to log in to the AWS Console. Click **Next**.



The screenshot shows the 'Specify user details' step of the 'Create user' wizard. On the left, a navigation pane shows 'Step 1: Specify user details' (selected), 'Step 2: Set permissions', and 'Step 3: Review and create'. The main area is titled 'Specify user details' and contains a 'User details' section. It includes a 'User name' field with 'Erin' entered, a note about character restrictions, and a checkbox for 'Provide user access to the AWS Management Console - optional'. A callout box provides instructions for generating programmatic access keys. At the bottom right are 'Cancel' and 'Next' buttons.

Provide user access to the AWS Management Console - optional
In addition to console access, users with `SigninLocalDevelopmentAccess` permissions can use the same console credentials for programmatic access without the need for access keys.

Autogenerated password
You can view the password after you create the user.

Custom password
Enter a custom password for the user.
.....
• Must be at least 8 characters long
• Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & () _ + - (hyphen) = [] { } !'

Show password

Users must create a new password at next sign-in - Recommended
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

Tip: If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

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

Step 5: Set Permissions

Choose **Attach policies directly** → assign permissions manually (e.g., `AmazonS3FullAccess`, `AdministratorAccess`, etc.).

Then click **Next**.

Step 1 Specify user details
Step 2
Step 3
Set permissions
Step 4
Review and create
Retrieve password

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.

Permissions policies (1426)

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

Filter by Type
Search: All types
Policy name:
Type: Policy name AccessAnalyzerServiceRolePolicy
Attached entities: AWS managed
0

[Create policy](#)

The screenshot shows the AWS IAM 'Create user' interface. A search bar at the top contains the query 's3'. Below it is a table listing 17 matches for AWS managed policies. The policy 'AmazonS3FullAccess' is selected, highlighted with a blue border. Other policies listed include 'AmazonDMSRedshiftS3Role', 'AmazonS3ObjectLambdaExecutionRole', 'AmazonS3OutpostsFullAccess', 'AmazonS3OutpostsReadOnlyAccess', 'AmazonS3ReadOnlyAccess', 'AmazonS3TablesFullAccess', 'AmazonS3TablesLakeFormationAccess', 'AmazonS3TablesReadOnlyAccess', 'AWSBackupServiceRolePolicyForAWSBackup', 'AWSBackupServiceRolePolicyForAWSBackup', and 'AWSQuickSetupSSMDeploymentRole'. The table has columns for 'Policy name', 'Type', and 'Attached entities'.

Step 6: Review and Create

Review all details carefully. Click **Create user**.

The screenshot shows the 'Review and create' step of the AWS IAM 'Create user' wizard. On the left, a navigation pane lists steps: Step 1 (Specify user details), Step 2 (Set permissions), Step 3 (Review and create), Step 4 (Retrieve password). Step 3 is currently selected. The main area contains three sections: 'User details' (User name: srishti2, Console password type: Custom password, Require password reset: Yes), 'Permissions summary' (listing 'AmazonS3FullAccess' and 'IAMUserChangePassword' policies), and 'Tags - optional' (a note about tags). At the bottom right, there is a link to 'Activate Windows'.

IAM > Users > Create user

Permissions summary

Name	Type	Used as
AmazonS3FullAccess	AWS managed	Permissions policy
IAMUserChangePassword	AWS managed	Permissions policy

Tags - optional
Tags 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.

Cancel Previous Create user
Activate Windows
Go to Settings to activate Windows.

Step 7: Save Login Details

- Once the user is created, AWS will show:
 - User ARN (Amazon Resource Name)**
 - Console login link**
 - Password or Access key/Secret key** (Download the .csv file — it won't be shown again).

IAM > Users > Create user

User created successfully
You can view and download the user's password and email instructions for signing in to the AWS Management Console.

[View user](#)

Step 1 Specify user details
Step 2 Set permissions
Step 3 Review and create
Step 4 Retrieve password

Retrieve password
You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details

Console sign-in URL
 <https://992382570760.signin.aws.amazon.com/console>

User name
 Erin

Console password
 ***** [Show](#)

Email sign-in instructions [Link](#)

aws | Search [Alt+S] | Recent download history | -0760 | Erin

IAM > Users > Create user

User created successfully
You can view and download the user's password and email instructions for signing in to the AWS Management Console.

[View user](#)

Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password

Retrieve password
You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details

Console sign-in URL
<https://992382570760.signin.aws.amazon.com/console>

User name
[Erin](#)

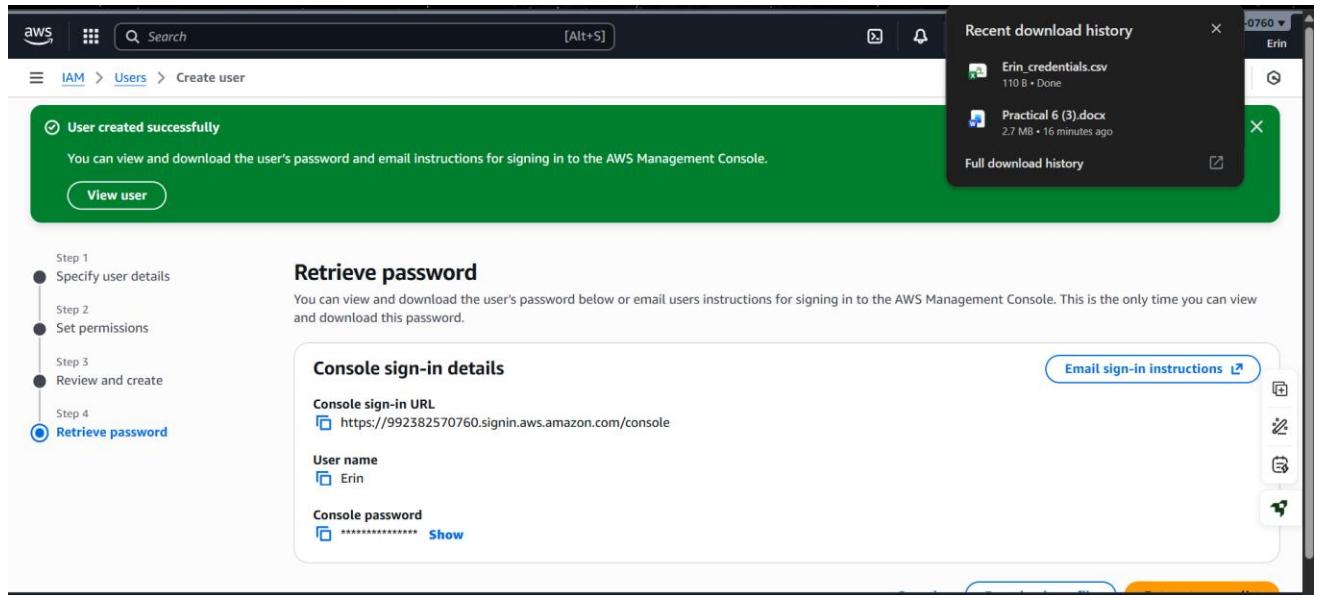
Console password
[***** Show](#)

[Email sign-in instructions](#)

Recent download history

- Erin.credentials.csv (110 B • Done)
- Practical 6 (3).docx (2.7 MB • 16 minutes ago)

Full download history



Practical:-7

Objective: To create a user group in AWS IAM in order to manage permissions collectively for multiple users having similar roles or responsibilities.

Step1:- Open the IAM Service:

In the search bar at the top of the console, type **IAM**, then select **Identity and Access Management** from the results.

The screenshot shows the IAM Dashboard with the following details:

- Security recommendations:**
 - Root user has MFA (Green)
 - Root user has no active access keys (Green)
- IAM resources:** Resources in this AWS Account
 - User groups: 3
 - Users: 3
 - Roles: 7
 - Policies: 1
 - Identity providers: 0
- AWS Account:**
 - Account ID: 372212891777
 - Account Alias: Create
 - Sign-in URL for IAM users in this account: https://372212891777.sigin.aws.amazon.com/console
- Quick Links:**
 - My security credentials
 - Manage your access keys, multi-factor authentication (MFA) and other credentials.
 - Activate Windows
 - Go to Settings to activate Windows.

Step 2:- Go to User Groups Section:

In the left-hand sidebar, click on **User groups**.

The screenshot shows the IAM Dashboard with the following details:

- Security recommendations:**
 - Root user has MFA (Green)
 - Root user has no active access keys (Green)
- IAM resources:** Resources in this AWS Account
 - User groups: 3
 - Users: 3
 - Roles: 7
 - Policies: 1
 - Identity providers: 0
- AWS Account:**
 - Account ID: 372212891777
 - Account Alias: Create
 - Sign-in URL for IAM users in this account: https://372212891777.sigin.aws.amazon.com/console
- Quick Links:**
 - My security credentials
 - Manage your access keys, multi-factor authentication (MFA) and other credentials.
 - Activate Windows
 - Go to Settings to activate Windows.

The screenshot shows the AWS Identity and Access Management (IAM) console. The left sidebar has a search bar and navigation links for Dashboard, Access management (User groups, Roles, Policies, Identity providers, Account settings, Root access management), and Access reports (Access Analyzer). The main content area is titled "User groups (3) Info" and contains a table of user groups. The table columns are Group name, Users, Permissions, and Creation time. The groups listed are:

Group name	Users	Permissions	Creation time
Administrator	2	Defined	5 days ago
Administrators	0	Defined	5 days ago
Group1	2	Defined	4 weeks ago

At the top right are "Delete" and "Create group" buttons. Below the table is a message: "Activate Windows Go to Settings to activate Windows."

Step 3:- Click on “Create group”:

On the User Groups page, click the “Create group” button to start creating a new group

Step 4:- Enter Group Name:

Type a **unique name** for your group (for example, *Developers*, *Admins*, or *ReadOnlyUsers*).

Step 5:- Attach Permissions Policies (Optional):

You can choose policies to attach to this group, such as:

- AmazonS3FullAccess
- AmazonEC2ReadOnlyAccess
- AdministratorAccess

If you want to add permissions later, you can **skip this step** and click **Next**.

The screenshot shows the AWS IAM console with the path: IAM > User groups > Create user group. On the left, there's a sidebar with 'Identity and Access Management (IAM)' selected. The main area is titled 'Attach permissions policies - Optional (3/1082)'. It says 'You can attach up to 10 policies to this user group. All the users in this group will have permissions that are defined in the selected policies.' A search bar shows 'admin'. A table lists policies with columns: Policy name, Type, Used as, and Description. One policy, 'AdministratorAccess', is selected (indicated by a checked checkbox). Other policies listed include 'AdministratorAccess-Amplify', 'AdministratorAccess-AWSElasticBeanstalk', 'AIOpsConsoleAdminPolicy', 'AmazonAPIGatewayAdministrator', 'AmazonNimbleStudio-StudioAdmin', 'AmazonSageMakerAdmin-ServiceCatalogProduct...', and 'AmazonSageMakerHyperPodObservabilityAdmin...'. The 'AdministratorAccess' row shows 'AWS managed - job func...' under 'Type', 'Permissions policy (8)' under 'Used as', and 'Provides full' under 'Description'.

Step 6:- Add Users to the Group (Optional):

You can select existing IAM users to include in this group now, or you can add users later after creating the group.

Step 7:- Review and Create Group:

Review the group details and attached policies, then click **Create group**.

The screenshot shows the AWS IAM console with the 'User groups' section selected. On the left, there's a navigation sidebar with 'Identity and Access Management (IAM)' at the top, followed by 'Dashboard', 'Access management', 'User groups' (which is highlighted), 'Users', 'Roles', 'Policies', 'Identity providers', 'Account settings', and 'Root access management'. Below that is 'Access reports' with 'Access Analyzer'. At the top right, it shows 'Account ID: 3722-1289-1777', the user name 'Srishti Mishra', and various global settings icons. The main content area is titled 'Create user group' and contains a table listing ten existing IAM groups. The columns are: 'Select' (checkbox), 'Group name' (link), 'Type' (AWS managed), 'Status' (None), and 'Policies' (Provides admin). The groups listed are: AmazonWorkSpacesApplicationManagerAdminAccess, AWS-SSM-DiagnosisAutomation-AdministrationRole, AWS-SSM-DiagnosisAutomation-OperationalAccess, AWS-SSM-RemediationAutomation-AdministratorAccess, AWS-SSM-RemediationAutomation-OperationalAccess, AWSAppSyncAdministrator, AWSAuditManagerAdministratorAccess, AWSBackupOrganizationAdminAccess, AWSBudgetsActions_RolePolicyForResourceAdministrator, and AWSCloud9Administrator. At the bottom right of the table are 'Cancel' and 'Create user group' buttons, with a note below them: 'Activate Windows Go to Settings to activate Windows.'

Select	Group name	Type	Status	Policies
<input type="checkbox"/>	AmazonWorkSpacesApplicationManagerAdminAccess	AWS managed	None	Provides admin
<input type="checkbox"/>	AWS-SSM-DiagnosisAutomation-AdministrationRole	AWS managed	None	Provide perm
<input type="checkbox"/>	AWS-SSM-DiagnosisAutomation-OperationalAccess	AWS managed	None	Provides perm
<input type="checkbox"/>	AWS-SSM-RemediationAutomation-AdministratorAccess	AWS managed	None	Provide perm
<input type="checkbox"/>	AWS-SSM-RemediationAutomation-OperationalAccess	AWS managed	None	Provides perm
<input type="checkbox"/>	AWSAppSyncAdministrator	AWS managed	None	Provides admin
<input type="checkbox"/>	AWSAuditManagerAdministratorAccess	AWS managed	None	Provides admin
<input type="checkbox"/>	AWSBackupOrganizationAdminAccess	AWS managed	None	This policy is
<input type="checkbox"/>	AWSBudgetsActions_RolePolicyForResourceAdministrator	AWS managed	None	This policy gi
<input type="checkbox"/>	AWSCloud9Administrator	AWS managed	None	Provides adm

IAM > User groups

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups (selected)
- Users
- Roles
- Policies
- Identity providers
- Account settings
- Root access management

Access reports

- Access Analyzer
- Resource analysis New

User groups (4) Info

A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

Search

Group name	Users	Permissions	Creation time
Administrator	2	Defined	5 days ago
Administrators	0	Defined	5 days ago
Developer	3	Defined	Now
Group1	2	Defined	4 weeks ago

View group Delete Create group

Activate Windows
Go to Settings to activate Windows

Practical:-8

Objective:-To create a security role in AWS IAM that allows AWS services or users to securely access specific AWS resources with defined permissions, ensuring controlled and temporary access without sharing long-term credentials.

Step 1:- Open the IAM Service:

In the search bar at the top of the console, type **IAM**, then select **Identity and Access Management** from the results.

The screenshot shows the IAM Dashboard. On the left, the navigation pane includes 'Dashboard', 'Access management' (User groups, Users, Roles, Policies, Identity providers, Account settings, Root access management), and 'Access reports' (Access Analyzer). The main area displays 'Security recommendations' with two items: 'Root user has MFA' (Having multi-factor authentication (MFA) for the root user improves security for this account) and 'Root user has no active access keys' (Using access keys attached to an IAM user instead of the root user improves security). Below this is the 'IAM resources' section, which shows counts for User groups (3), Users (3), Roles (7), Policies (1), and Identity providers (0). To the right is the 'AWS Account' section, which includes the Account ID (372212891777), Account Alias (Create), and Sign-in URL (https://372212891777.sigin.aws.amazon.com/console). A 'Quick Links' section for 'My security credentials' is also present.

Step 2:- Go to Roles Section:

In the left-hand navigation pane, click on **Roles**.

The screenshot shows the 'Roles' page. The left navigation pane is identical to the previous dashboard view. The main area shows a table titled 'Roles (4)' with columns for 'Role name', 'Trusted entities', and 'Last activity'. The roles listed are 'AWSServiceRoleForResourceExplorer' (AWS Service: resource-explorer-2, Last activity 13 minutes ago), 'AWSServiceRoleForSupport' (AWS Service: support (Service-Linker), Last activity -), 'AWSServiceRoleForTrustedAdvisor' (AWS Service: trustedadvisor (Service), Last activity -), and 'myfirstlambda-role-x54bi11s' (AWS Service: lambda, Last activity 47 days ago). Below the table is a 'Roles Anywhere' section with a note about authenticating non-AWS workloads and a 'Manage' button. At the bottom, there are links for 'Access AWS from your non AWS' (X.509 Standard), 'Temporary credentials' (Activate Windows), and 'Activate Windows'.

Step 3:- Click on “Create role”:

On the Roles page, click the “Create role” button to start the process.

The screenshot shows the 'Select trusted entity' step of the IAM Role creation wizard. On the left, a sidebar lists steps: Step 1 (Select trusted entity), Step 2 (Add permissions), and Step 3 (Name, review, and create). Step 1 is highlighted with a blue circle. The main area is titled 'Select trusted entity' with an 'Info' link. It features a section titled 'Trusted entity type' with five options: 'AWS service' (selected, highlighted in blue), 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. Below this is a 'Use case' section with a dropdown menu set to 'EC2'. At the bottom right are 'Activate Windows' and 'Go to Settings to activate Windows' links.

Step 4:- Select Trusted Entity Type:

Choose who will use the role, such as:

- **AWS Service** (e.g., EC2, Lambda)
- **Another AWS Account**
- **Web Identity or SAML 2.0 Federation**

Click **Next** after selecting the appropriate option.

The screenshot shows the 'Use case' step of the IAM Role creation wizard. The sidebar shows Step 1 (selected) and Step 2 (Add permissions). The main area has a 'Use case' section with a dropdown menu showing 'EC2'. Below it is a 'Service or use case' section with a dropdown menu also set to 'EC2'. A detailed list of EC2 use cases follows: 'EC2' (selected), 'EC2 Role for AWS Systems Manager', 'EC2 Spot Fleet Role', 'EC2 - Spot Fleet Auto Scaling', 'EC2 - Spot Fleet Tagging', 'EC2 - Spot Instances', and 'EC2 - Spot Fleet'. At the bottom right are 'Activate Windows' and 'Go to Settings to activate Windows' links.

Step 5:-Attach Permissions Policies:

Select the **permissions policies** that define what actions the role can perform (for example, AmazonS3FullAccess or AmazonEC2FullAccess).

The screenshot shows the 'Add permissions' step of creating a new IAM role. On the left, a sidebar lists three steps: Step 1 (Select trusted entity), Step 2 (Add permissions - currently selected), and Step 3 (Name, review, and create). The main area is titled 'Add permissions' and displays a list of 'Permissions policies (1082)'. A search bar and a 'Filter by Type' dropdown are at the top. Below is a table with columns for Policy name, Type, and Description. Some policies have checkboxes next to them. A note at the bottom right says 'Activate Windows'.

Policy name	Type	Description
AdministratorAccess	AWS managed - job function	Provides full access to AWS services an...
AdministratorAccess-Amplify	AWS managed	Grants account administrative permis...
AdministratorAccess-AWSElasti...	AWS managed	Grants account administrative permis...
AIOpsAssistantIncidentReportP...	AWS managed	Provides permissions required by the A...
AIOpsAssistantPolicy	AWS managed	Provides ReadOnly permissions requir...
AIOpsConsoleAdminPolicy	AWS managed	Grants full access to Amazon AI Opera...

The screenshot shows the 'Add permissions' step of creating a new IAM role. The sidebar and main title are identical to the previous screenshot. The main area displays a list of 'Permissions policies (1082)' with a subset shown in the table below. A note at the bottom right says 'Activate Windows'.

Policy name	Type	Description
AmazonEC2RoleforAWSCodeDeployLimited	AWS managed	Provides EC2 limited access t
AmazonEC2RoleforDataPipelineRole	AWS managed	Default policy for the Amazo
AmazonEC2RoleforSSM	AWS managed	This policy will soon be depr
AmazonEC2RolePolicyForLaunchWizard	AWS managed	Managed policy for the Ama
AmazonEC2SpotFleetAutoscaleRole	AWS managed	Policy to enable Autoscaling
AmazonEC2SpotFleetTaggingRole	AWS managed	Allows EC2 Spot Fleet to req
AmazonElasticMapReduceforEC2Role	AWS managed	Default policy for the Amazo
AmazonSSMManagedEC2InstanceDefaultPolicy	AWS managed	This policy enables AWS Syst

▶ Set permissions boundary - optional

Cancel Previous Next
Activate Windows
Go to Settings to activate Windows.

Step 6:- Name and Review the Role:

Enter a **role name** (for example, *EC2SecurityRole* or *LambdaAccessRole*) and review all selected settings.

IAM > Roles > Create role

Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.
Developer
Maximum 64 characters. Use alphanumeric and '+-,.,@-' characters.

Description
Add a short explanation for this role.
Allows EC2 instances to call AWS services on your behalf.
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/[\{\}][#\$%^()~`]

Step 1: Select trusted entities

Trust policy

Activate Windows
Go to Settings to activate Windows.

IAM > Roles > Create role

Permissions policy summary

Policy name	Type	Attached as
AmazonEC2FullAccess	AWS managed	Permissions policy
AmazonS3FullAccess	AWS managed	Permissions policy

Step 3: Add tags

Add tags - optional Info
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.
No tags associated with the resource.

Add new tag
You can add up to 50 more tags.

Cancel Previous **Create role**
Activate Windows
Go to Settings to activate Windows.

Step7:- Create the Role:

Click **Create role** to finish.

The screenshot shows the AWS IAM Roles page. At the top, a green banner displays the message "Role Developer created." with a "View role" button. Below the banner, the title "Roles (5) Info" is shown, along with a description: "An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust." A search bar labeled "Search" is available. The main table lists five roles:

Role name	Trusted entities	Last activity
AWSServiceRoleForResourceExplorer	AWS Service: resource-explorer-2 (Service-Linker)	22 minutes ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linker)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service)	-
Developer	AWS Service: ec2	-
myfirstlambda-role-x54bi11s	AWS Service: lambda	47 days ago

At the bottom right, there is a "Manage" button and a note: "Activate Windows Go to Settings to activate Windows".