

Terna Engineering College

Computer Engineering Department

Program: Sem VI

Course: Cloud Computing Lab(CSL605)

PART A

(PART A: TO BE REFERRED BY STUDENTS)

Experiment No.8

A.1 Aim:

Understand Security of Web Server and demonstration of IAM using own cloud/AWS

A.2 Prerequisite:

Knowledge of Access Control, Authentication and Authorization

A.3 Objective:

Objectives this experiment is to provide students an overview of Security issues of Cloud and how to manage various user groups over cloud.

A.4 Outcome: (LO 4)

After successful completion of this experiment student will be able to

- Analyze security issues on cloud

A.5 Theory:



AWS Identity and Access Management (IAM)

IAM refers to a **framework or policies** and **technologies** for ensuring that **the people in an organization have the appropriate access to technology resources.**

OR

AWS Identity Access Management (IAM) is a web service that helps you securely control access to AWS resources. You use IAM to control to check who is authenticated (signed-in) and authorized to use resources.



Sign in

☒ **Root user**

Account owner that performs tasks requiring unrestricted access. [Learn more](#)

☐ **IAM user**

User within an account that performs daily tasks. [Learn more](#)

Root user email address

username@example.com

Next

By continuing, you agree to the [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

— New to AWS? —

Create a new AWS account

Note: Only one account is created (if 200 employees using same account then there is need of IAM)(HR+ Marketing+ Finance+ Development)(for 200 employees account need to be created)

- IAM user Limits=500 user per root account(AWS account)
- 300 groups per AWS account (like HR, Development, Etc)
- 1000 roles per AWS account

- ✓ When you first create an AWS account, you begin with a single sign-in identity **that has complete access to all AWS services and resources in the account.**
- ✓ This identity is called the AWS account root user and is accessed by signing in with the email address and password that you used to create the account.

- ✓ **We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones.** Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

IAM gives you the following features:

1. Shared access to your AWS account:

You can grant other people permission to administer and use resources in your AWS account without having to share your password or access key. (by creating user name and password)

2. Granular permissions (Read only/Read write/etc permission)

You can grant different permissions to different people for different resources.

For example, you might allow some users complete access to Amazon Elastic Compute Cloud (Amazon EC2), Amazon Simple Storage Service (Amazon S3), Amazon Dynamo, Amazon Red-shift, and other AWS services.

For other users, you can allow read-only access to just some S3 buckets, or permission to administer just some EC2 instances, or to access your billing information but nothing else.

3. Secure access to AWS resources for applications that run on Amazon EC2

You can use IAM features to securely provide credentials for applications that run on EC2 instances. These credentials provide permissions for your application to access other AWS resources.

Examples include S3 buckets and Dynamo tables.

4. Multi-factor authentication (MFA)

You can add two-factor authentication to your account and to individual users for extra security. With MFA you or your users must provide not only a password or access key to work with your account, **but also a code from a specially configured device.**

5. Identity federation

You can allow users, who already have passwords elsewhere (like face-book or elsewhere can use that account and login)— for example, in your corporate network or with an internet identity provider—to **get temporary access to your AWS account. (Trust between company ids (or face book id, Gmail id) and AWS)**

6. Identity information for assurance

If you use AWS Cloud-Trail, **you receive log records that include information about those who made requests for resources in your account.** That information is based on IAM identities.

7. PCI DSS Compliance

IAM supports the processing, storage, and transmission of credit card data by a merchant or service provider, and has been validated as being compliant with Payment Card Industry (PCI) Data Security Standard (DSS). For more information about PCI DSS, including how to request a copy of the AWS PCI Compliance Package, see PCI DSS Level 1.

8. Integrated with many AWS services

For a list of AWS services that work with IAM, see AWS Services That Work with IAM (p. 502).

9. Eventually Consistent

IAM, like many other AWS services, is eventually consistent (IAM work is replicated like multiple zones). IAM achieves high availability by replicating data across multiple servers within Amazon's data centres around the world.

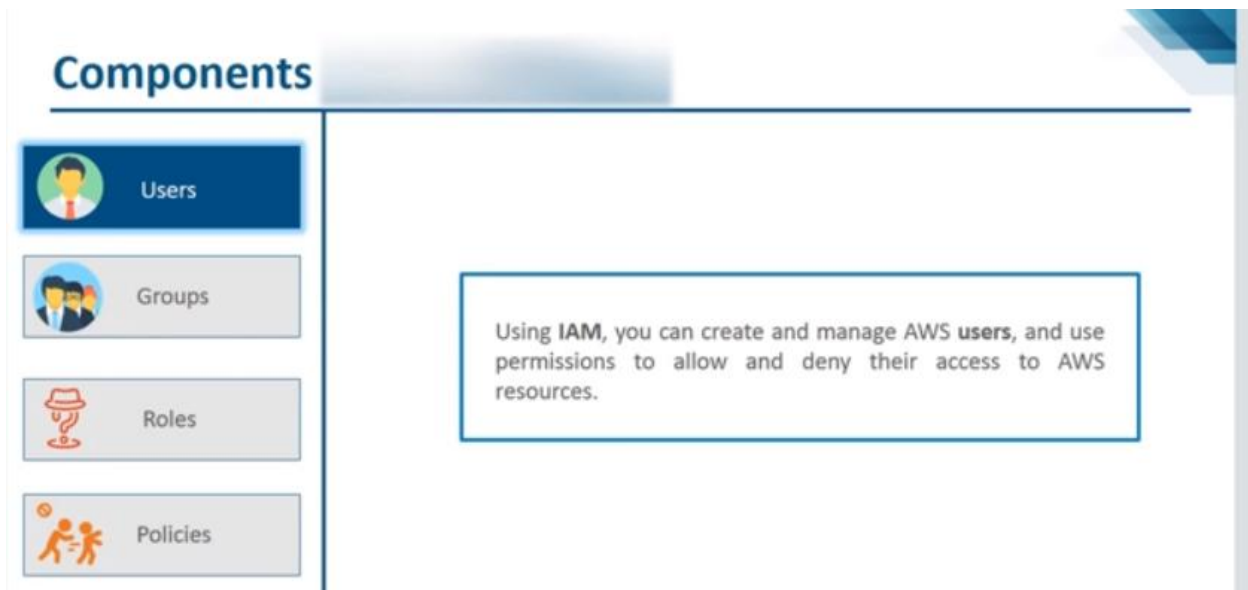
If a request to change some data is successful, the change is committed and safely stored. However, the change must be replicated across IAM, which can take some time. Such changes include creating or updating users, groups, roles, or policies. We recommend that you do not include such IAM changes in the critical, high-availability code paths of your application. Instead, make IAM changes in a separate initialization or setup routine that you run less frequently. Also, be sure to verify that the changes have been propagated before production workflows depend on them. For more information, see *Changes that I make are not always immediately visible* (p. 466).

10. Free to use

AWS Identity and Access Management (IAM) and AWS Security Token Service (AWS STS) are features of your AWS account offered at no additional charge. You are charged only when you access other AWS services using your IAM users or AWS STS temporary security credentials. For information about the pricing of other AWS products, see the [Amazon Web Services pricing page](#).

Components of IAM

1. Users
2. Groups
3. Roles
4. Policies



The users created, can also be divided among **groups**, and then the **rules** and **policies** that apply on the **group**, apply on the user level as well.

An IAM **role** is an IAM entity that defines a set of permissions for making **AWS** service requests. IAM **roles** are not associated with a specific user or group. Instead, trusted entities assume **roles**, such as IAM users, applications, or **AWS** services such as EC2

I.e. roles are assigned to applications, user are assigned to peoples

(Suppose you have created an EC2 instance and inside that instance you have hosted a website and it is accessing s3 services, i.e. application has to interact with S3 service, so I have to give permission to that web application to access that s3 service, that to give permission you need to create a role)

Steps For demonstrating AWS IAM:

1. Login to root account
2. Go to dashboard-Click on IAM
3. Create new user(IAM user-give rights)(copy the URL , so that IAM user login using that URL-Need to remember username and password)
4. Login using IAM user (use provided URL), create group(give rights) and then add user.
5. Login to root account, Go to dashboard- create policies as per requirement and attached policies to applications.

Accessing IAM:

You can work with AWS Identity and Access Management in any of the following ways.

1. AWS Management Console

The console is a browser-based interface to manage IAM and AWS resources. For more information about accessing IAM through the console, see *The IAM Console and Sign-in Page* (p. 55). For a tutorial that guides you through using the console, see *Creating Your First IAM Admin User and Group* (p. 17).

2. AWS Command Line Tools

You can use the AWS command line tools to issue commands at your system's command line to perform IAM and AWS tasks. Using the command line can be faster and more convenient than the console. The command line tools are also useful if you want to build scripts that perform AWS tasks.

AWS provides two sets of command line tools: the AWS Command Line Interface (AWS CLI) and the AWS Tools for Windows PowerShell. For information about installing and using the AWS CLI, see the *AWS Command Line Interface User Guide*. For information about installing and using the Tools for Windows PowerShell, see the *AWS Tools for Windows PowerShell User Guide*.

3. AWS SDKs

AWS provides SDKs (software development kits) that consist of libraries and sample code for various programming languages and platforms (Java, Python, Ruby, .NET, iOS, Android, etc.). The SDKs provide a convenient way to create programmatic access to IAM and AWS. For example, the SDKs take care of tasks such as cryptographically signing requests, managing errors, and retrying requests automatically. For information about the AWS SDKs, including how to download and install them, see the *Tools for Amazon Web Services* page.

4. IAM HTTPS API

You can access IAM and AWS programmatically by using the IAM HTTPS API, which lets you issue HTTPS requests directly to the service. When you use the HTTPS API, you must include code to digitally sign requests using your credentials. For more information, see *Calling the API by Making HTTP Query Requests* (p. 1239) and the *IAM API Reference*.

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the ERP or emailed to the concerned lab in charge faculties at the end of the practical in case there is no ERP access available)

Roll No. B48	Name: Aryan Unhale
Class :TE B COMPS	Batch :B3
Date of Experiment: 3/4/25	Date of Submission: 3/4/25
Grade :	

B.1 Question of Curiosity:

Q.1 Explain User Management in cloud computing Detail?

Ans:

User Management in Cloud Computing refers to the process of managing digital identities and the permissions assigned to those identities within cloud services. AWS uses IAM (Identity and Access Management) for user management.

Key Components:

IAM Users: Represent individual people or services.

IAM Groups: Collections of users with common permissions (e.g., HR, Dev).

IAM Roles: Temporary access given to services or applications.

Policies: Define permissions in JSON (read, write, full access, etc.)

AWS allows you to:

Create multiple users without sharing the root credentials.

Assign granular permissions.

Use MFA for stronger authentication.

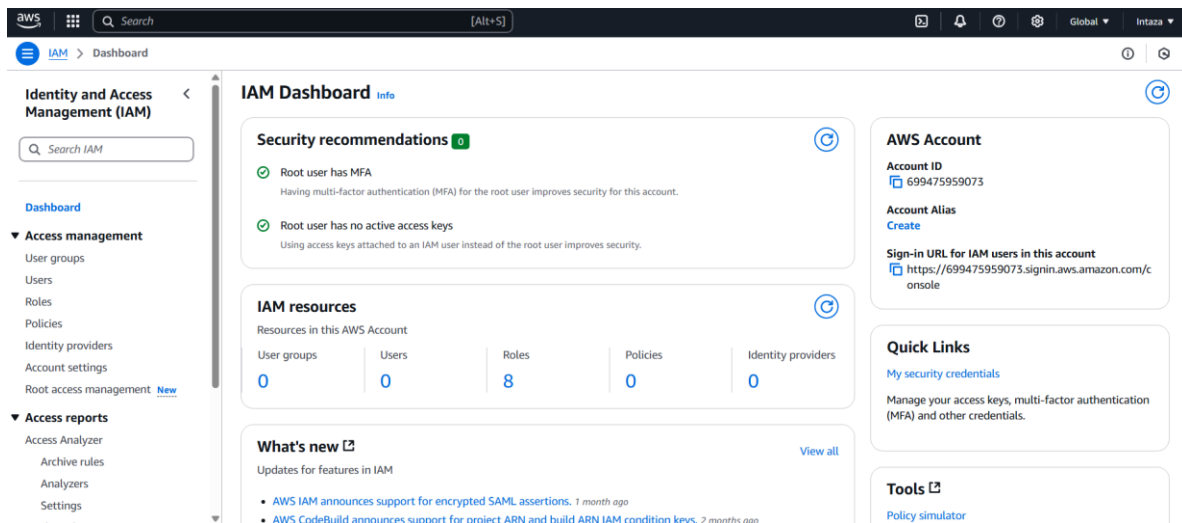
Audit activity through AWS CloudTrail.

Q.2 Add snapshots for creating IAM user and user groups (using AWS IAM service).

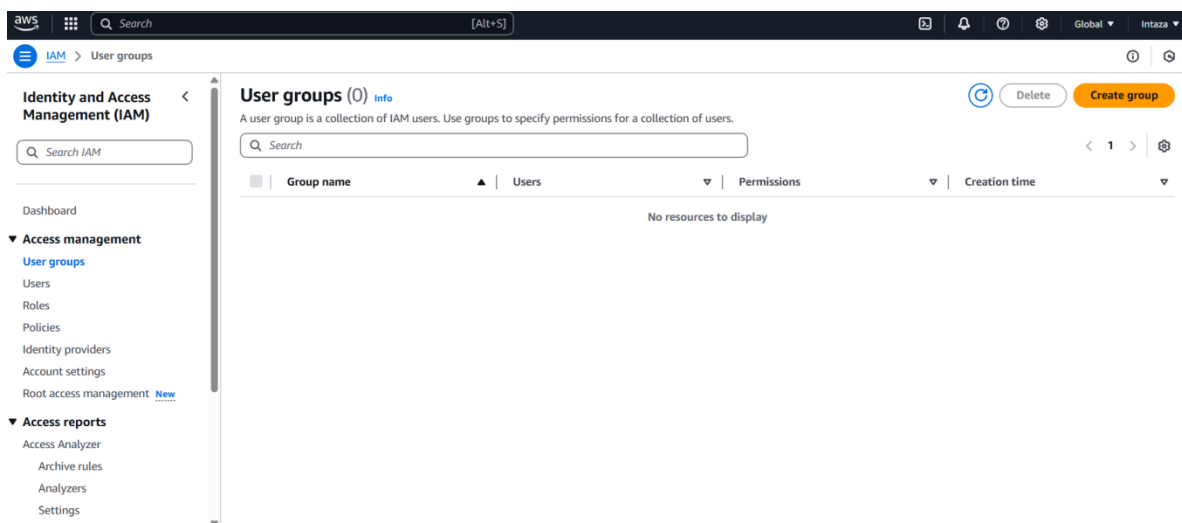
Step 1: Go to IAM Service

Open [AWS Console](#)

Search for IAM in the search bar and click it.



Step 2: Create a User Group



In the left sidebar, click on “User groups”

Enter a name (e.g., Developers, HR_Team)

Create user group

Name the group

User group name
Enter a meaningful name to identify this group.

Maximum 128 characters. Use alphanumeric and '+', '@', '-' characters.

Add users to the group - Optional (0) [Info](#)

An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.

☐ User name ☐ Groups | Last activity | Creation time

No resources to display

Attach permissions policies - Optional (1038) [Info](#)

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.

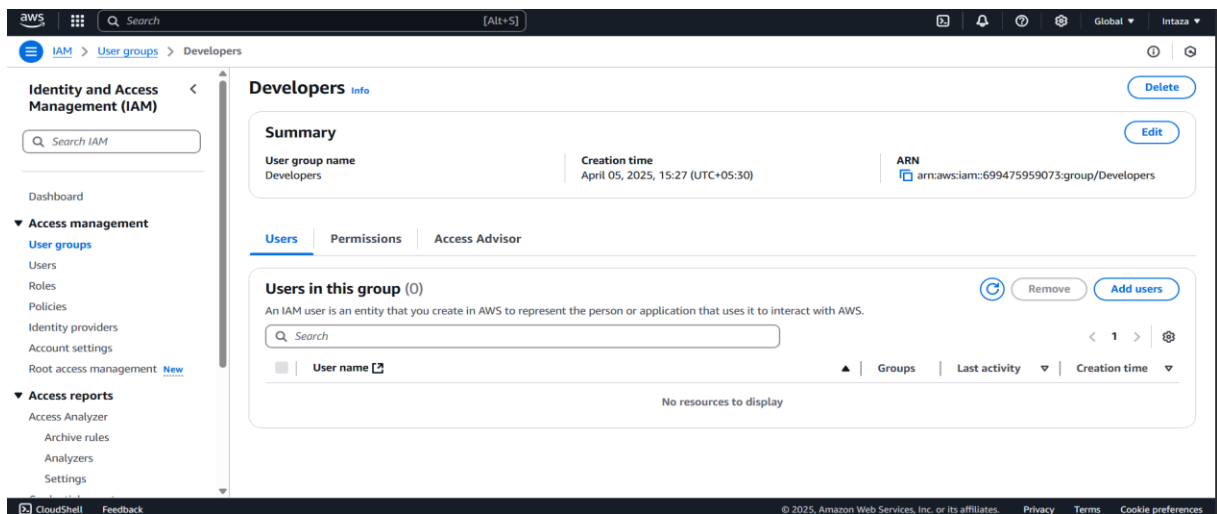
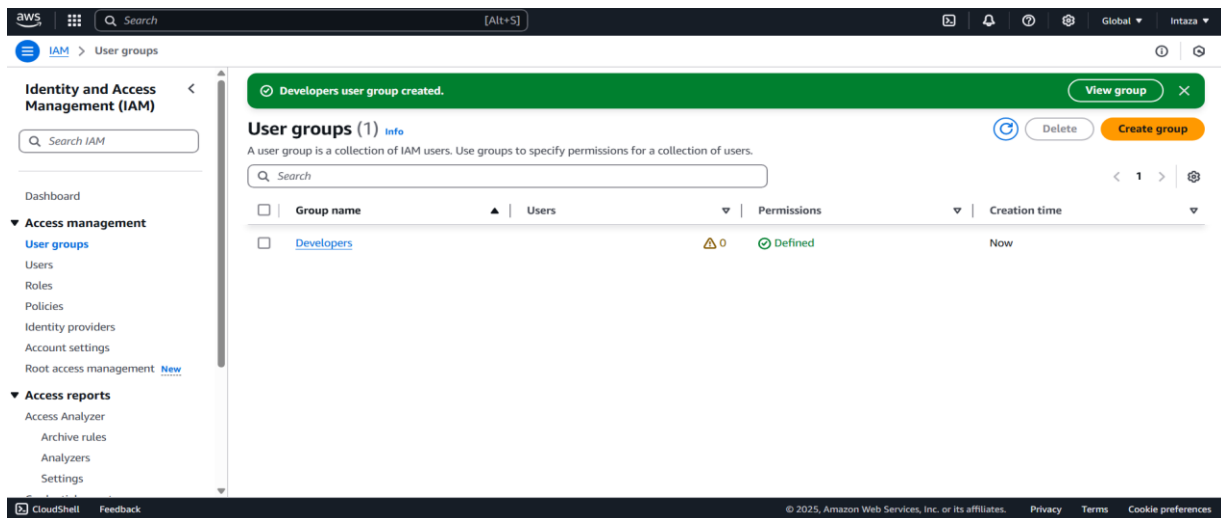
Filter by Type

Attach a permission policy (for example: AmazonS3ReadOnlyAccess or AdministratorAccess)

<input type="checkbox"/>	Policy Name	Managed By	Permissions	Description
<input type="checkbox"/>	AmazonDMSRedshiftS3...	AWS managed	None	Provides access to manage S3 settings ...
<input type="checkbox"/>	AmazonS3FullAccess	AWS managed	Permissions policy (1)	Provides full access to all buckets via t...
<input type="checkbox"/>	AmazonS3ObjectLambd...	AWS managed	None	Provides AWS Lambda functions permi...
<input type="checkbox"/>	AmazonS3OutpostsFull...	AWS managed	None	Provides full access to Amazon S3 on ...
<input type="checkbox"/>	AmazonS3OutpostsRea...	AWS managed	None	Provides read only access to Amazon S...
<input checked="" type="checkbox"/>	AmazonS3ReadOnlyAccess	AWS managed	None	Provides read only access to all bucket...
<input type="checkbox"/>	AmazonS3TablesFullAcc...	AWS managed	None	Provides full access to all S3 table buc...
<input type="checkbox"/>	AmazonS3TablesReadQ...	AWS managed	None	Provides read only access to all S3 tabl...
<input type="checkbox"/>	AWSBackupServiceRole...	AWS managed	None	Policy containing permissions necessar...
<input type="checkbox"/>	AWSBackupServiceRole...	AWS managed	None	Policy containing permissions necessar...
<input type="checkbox"/>	AWSQuickSetupSSMDep...	AWS managed	None	This policy grants permissions for listi...
<input type="checkbox"/>	QuickSightAccessForS3S...	AWS managed	None	Policy used by QuickSight team to acc...
<input type="checkbox"/>	S3UnlockBucketPolicy	AWS managed	None	Provides access required to unlock a S...

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

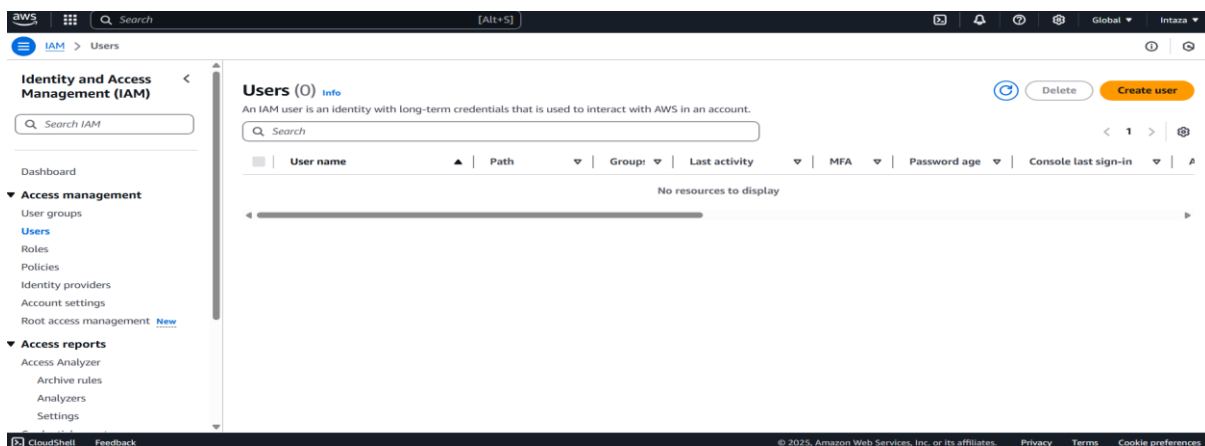
Click Create group



Step 3: Create an IAM User

In the left sidebar, click on “Users”

Click “Create/Add users”



Enter a username (e.g., intaza_user)

The screenshot shows the AWS IAM 'Create user' console. On the left, a progress bar indicates four steps: Step 1 (Specify user details), Step 2 (Set permissions), Step 3 (Review and create), and Step 4 (Retrieve password). The main area is titled 'Specify user details' and contains a 'User details' section. The 'User name' field is filled with 'intaza_b44'. Below it, a note states: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , @ _ - (hyphen)'. A checkbox labeled 'Provide user access to the AWS Management Console - optional' is checked. Below this, a light blue box titled 'Are you providing console access to a person?' contains two radio button options: 'Specify a user in Identity Center - Recommended' (unselected) and 'I want to create an IAM user' (selected). The 'I want to create an IAM user' option has a sub-note: 'We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.'

Set a custom password or auto-generate one

Uncheck "Require password reset"

Click Next

The screenshot shows the 'Console password' section of the 'Create user' console. It has two radio button options: 'Autogenerated password' (unselected) and 'Custom password' (selected). Below the 'Custom password' option is a text input field containing eight asterisks. To the right of the field, a note lists password requirements: 'Must be at least 8 characters long' and '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 ! @ # \$ % ^ & * () _ + = [] { } ' '. Below the input field is a checkbox labeled 'Show password' which is unchecked. Further down is another checkbox labeled 'Users must create a new password at next sign-in - Recommended' which is also unchecked. A light blue box at the bottom contains an information icon and text: '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](#)'. At the bottom right of the console, there are 'Cancel' and 'Next' buttons.

Step 4: Add User to Group

Select the group you just created (e.g., Developers)

aws [Search] [Alt+S] Global Intaza

IAM > Users > Create user

Step 1: Specify user details
Step 2: **Set permissions**
Step 3: Review and create
Step 4: 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.

User groups (1/1) [Create group](#)

Search

<input checked="" type="checkbox"/>	Group name	Users	Attached policies	Created
<input checked="" type="checkbox"/>	Developers	0	AmazonS3ReadOnlyAccess	2025-04-05 (4 minutes ago)

► **Set permissions boundary - optional**

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

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click Next and then Create User

aws [Search] [Alt+S] Global Intaza

IAM > Users > Create user

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

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name intaza_b44	Console password type Custom password	Require password reset No
-------------------------	--	------------------------------

Permissions summary < 1 >

Name	Type	Used as
Developers	Group	Permissions group

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](#)

aws [Search] [Alt+S] Global Intaza

IAM > Users

Identity and Access Management (IAM)

Search IAM

Dashboard

▼ **Access management**

- User groups
- Users**
- Roles
- Policies
- Identity providers
- Account settings
- Root access management [New](#)

▼ **Access reports**

- Access Analyzer
- Archive rules
- Analysers
- Settings

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](#)

Users (1) Info [Delete](#) [Create user](#)

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Search

<input type="checkbox"/>	User name	Path	Group	Last activity	MFA	Password age	Console last sign-in
<input type="checkbox"/>	intaza_b44	/	1	-	-	∞	-

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Identity and Access Management (IAM)

Dashboard

▼ Access management

Users

Roles

Policies

Identity providers

Account settings

Root access management [New](#)

▼ Access reports

Access Analyzer

Archive rules

Analyzers

Settings

IAM > Users > intaza_b44

intaza_b44 info

Delete

Summary

ARN
[arn:aws:iam::699475959073:user/intaza_b44](#)

Created
April 05, 2025, 15:32 (UTC+05:30)

Console access
[Enabled without MFA](#)

Last console sign-in
[Never](#)

Access key 1
[Create access key](#)

Permissions

Groups (1)

Tags

Security credentials

Last Accessed

Permissions policies (1)

[Refresh](#) [Remove](#) [Add permissions](#)

Permissions are defined by policies attached to the user directly or through groups.


Search

Filter by Type

All types

< 1 >

<input type="checkbox"/>	Policy name	Type	Attached via
<input type="checkbox"/>	AmazonS3ReadOnlyAccess	AWS managed	Group Developers



Sign in as IAM user

Account ID (12 digits) or account alias

IAM user name

Password

☐ Remember this account

[Sign in](#)

[Sign in using root user email](#)

[Forgot password?](#)

Unlock healthcare innovation in the cloud

Learn how healthcare organizations improve outcomes and enhance patient experiences

[Get started »](#)

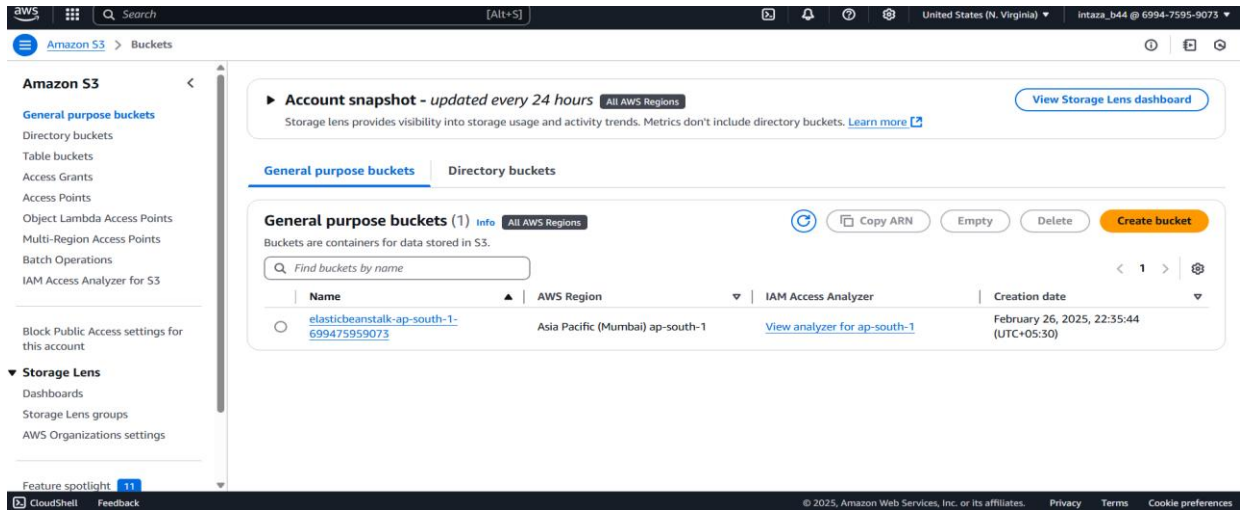
Check if account permissions are working properly in different tools:

The screenshot shows the AWS Console Home page. The 'Recently visited' section lists services like IAM, Elastic Beanstalk, S3, EC2, Key Management Service, AWS Private Certificate Authority, Amazon Lex, and Billing and Cost Management. The 'Applications' section shows a table with one entry: 'Access denied to servicecatalog:ListApplications'. A red box highlights this error, and a 'Diagnose with Amazon Q' button is visible. The 'Welcome to AWS', 'AWS Health', and 'Cost and usage' sections are also visible at the bottom.

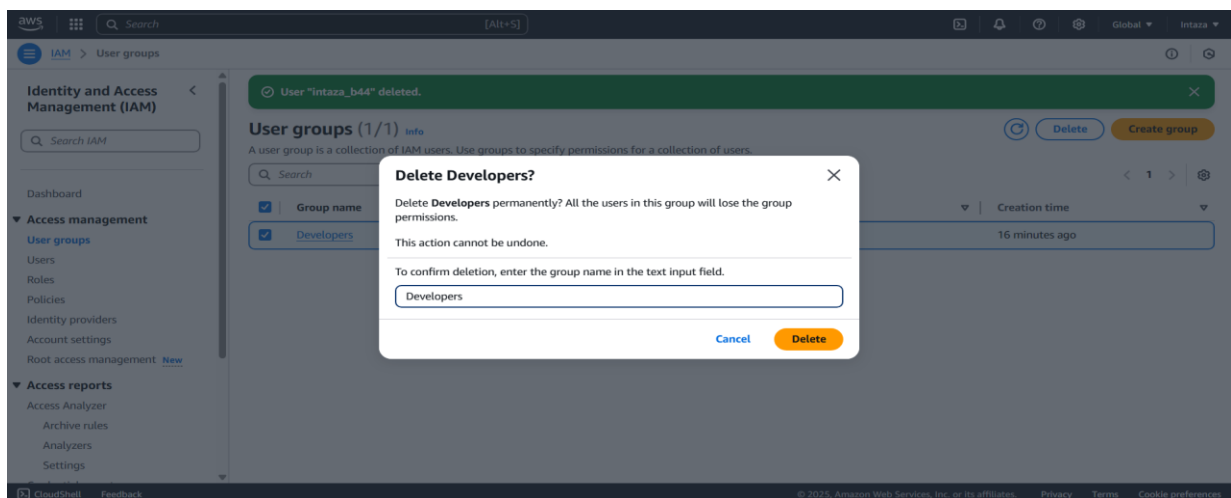
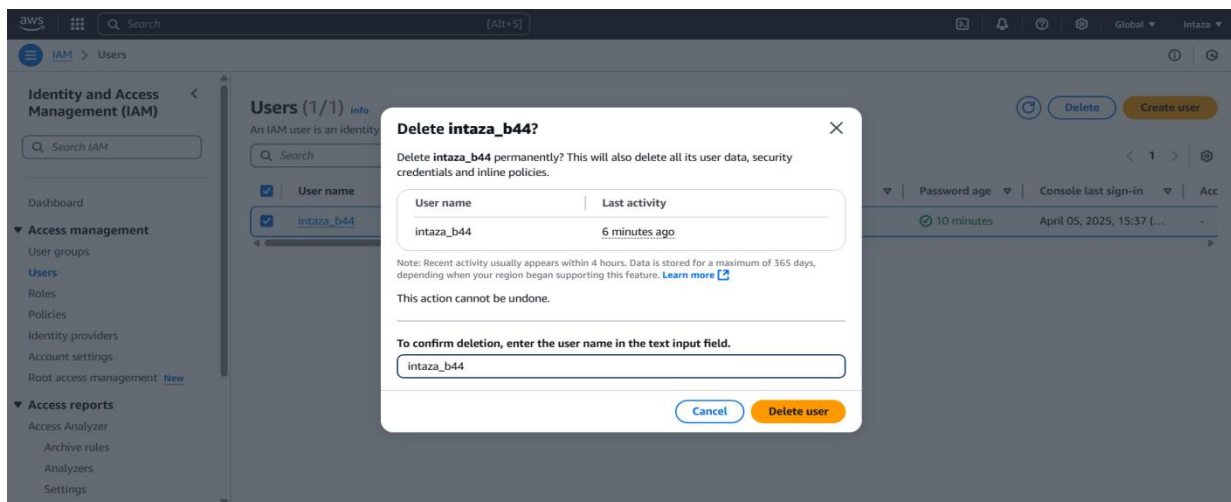
The screenshot shows the AWS EC2 console. The 'Resources' section displays a table of EC2 resources with columns for Instances (running), Auto Scaling Groups, Capacity Reservations, Dedicated Hosts, Elastic IPs, Load balancers, Key pairs, Snapshots, Placement groups, and Volumes. The 'Launch instance' section provides instructions on how to launch an Amazon EC2 instance. The 'Service health' section shows an error: 'An error occurred: An error occurred retrieving service health information'. The 'EC2 Free Tier' section displays the end of month forecast and account attributes.

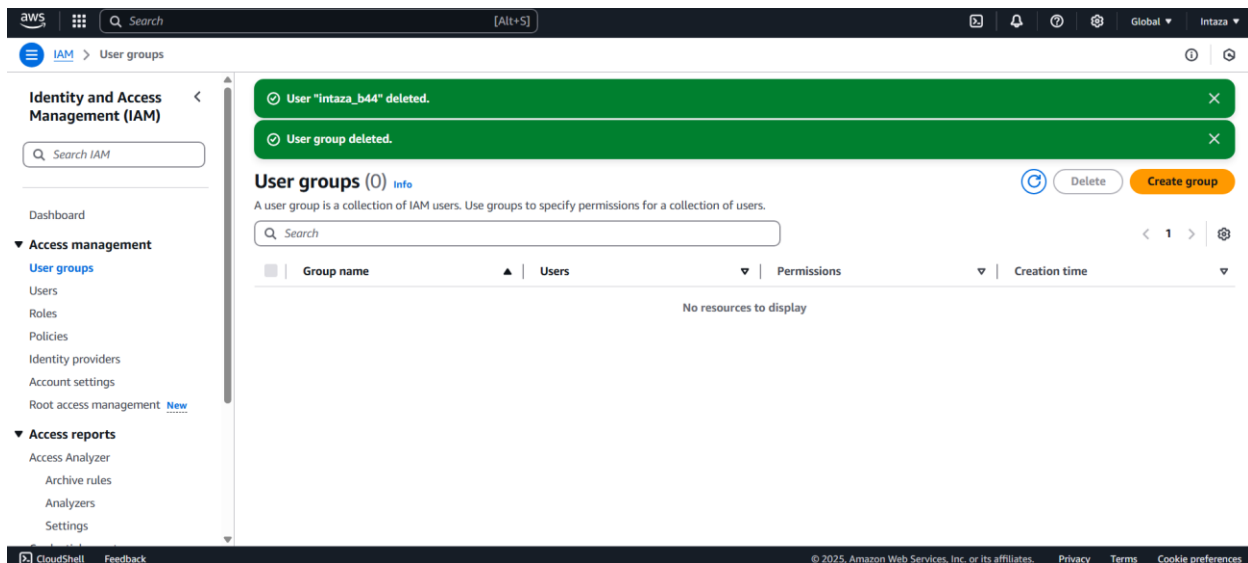
The screenshot shows the AWS IAM console. The 'Security recommendations' section displays two 'Access denied' errors. The first error states: 'You don't have permission to iam:GetAccountSummary. To request access, copy the following text and send it to your AWS administrator. Learn more about troubleshooting access denied errors.' The second error states: 'You don't have permission to iam:ListMFADevices. To request access, copy the following text and send it to your AWS administrator. Learn more about troubleshooting access denied errors.' The 'AWS Account' section shows an 'Access denied' error: 'You don't have permission to iam:ListAccountAliases. To request access, copy the following text and send it to your AWS administrator. Learn more about troubleshooting access denied errors.' The 'Quick Links' section provides links to 'My security credentials' and 'Manage your access keys, multi-factor authentication (MFA) and other credentials.'

The only access we provided:



Step 6: Delete User group and Users:





Q.3 Explain various parameters to measure security of web server?

Ans:

The security of a web server can be measured by evaluating the following parameters:

Access Control & IAM: Ensuring only authorized users and services can access the server.

Firewall Protection (e.g., AWS WAF): Prevents attacks like SQL injection, XSS.

SSL/TLS Encryption: Ensures data is encrypted in transit.

Security Groups (AWS): Acts like a virtual firewall for EC2 instances.

Logging and Monitoring: Tools like CloudWatch and CloudTrail help track access and detect suspicious activity.

Patching and Updates: The server OS and web applications must be regularly updated.

Multi-Factor Authentication: Adds an extra layer of login security.

Backup and Recovery: Regular backups to recover data in case of an attack.

Role-based Access Control (RBAC): Permissions based on the job role.

B.3 Conclusion:

Through this experiment, I learned how to create IAM users and manage access to AWS resources securely. I understood the importance of separating the root user from daily operations, creating user groups, and assigning policies. I also understood how IAM roles can help provide temporary and secure access to AWS resources. This practical helped me analyze the security aspects of cloud infrastructure and manage user access effectively, meeting the objective of understanding cloud security.