Terna Engineering College

**Computer Engineering Department**

Program: Sem VI

**Course: Cloud Computing Lab(CSL605)**

# Aim:

## PART A

(PART A: TO BE REFFERED BY STUDENTS)

Experiment No.8

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

* 1. **Prerequisite:**

Knowledge of Access Control, Authentication and Authorization

* 1. **Objective**:

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

* 1. **Outcome: (LO 4)**

After successful completion of this experiment student will be able to

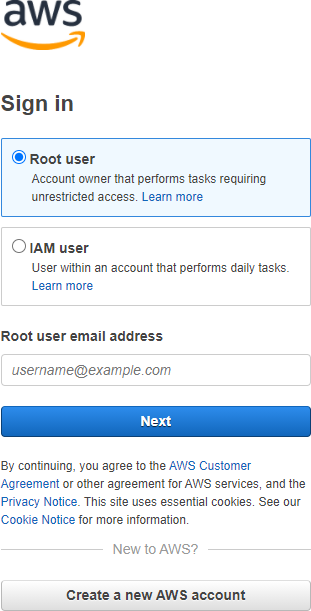
* + - Analyze security issues on cloud
  1. **Theory:**

# 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.



**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)

1. **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.

1. **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.

1. **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.

1. **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)**

1. **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.

1. **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.

1. **Integrated with many AWS services**

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

1. **Eventually Consistent**

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

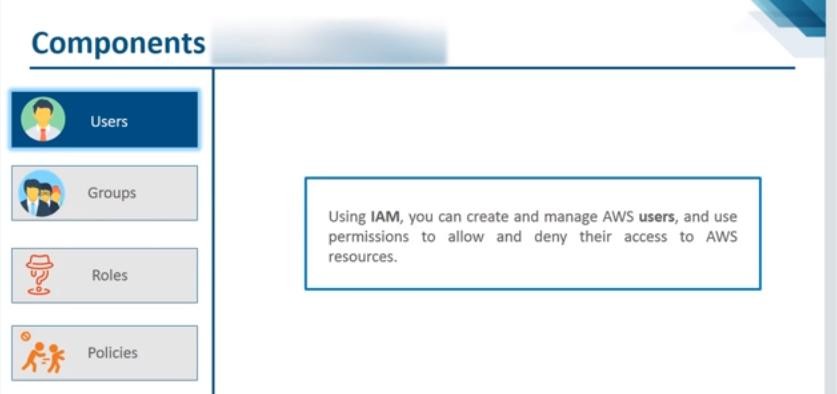
If a request to change some data is successful, the change is committed and safely stored. However, the change must be replicated across IAM, which can take some time. Such changes include creating or updating users, groups, roles, or policies. We recommend that you do not include such IAM changes in 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).

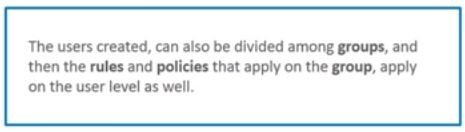
1. **Free to use**

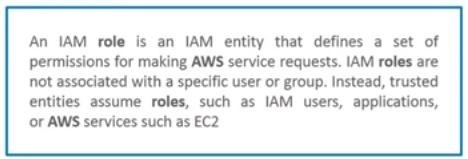
AWS Identity and Access Management (IAM) and AWS Security Token Service (AWS STS) are features of your AWS account offered at no additional charge. You are charged only when you access other AWS services using your IAM users or AWS STS temporary security credentials. For information about the pricing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

# Components of IAM

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

****

****



**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).

1. **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 Power Shell. 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 Power Shell, see the AWS Tools for Windows Power Shell User Guide.

1. **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 SDKsprovide 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.

## 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 the there is no ERP access available)***

|  |  |
| --- | --- |
| Roll No. B30 | Name: Pranjal Bhatt |
| Class :TE B COMPS | Batch :B2 |
| Date of Experiment: | Date of Submission: |
| Grade : |  |

* 1. **Question of Curiosity:**
  2. ***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.

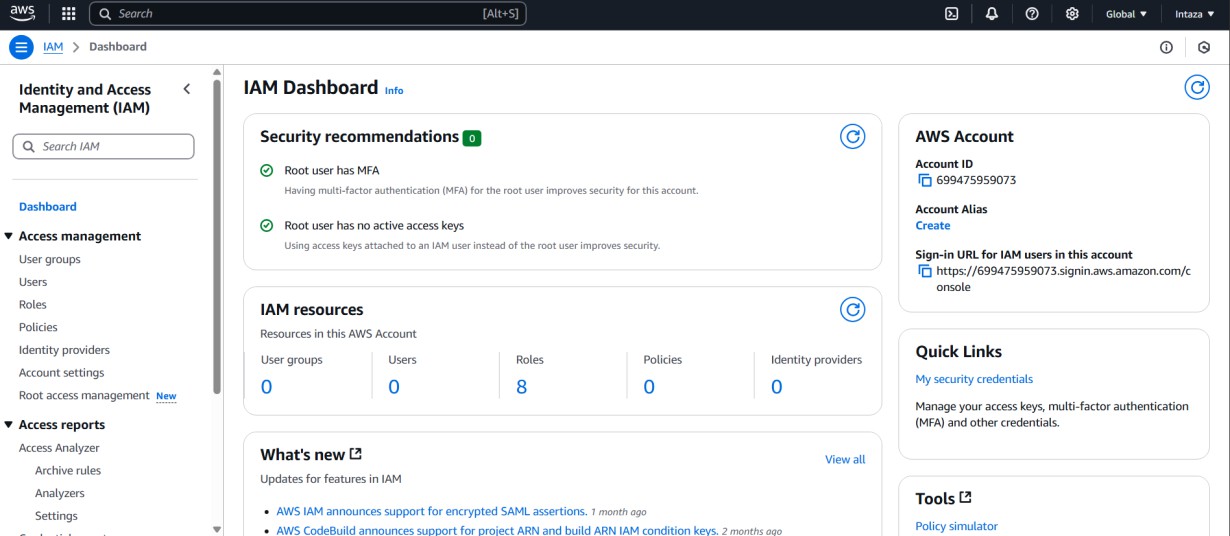
Assign granular permissions.

Use MFA for stronger authentication. Audit activity through AWS CloudTrail.

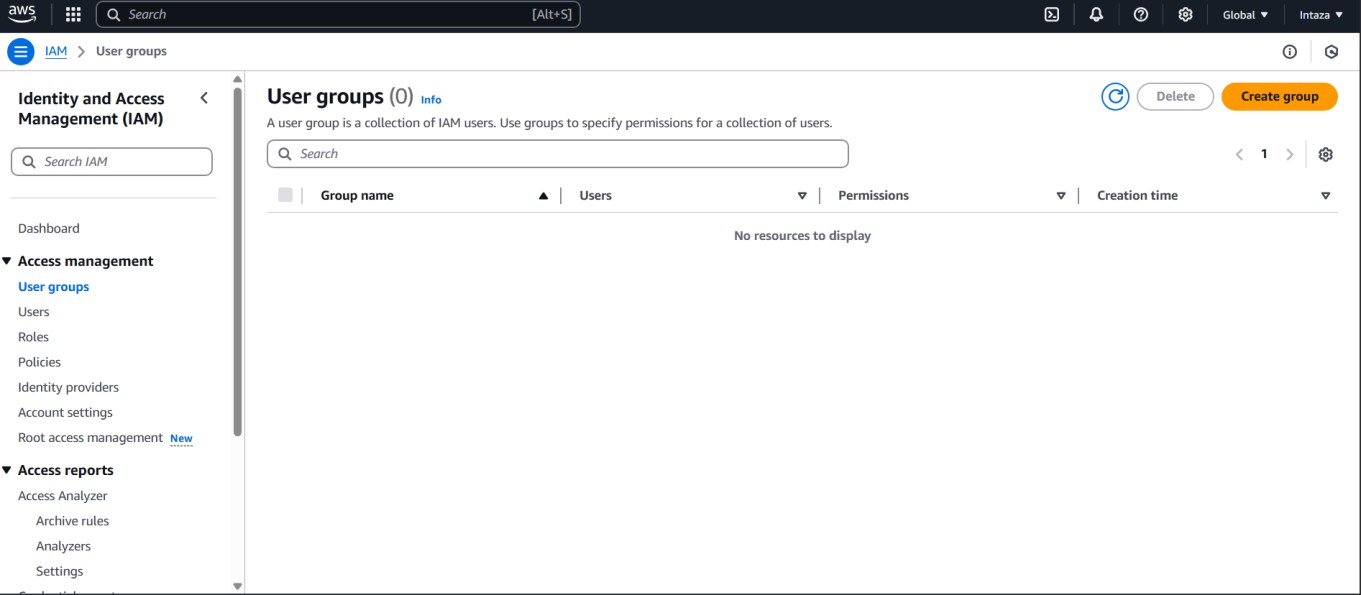
* 1. ***Add snapshots for creating IAM user and user groups (using AWS IAM service).***

**Step 1: Go to IAM Service Open** [**AWS Console**](https://console.aws.amazon.com/)

**Search for IAM in the search bar and click it.**

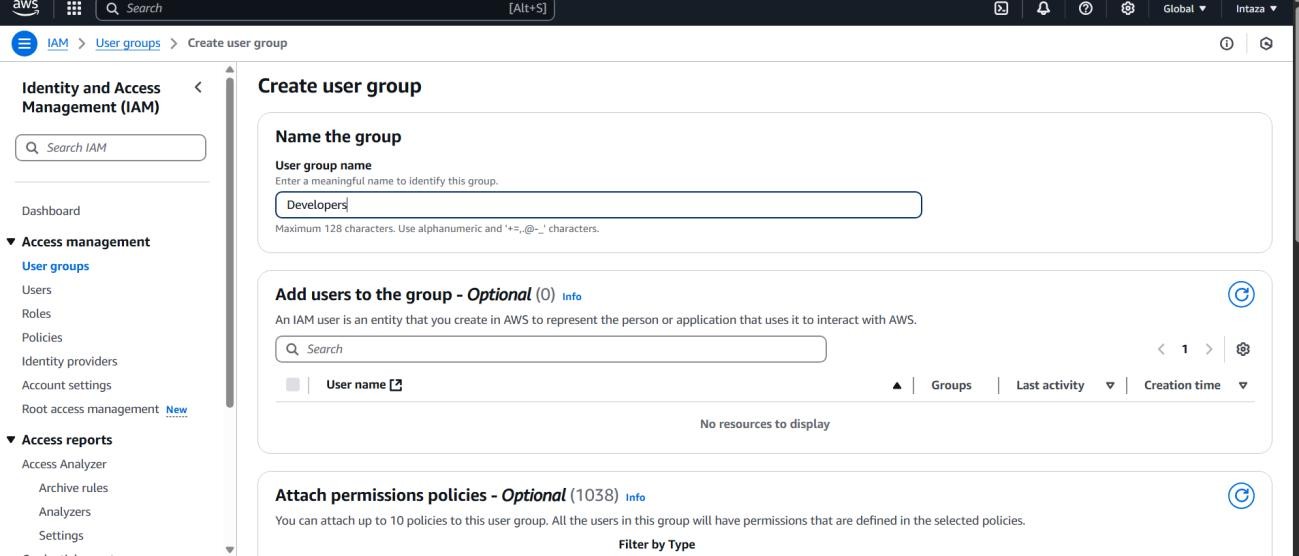
****

**Step 2: Create a User Group**

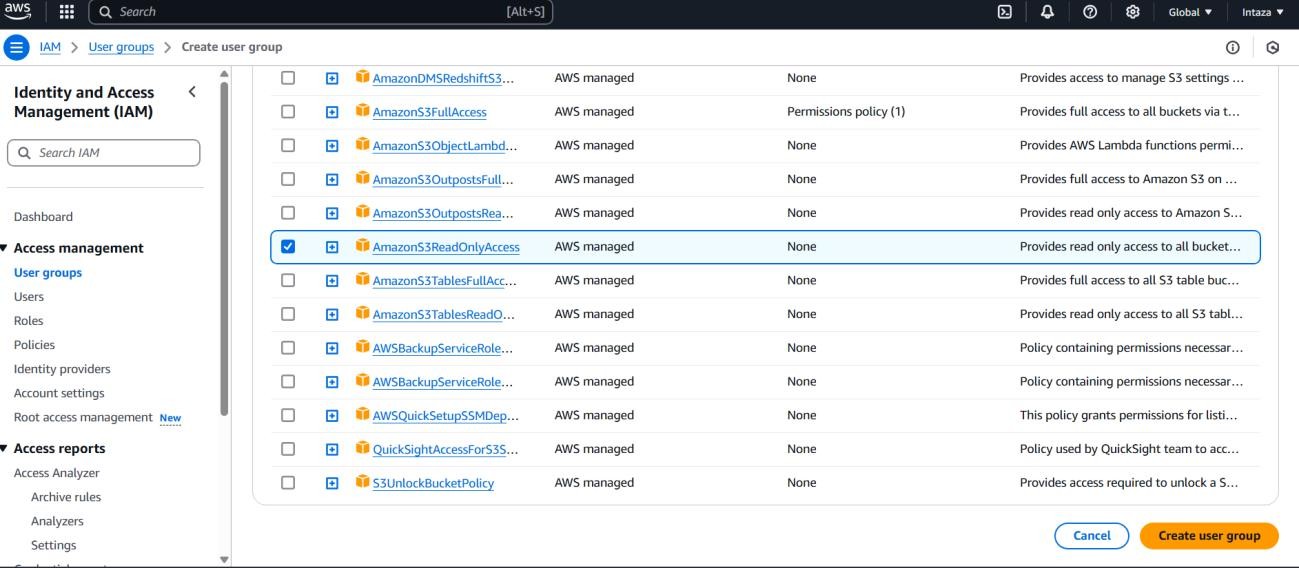
****

**In the left sidebar, click on “User groups”**

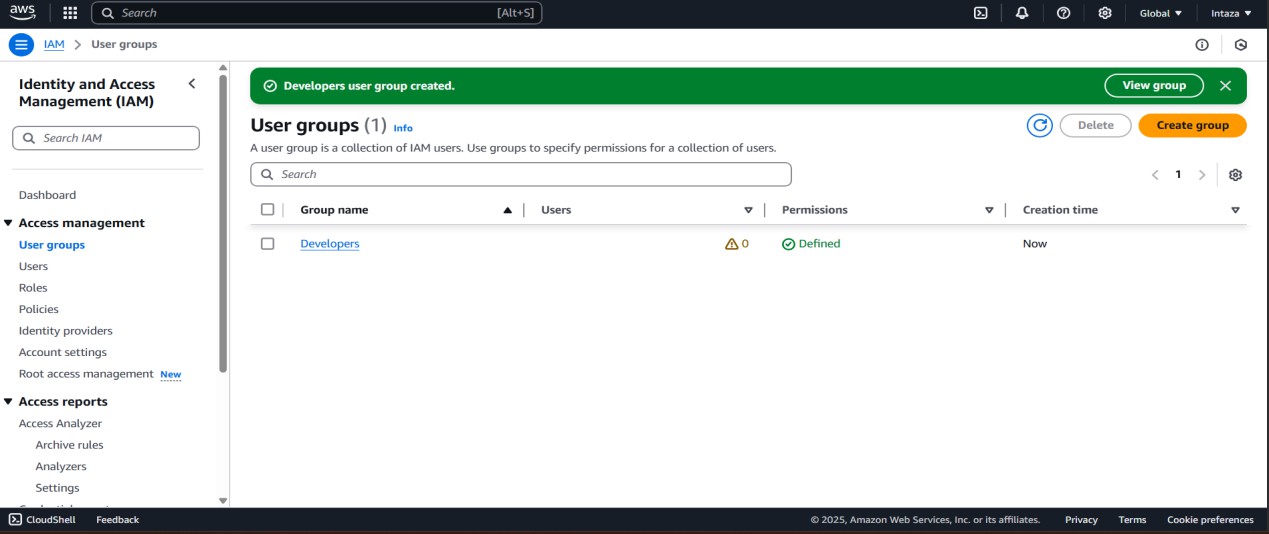
**Enter a name (e.g., Developers, HR\_Team)**

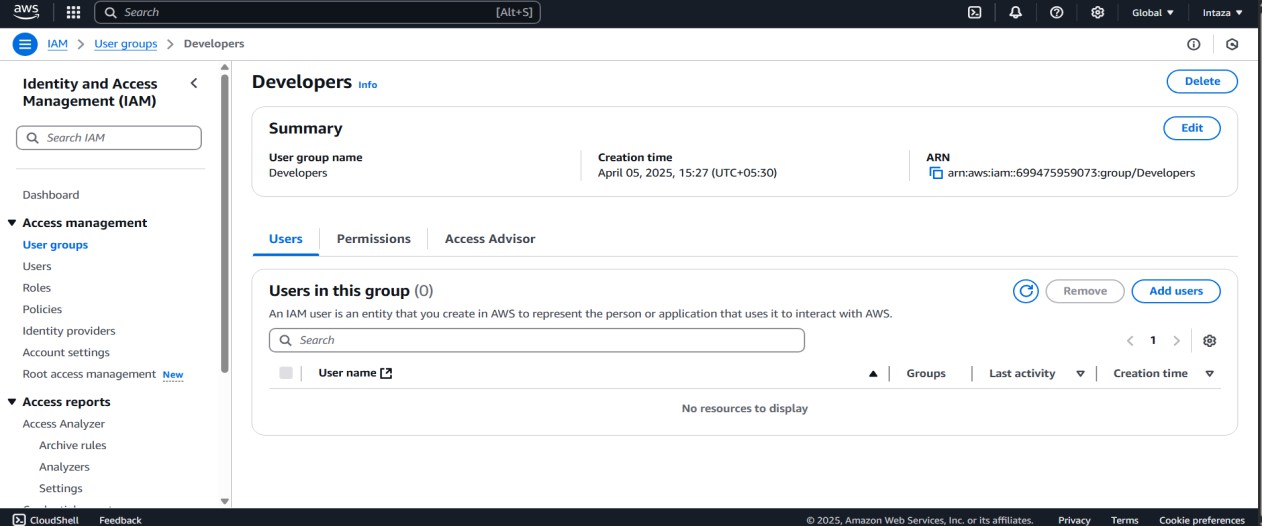
****

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

****

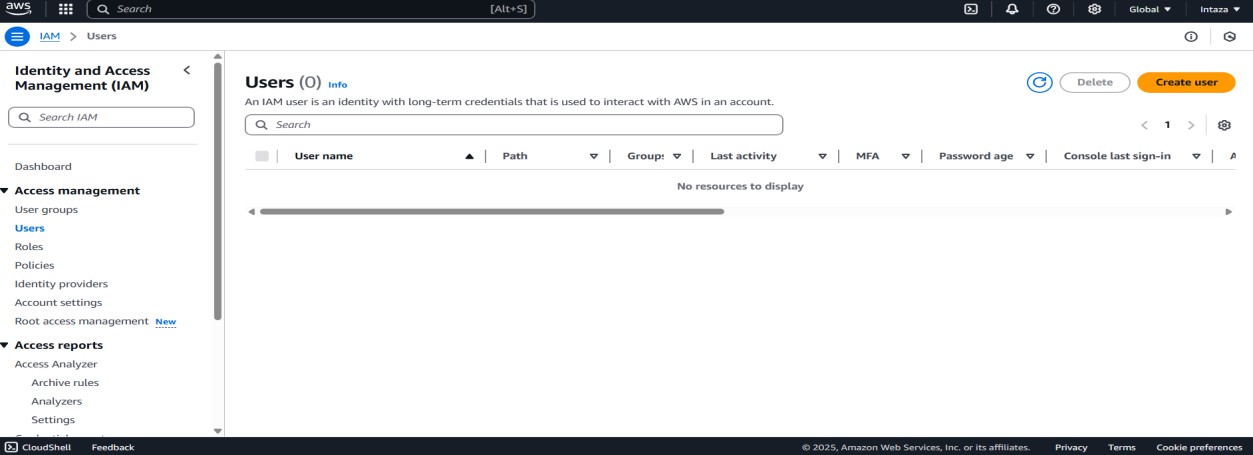
**Click Create group**



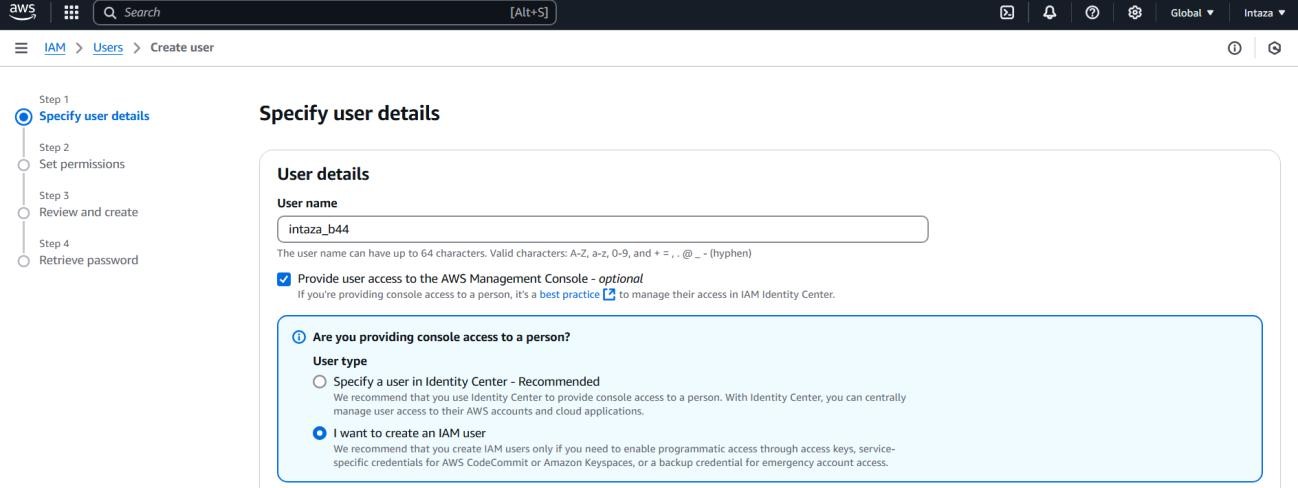
****

**Step 3: Create an IAM User**

**In the left sidebar, click on “Users” Click “Create/Add users”**



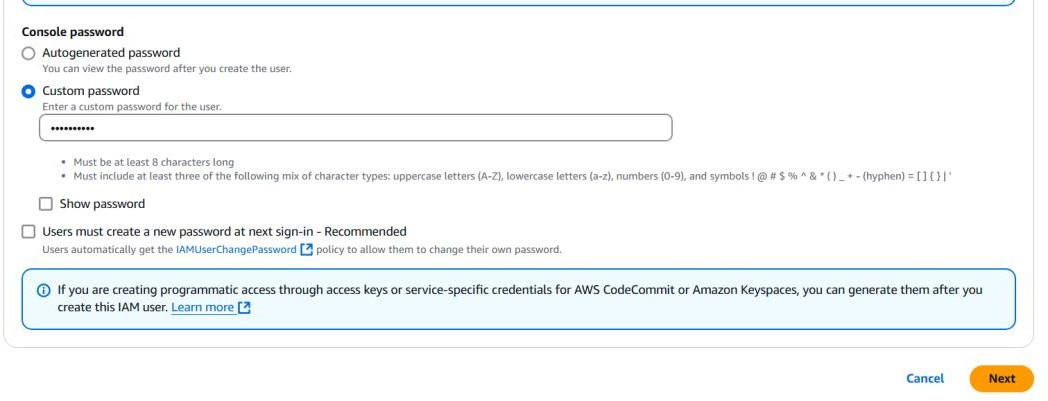
**Enter a username (e.g., intaza\_user)**

****

**Set a custom password or auto-generate one**

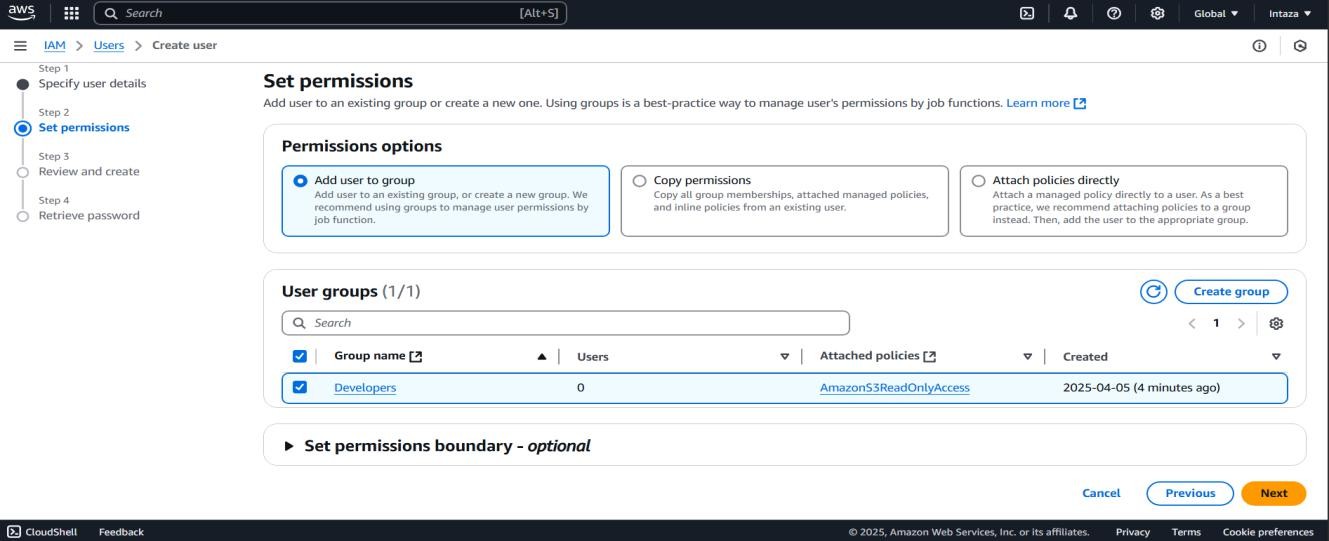
**Uncheck “Require password reset”**

**Click Next**

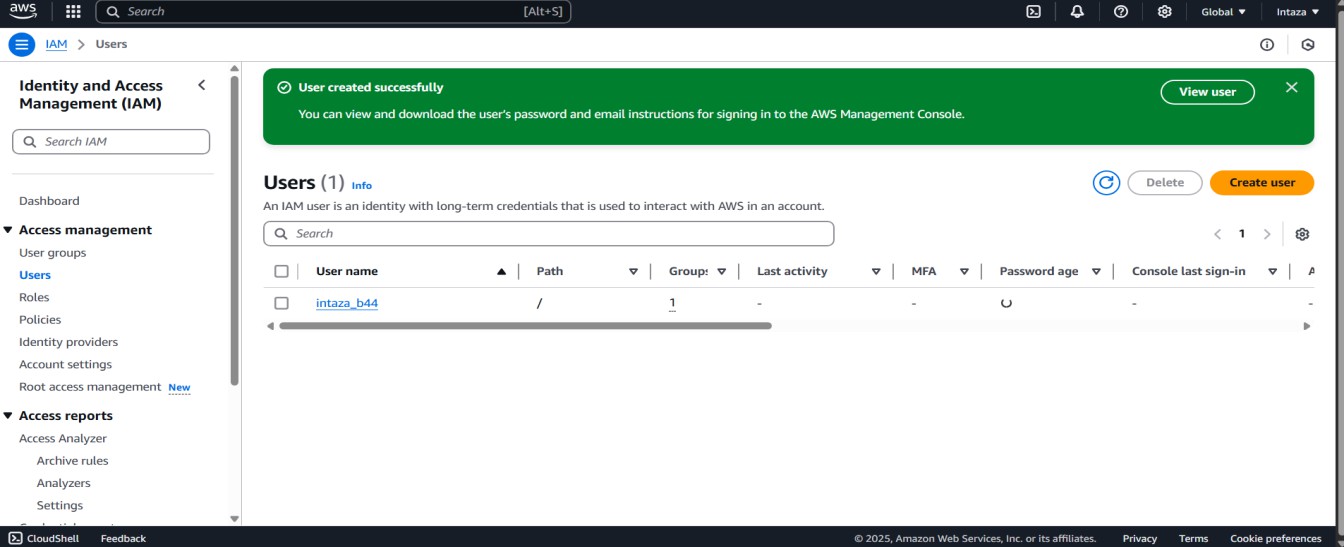
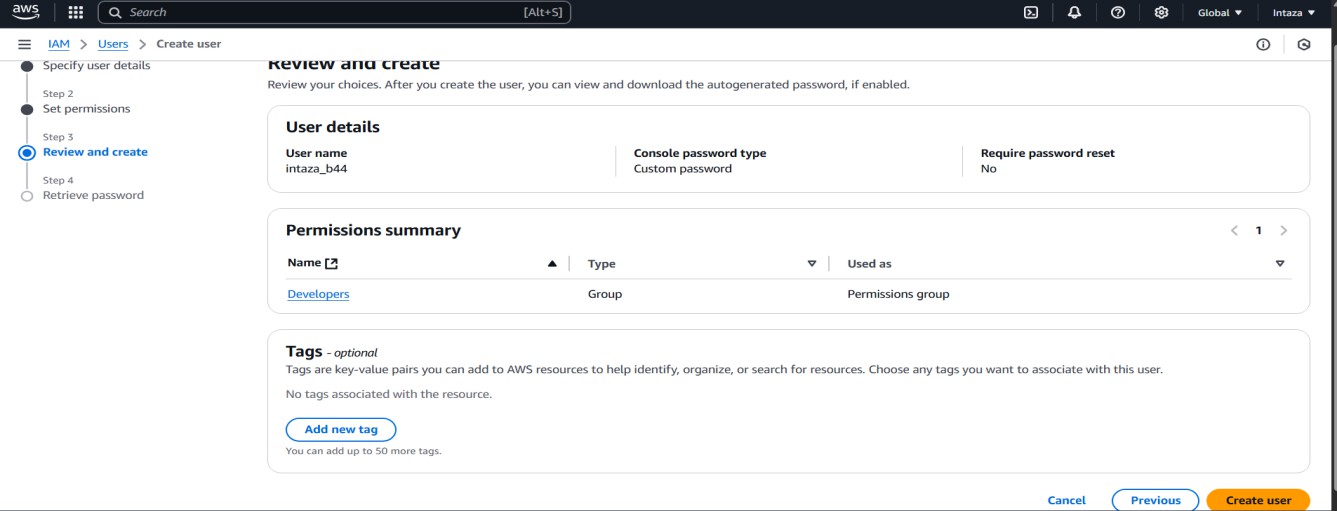
****

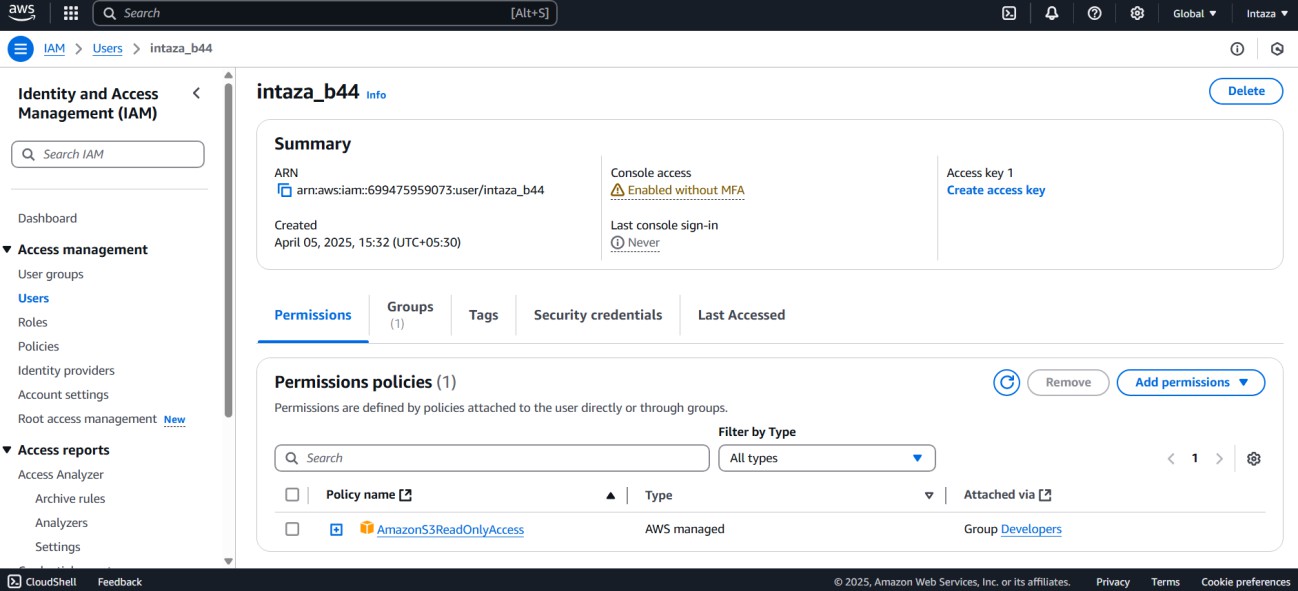
**Step 4: Add User to Group**

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

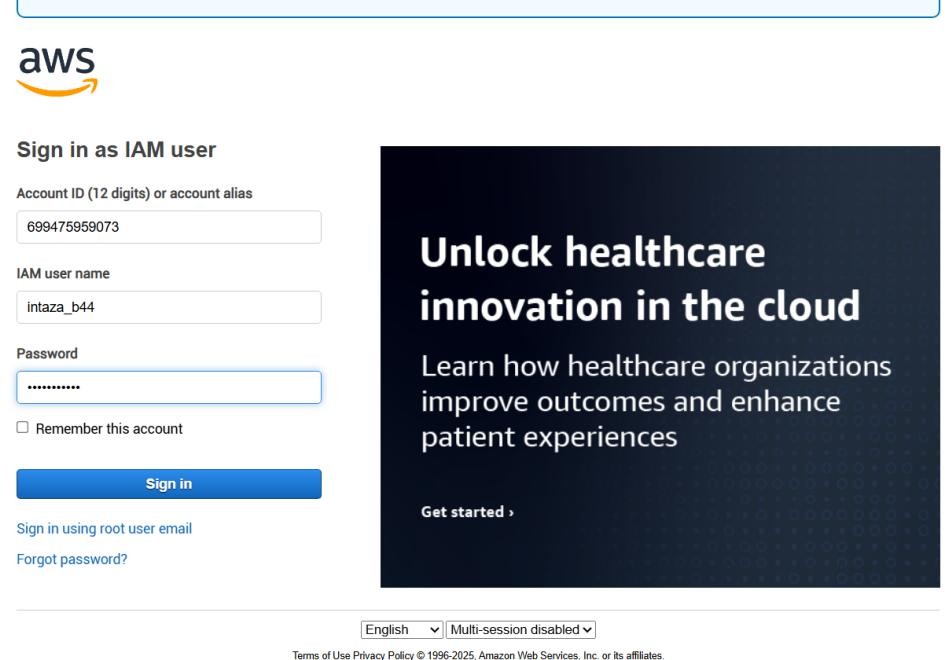


**Click Next and then Create User**

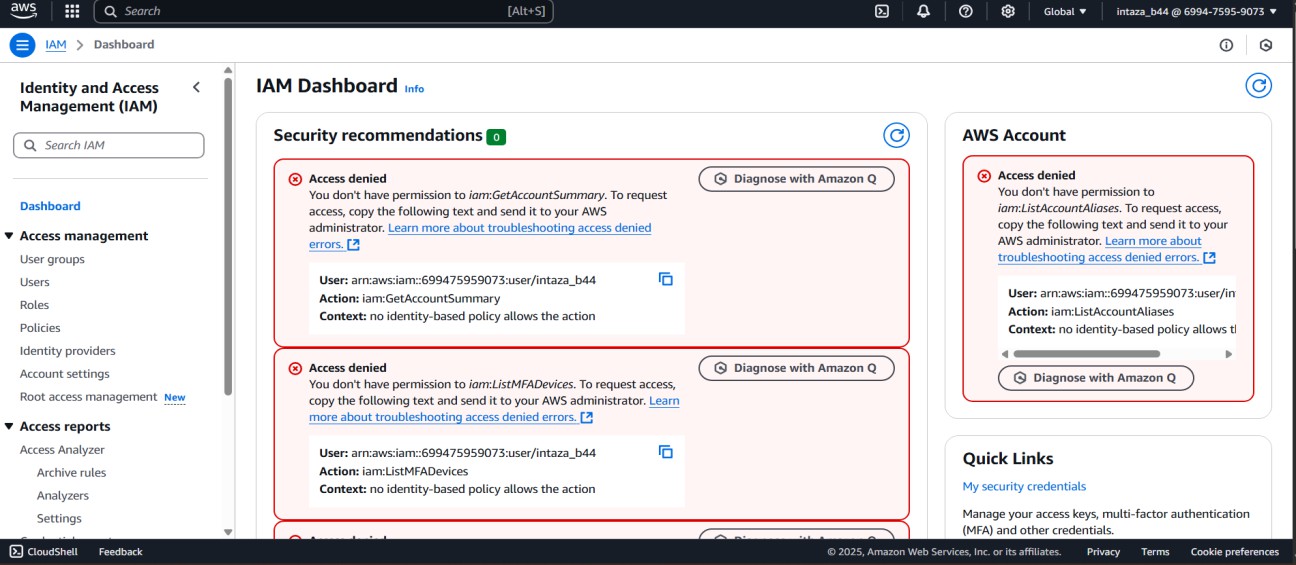
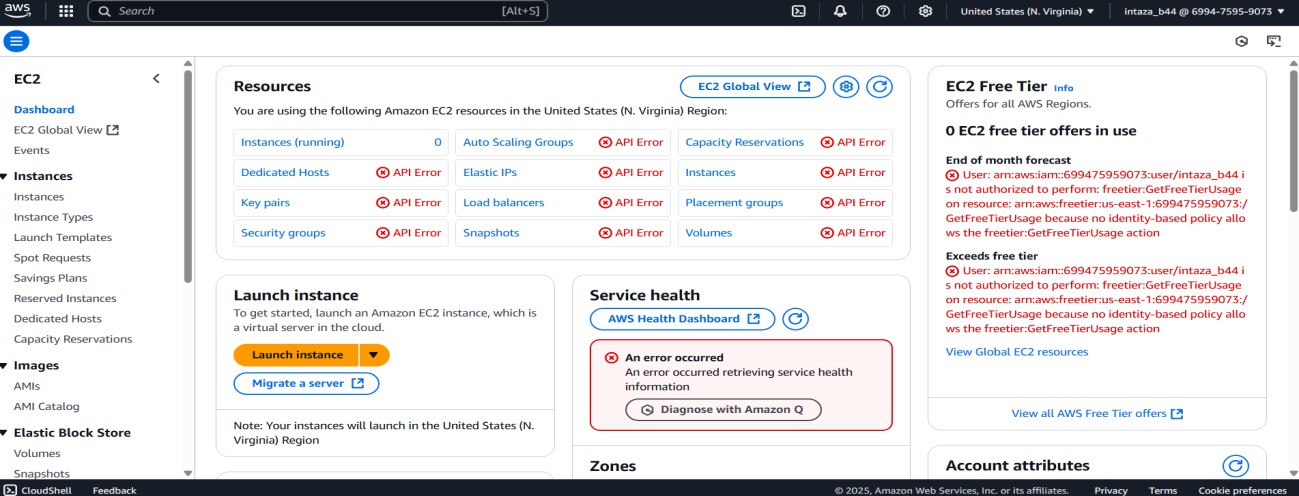
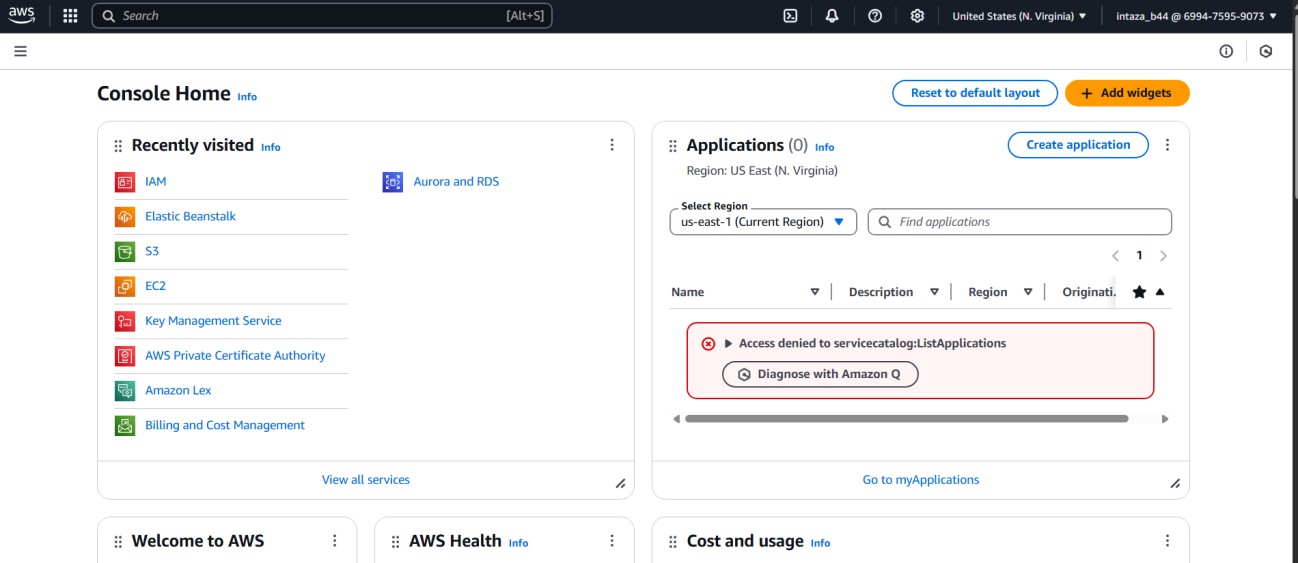
****



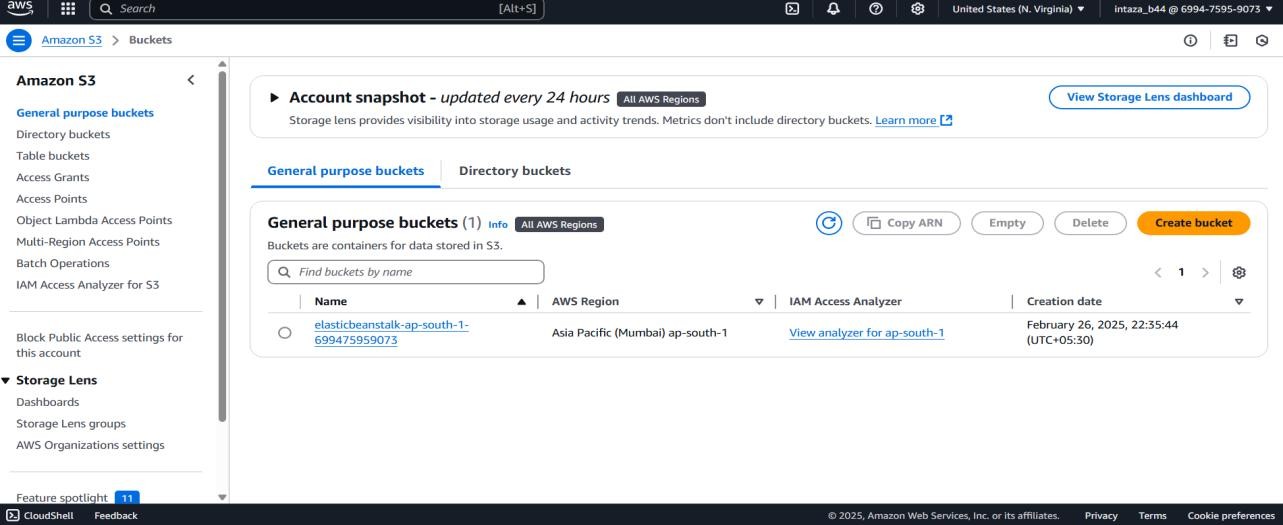
**Step 5: Sign-in with the IAM User**

****

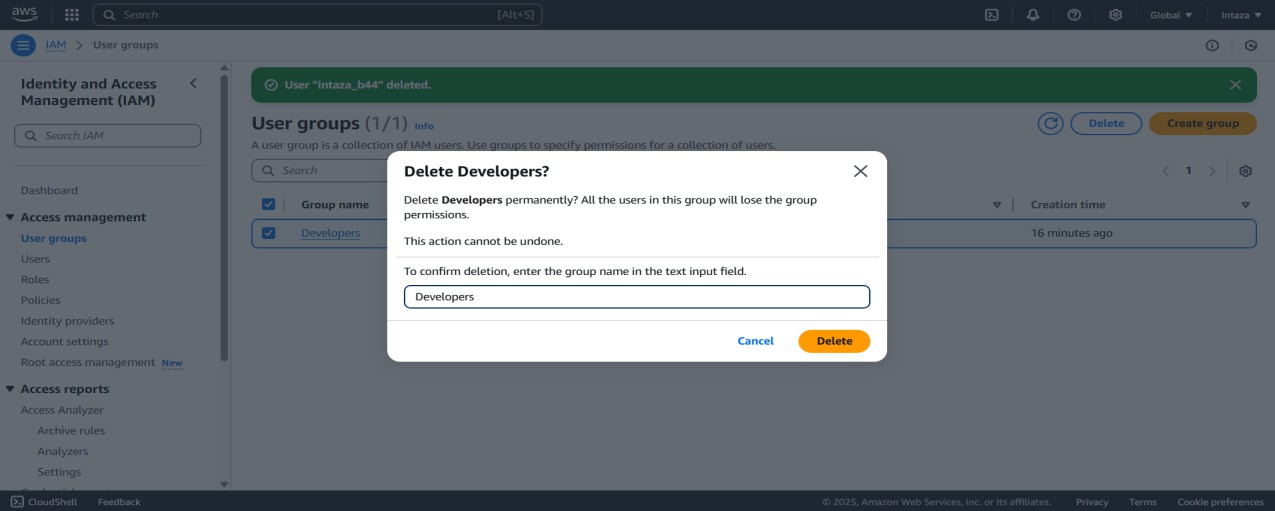
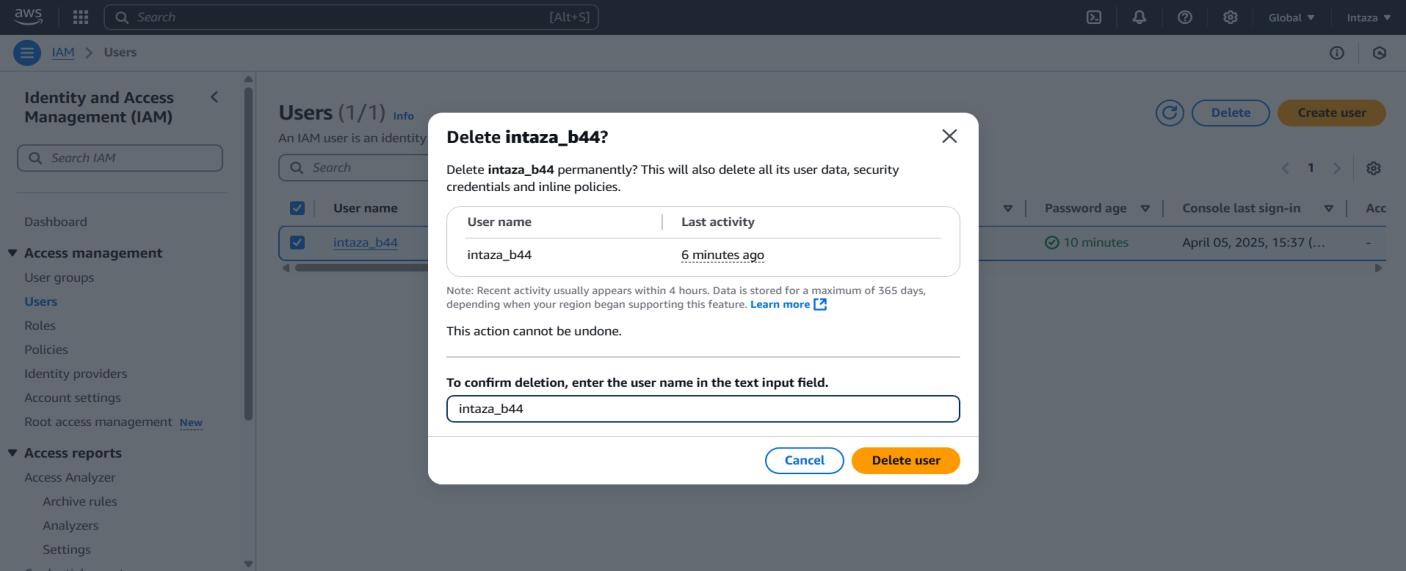
**Check if account permissions are working properly in different tools:**

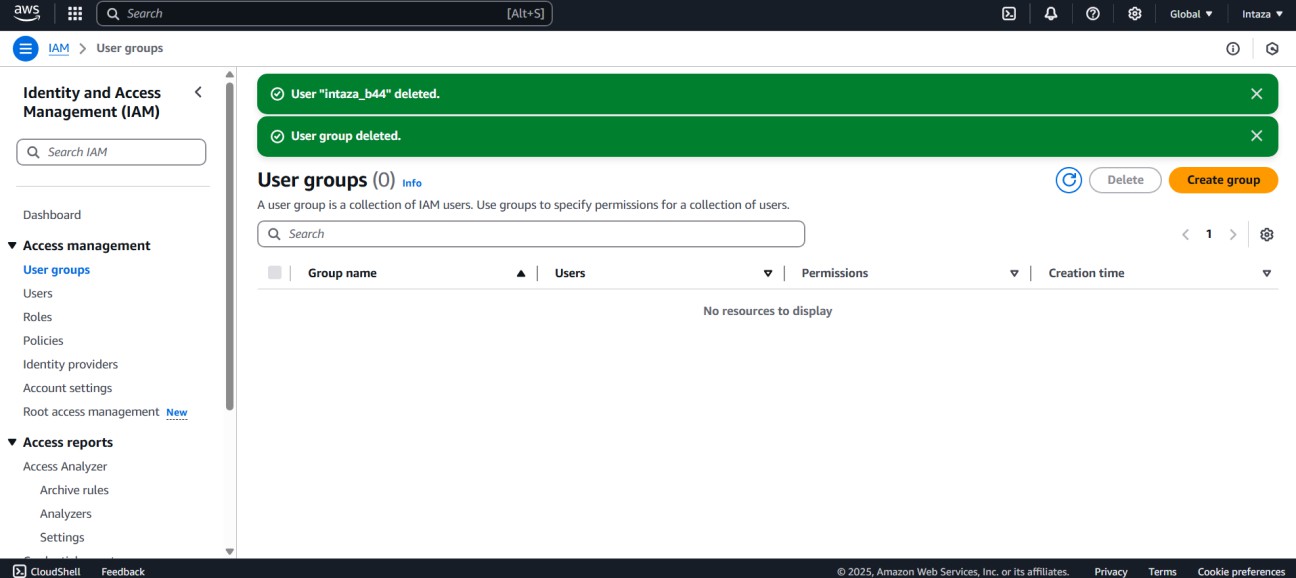
****

**The only access we provided:**

****

**Step 6: Delete User group and Users:**

****



* 1. ***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.