

# AWS Identity and Access Management (IAM)

## 1. Introduction

AWS Identity and Access Management (IAM) is a critical security service in AWS that allows organizations to control access to AWS resources securely. With IAM, administrators can create users, groups, policies, and roles, assigning permissions based on the principle of least privilege, ensuring users can only access what is necessary for their job.

In this lab, we explored pre-created IAM users and groups, examined the attached managed and inline policies, assigned users to groups according to job roles, and tested their access through the IAM sign-in URL. This provides hands-on understanding of how IAM manages secure access in real AWS environments.

## 2. Lab Objectives

- To understand IAM user and group management.
- To explore managed and inline IAM policies.
- To assign users to appropriate groups based on business roles.
- To verify access restrictions through real-time testing using login URLs.
- To observe how permission boundaries affect AWS service access.

## 3. AWS IAM Key Concepts (Explanation)

**a) IAM Users:** IAM Users represent individual identities that can authenticate into AWS using passwords or access keys.

**b) IAM Groups:** IAM Groups are collections of users, allowing permissions to be assigned collectively instead of individually.

**c) IAM Policies:** Policies contain permission rules in JSON format which define:

- **Effect** – Allow or Deny
- **Action** – What operations are permitted (example: s3:ListBucket)
- **Resource** – Which AWS resource(s) the policy applies to (\* or ARN)

Policies are of two types:

- **Managed Policy** – Prebuilt by AWS or admins, reusable across users and groups.
- **Inline Policy** – Attached directly to a single user/group for unique permission use cases.

**d) IAM Roles:** A temporary access identity used by services, applications, or federated users (not used in this lab but important conceptually).

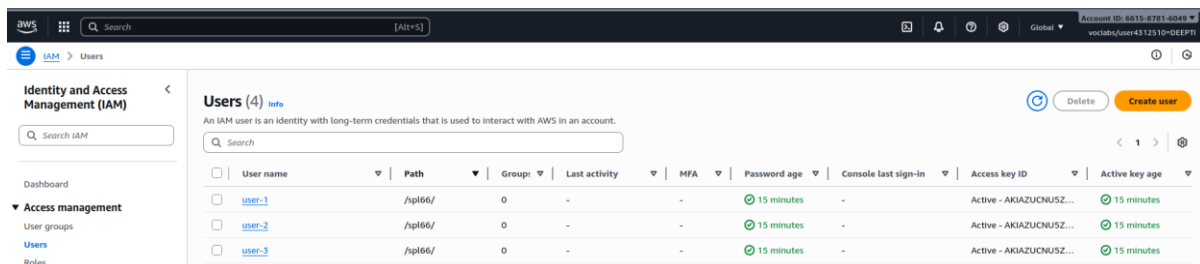
## 4. Business Scenario Summary

User	Assigned Group	Permissions Access
user-1	S3-Support	Read-Only access to Amazon S3
user-2	EC2-Support	Read-Only access to Amazon EC2
user-3	EC2-Admin	View, Start, Stop Amazon EC2 instances

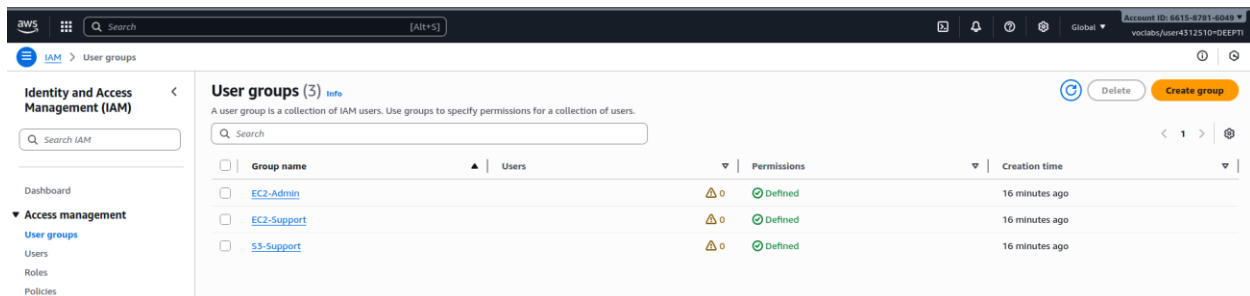
## 5. Task-Wise Lab Explanation

### Task 1: Explore Users and Groups

- Open IAM from AWS console.
- Observed three pre-created users: user-1, user-2, user-3.



- Verified that user-1 initially had no permissions and no group membership.
- Observed pre-created groups: EC2-Admin, EC2-Support, S3-Support.



Checked permissions for each group:

- EC2-Support → AmazonEC2ReadOnlyAccess (Managed Policy)
- S3-Support → AmazonS3ReadOnlyAccess (Managed Policy)
- EC2-Admin → Inline Policy allowing Describe, Start, Stop EC2 instances

### Task 2: Add Users to Groups

Based on business requirements:

- Added user-1 to S3-Support

AWS IAM console screenshot showing the 'S3-Support' user group. The left sidebar shows the 'Identity and Access Management (IAM)' menu with options like Dashboard, Access management, User groups, Roles, Policies, Identity providers, Account settings, Root access management, Access reports, Access Analyzer, Resource analysis, Unused access, Analyzer settings, Credential report, Organization activity, Service control policies, Resource control policies, IAM Identity Center, and AWS Organizations.

The main content area shows the 'S3-Support' user group summary. The 'Users' tab is selected, showing 'Users in this group (0)'. A search bar and a table with columns 'User name', 'Groups', 'Last activity', and 'Creation time' are visible. The table is empty, displaying 'No resources to display'.

AWS IAM console screenshot showing the 'S3-Support' user group after adding a user. A green notification banner at the top states '1 user added to this group.' The 'Users' tab now shows 'Users in this group (1)'. The table lists one user, 'user-1', with columns for 'User name', 'Groups', 'Last activity', and 'Creation time'.

- Added user-2 to EC2-Support

AWS IAM console screenshot showing the 'EC2-Support' user group. The left sidebar is the same as the previous screenshots. The main content area shows the 'EC2-Support' user group summary. The 'Users' tab is selected, showing 'Users in this group (0)'. A search bar and a table with columns 'User name', 'Groups', 'Last activity', and 'Creation time' are visible. The table is empty, displaying 'No resources to display'.

AWS IAM console screenshot showing the EC2-Support user group. A green notification bar at the top states "1 user added to this group." The summary section shows the user group name as EC2-Support, created on November 18, 2025, 13:51 (UTC+05:30), with ARN arn:aws:iam::661587816049:group/spl66/EC2-Support. The Users tab shows one user, user-2, in a table with columns for User name, Groups, and Last activity.

- Added user-3 to EC2-Admin

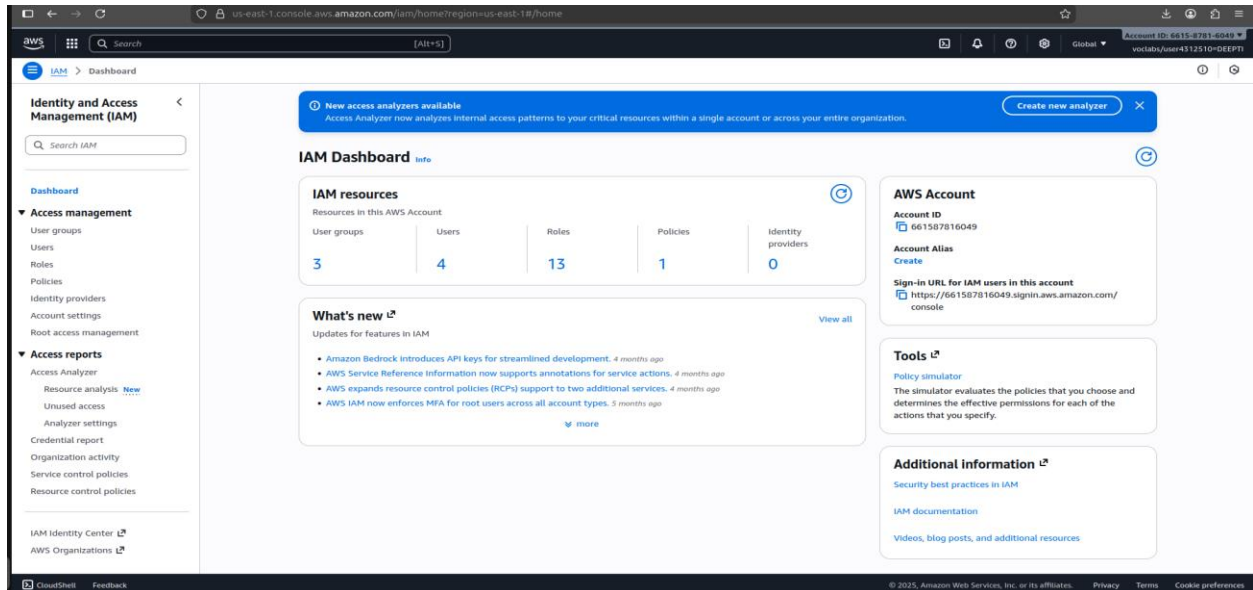
AWS IAM console screenshot showing the EC2-Admin user group. The summary section shows the user group name as EC2-Admin, created on November 18, 2025, 13:51 (UTC+05:30), with ARN arn:aws:iam::661587816049:group/spl66/EC2-Admin. The Users tab shows zero users, with a message "No resources to display" and a table with columns for User name, Groups, Last activity, and Creation time.

AWS IAM console screenshot showing the EC2-Admin user group after adding user-3. A green notification bar at the top states "1 user added to this group." The summary section shows the user group name as EC2-Admin, created on November 18, 2025, 13:51 (UTC+05:30), with ARN arn:aws:iam::661587816049:group/spl66/EC2-Admin. The Users tab shows one user, user-3, in a table with columns for User name, Groups, and Last activity.

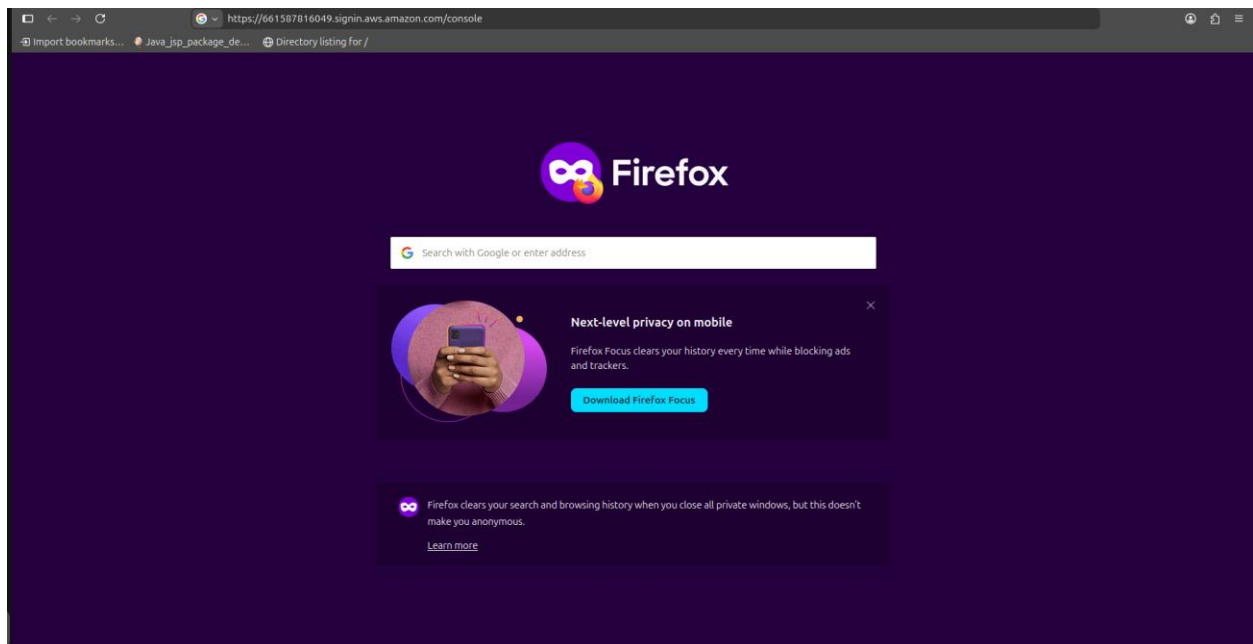
Finally verified that each group displayed 1 assigned user.

## Task 3: Sign-In and Permissions Testing

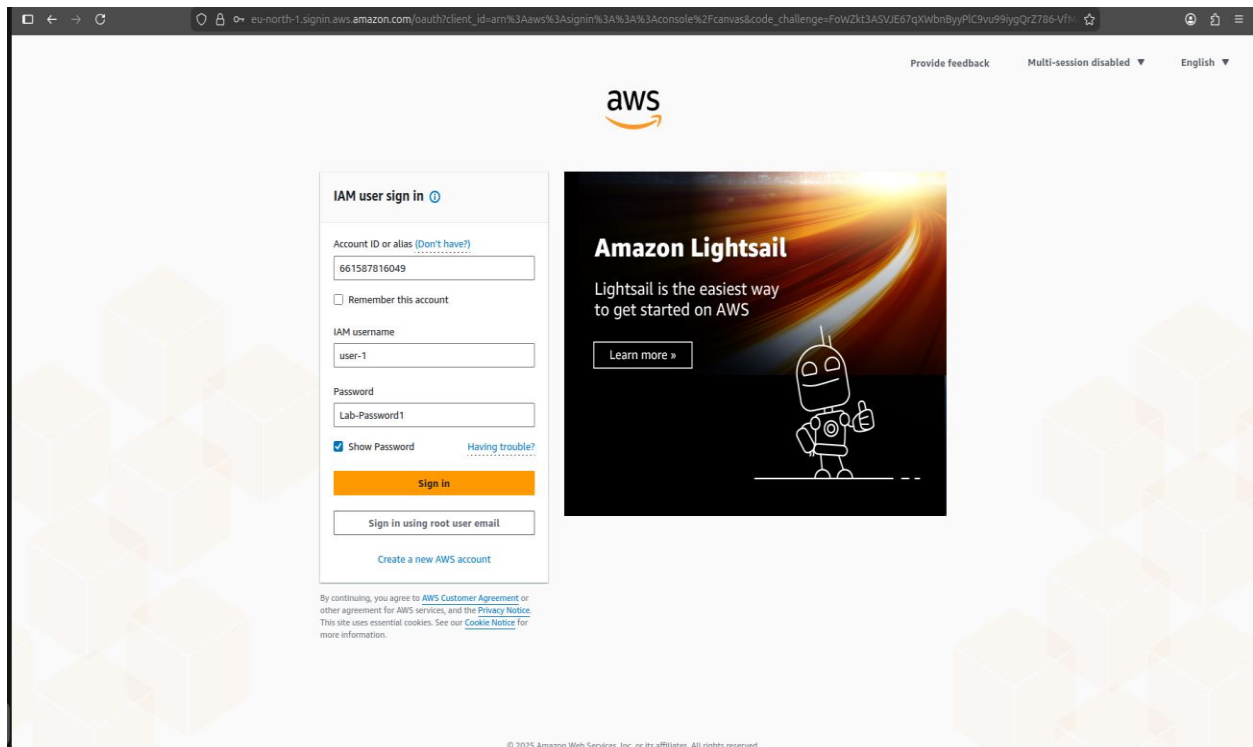
- In the navigation pane on the left, choose the Dashboard. On the right side, a Sign-in URL for IAM users in this account is displayed. The sign-in link shown in the lab environment was: <https://661587816049.signin.aws.amazon.com/console>



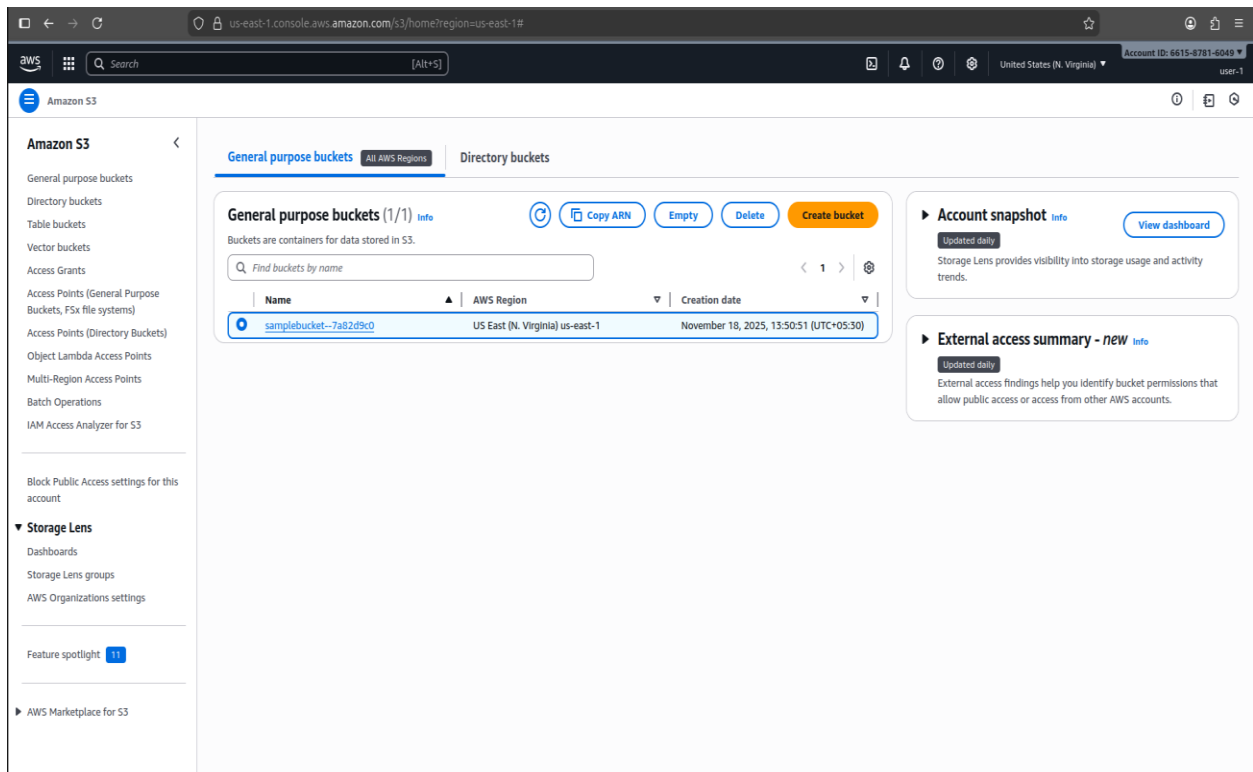
- This link can be used to sign-in to the AWS Account you are currently using. Copy the Sign-in URL for IAM users in this account to a text editor. Open a private (Incognito) window.



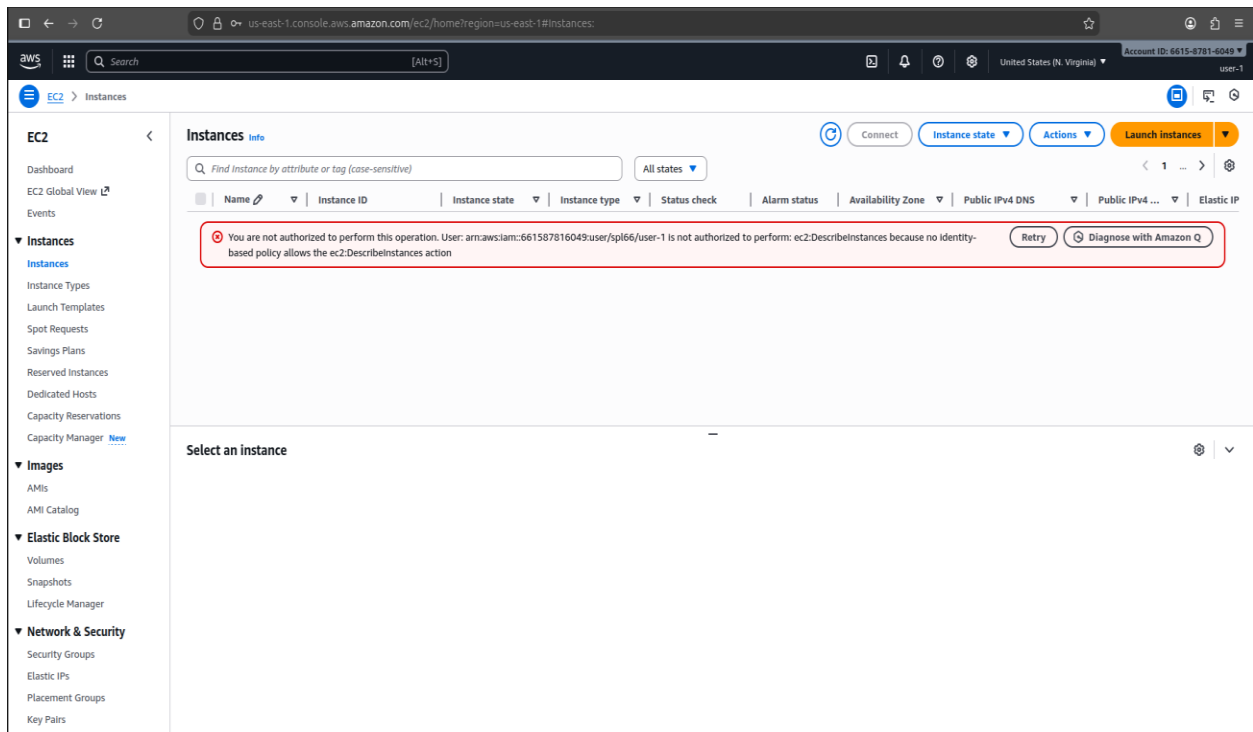
Logged in as user-1



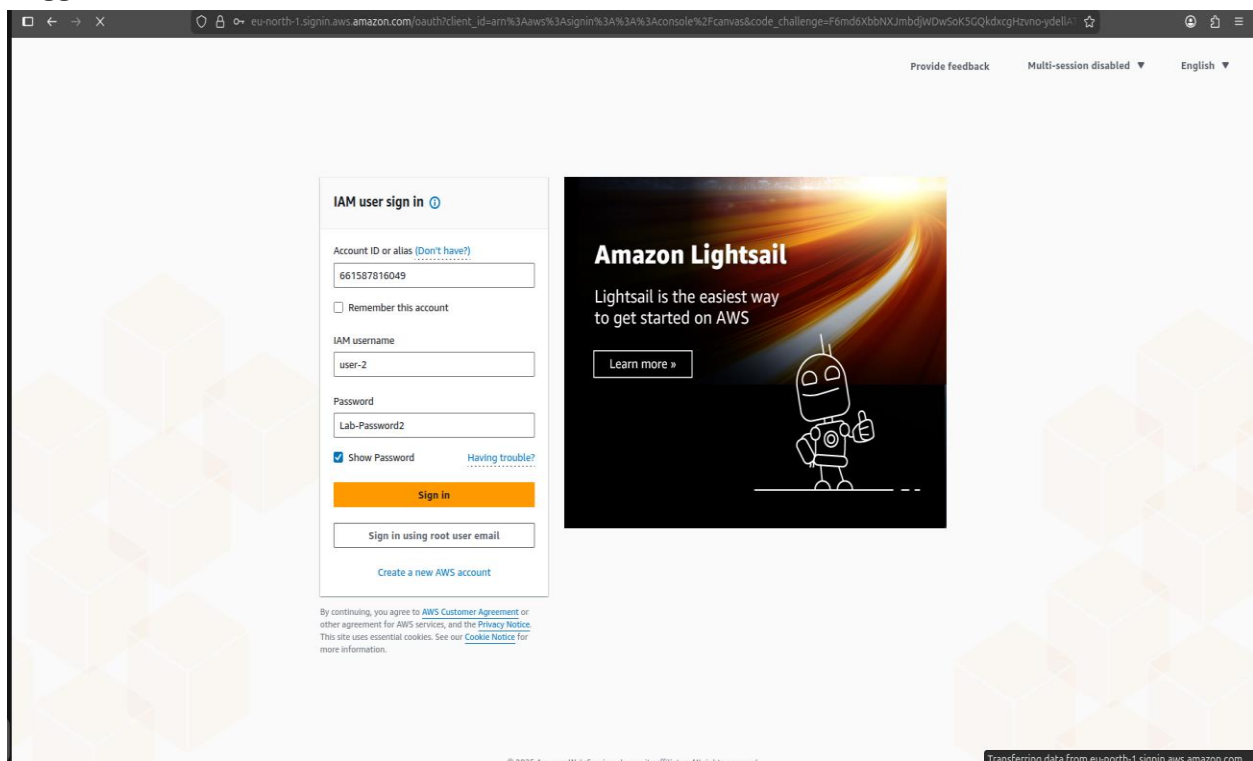
## Successfully accessed S3 and viewed bucket list (read-only)



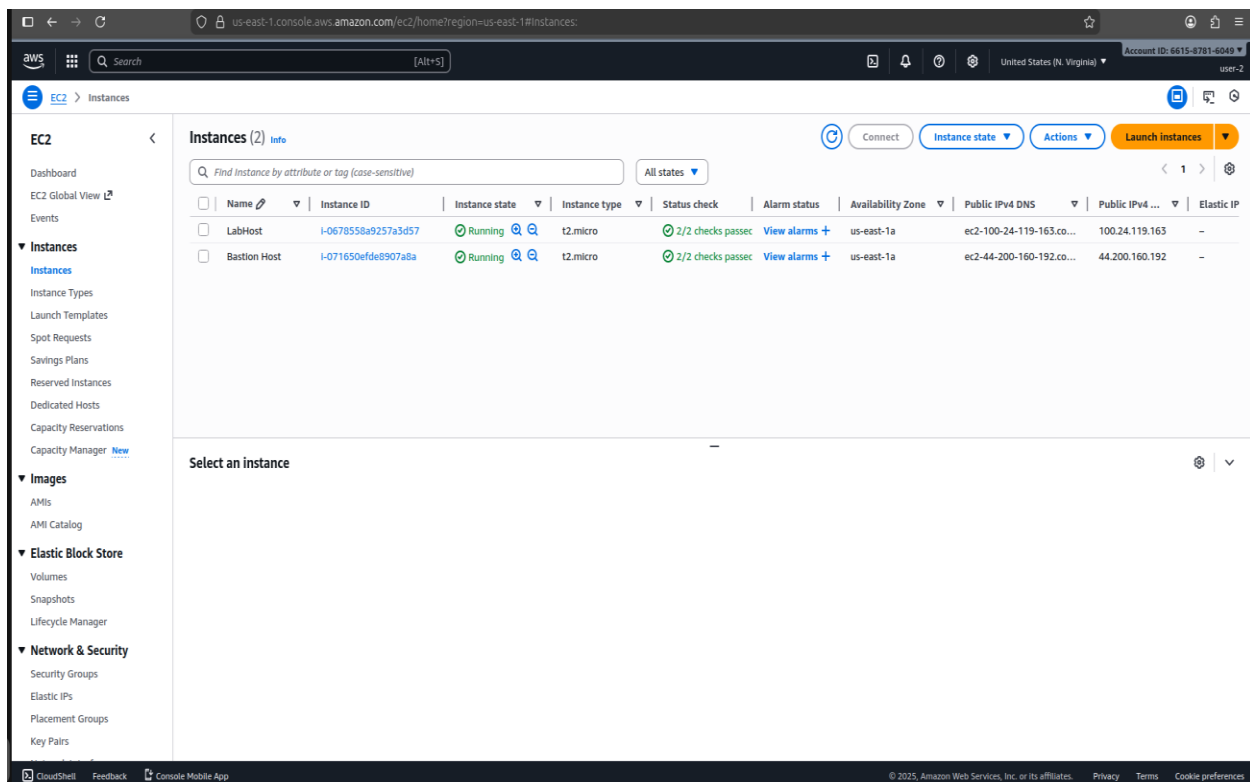
Tried accessing EC2 but received "not authorized" error



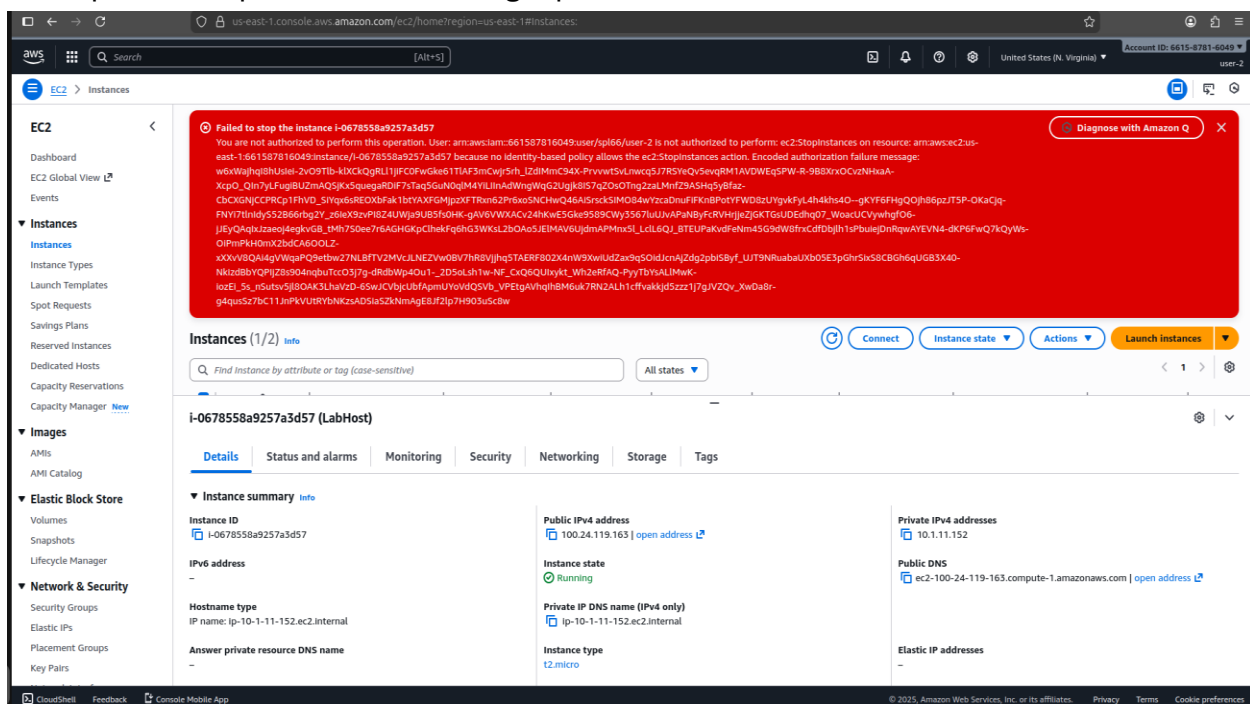
Logged in as user-2



Successfully viewed EC2 instance (read-only)

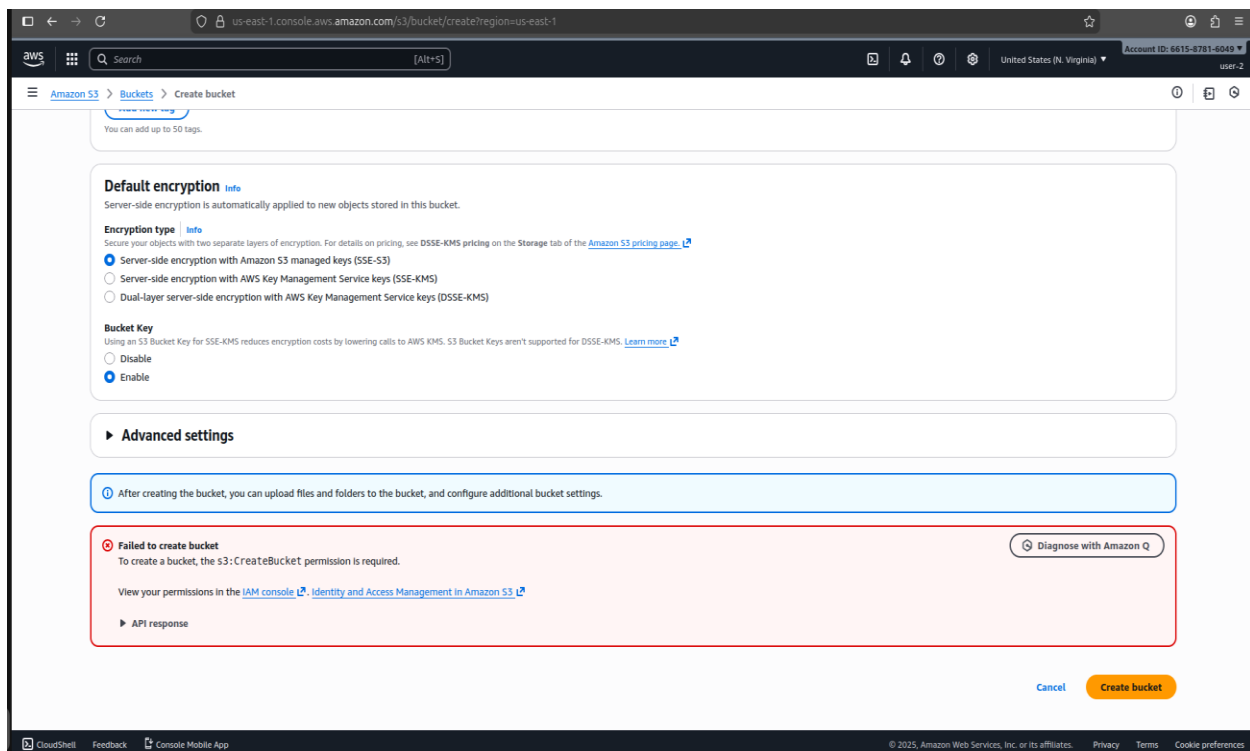


## Attempted to stop EC2 instance but got permission denied

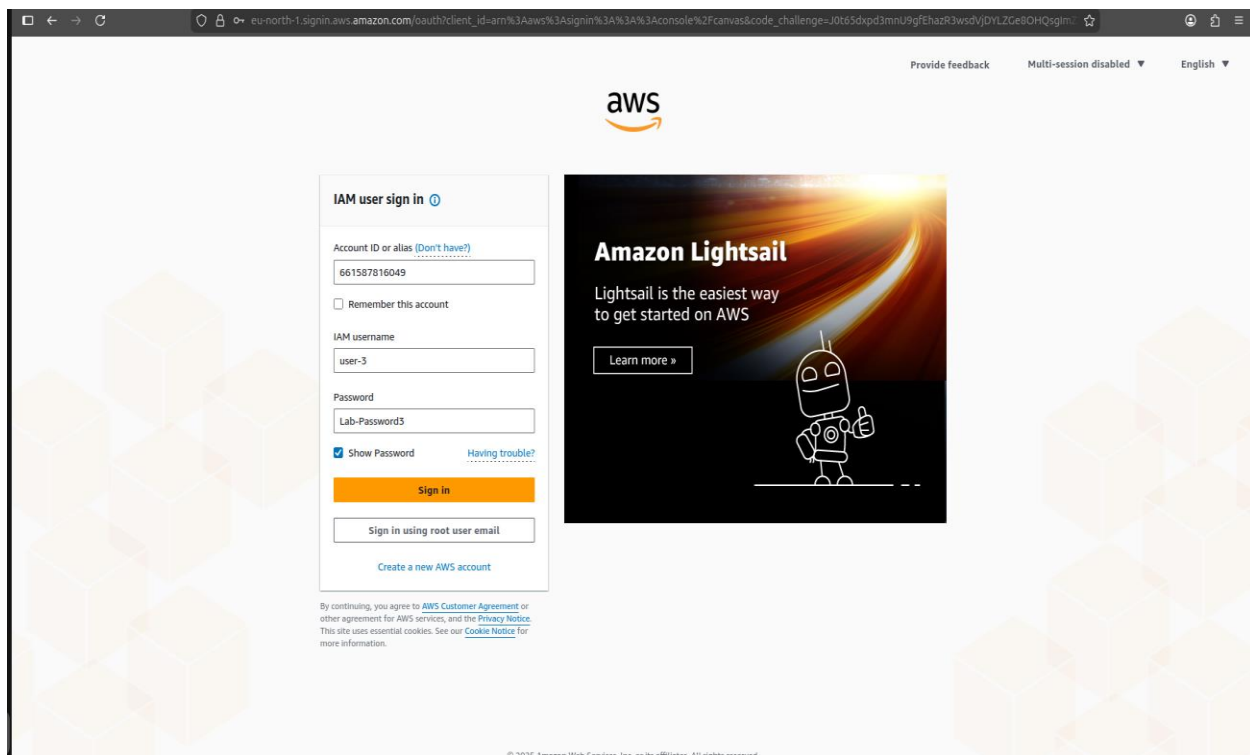


## Tried accessing S3 but was denied





Logged in as user-3



Accessed EC2 and successfully stopped the LabHost instance

us-east-1 console.aws.amazon.com/ec2/home?region=us-east-1#instances

EC2 > Instances

Successfully initiated stopping of i-0678558a9257a3d57

Instances (1/2) Info

Find Instance by attribute or tag (case-sensitive) All states

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
LabHost	i-0678558a9257a3d57	Stopping	t2.micro	2/2 checks passed	User: arn:aws:us-east-1a	us-east-1a	ec2-100-24-119-163.co...	100.24.119.163	-
Bastion Host	i-071650efde8907a1a	Running	t2.micro	2/2 checks passed	User: arn:aws:us-east-1a	us-east-1a	ec2-44-200-160-192.co...	44.200.160.192	-

i-0678558a9257a3d57 (LabHost)

Details Status and alarms Monitoring Security Networking Storage Tags

▼ Instance summary Info

Instance ID  
i-0678558a9257a3d57

IPv6 address  
-

Hostname type  
IP name: ip-10-1-11-152.ec2.internal

Answer private resource DNS name  
-

Public IPv4 address  
100.24.119.163 | open address

Instance state  
Stopping

Private IP DNS name (IPv4 only)  
ip-10-1-11-152.ec2.internal

Instance type  
t2.micro

Private IPv4 addresses  
10.1.11.152

Public DNS  
ec2-100-24-119-163.compute-1.amazonaws.com | open address

Elastic IP addresses  
-

Permissions worked as per admin role design

## 6. Conclusion

Through this lab, we successfully learned how IAM enables secure access control in AWS environments. We explored and analyzed IAM users, groups, and policies, assigned users to groups based on job roles, and verified permission restrictions by testing actions in AWS Management Console.

This demonstrates how IAM enforces secure, role-based access, ensuring every individual has only the required level of access, maintaining security and operational integrity.