ACTIVITY 1 – IAM Users & Authentication

Objective

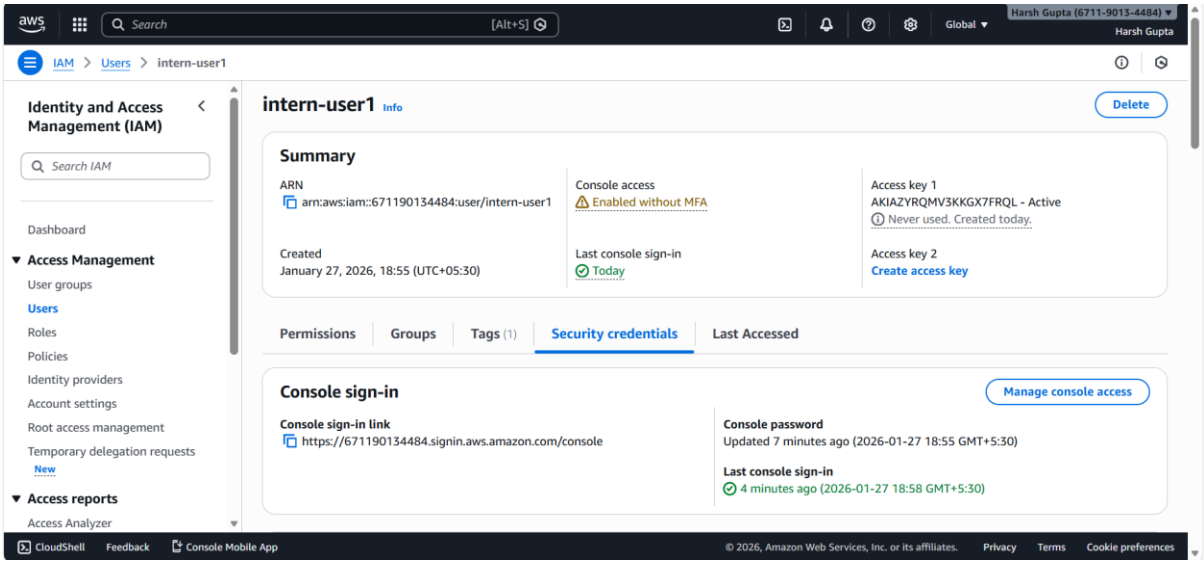
Understand IAM users and authentication mechanisms.

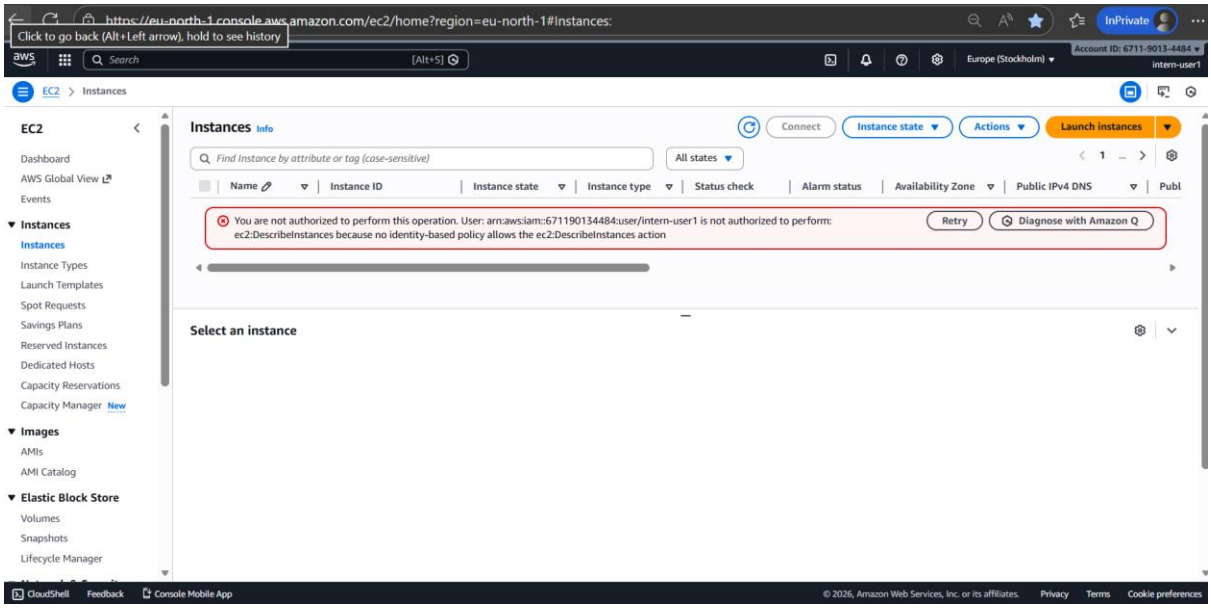
Tasks

1. Create an IAM user named: intern-user1
2. Enable **AWS Management Console access**
3. Set a **custom password**
4. Log in using the IAM user and verify:
    - You cannot access S3
    - You cannot access EC2

Expected Outcome

- User exists
- Login works
- Everything shows **Access Denied**






```
C:\Users\260016593>aws configure
AWS Access Key ID [****************FRQL]: AKIAZYRQMV3KKGX7FRQL
AWS Secret Access Key [****************WH3N]: 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
Default region name [eu-north-1]: eu-north-1
Default output format [json]: json

C:\Users\260016593>aws s3 ls

An error occurred (AccessDenied) when calling the ListBuckets operation: User: arn:aws:iam::671190134484:user/intern-user1 is not authorized to perform: s3:ListAllMyBuckets
 because no identity-based policy allows the s3:ListAllMyBuckets action

C:\Users\260016593>
```

ACTIVITY 2 – IAM Groups & Permission Inheritance

Objective

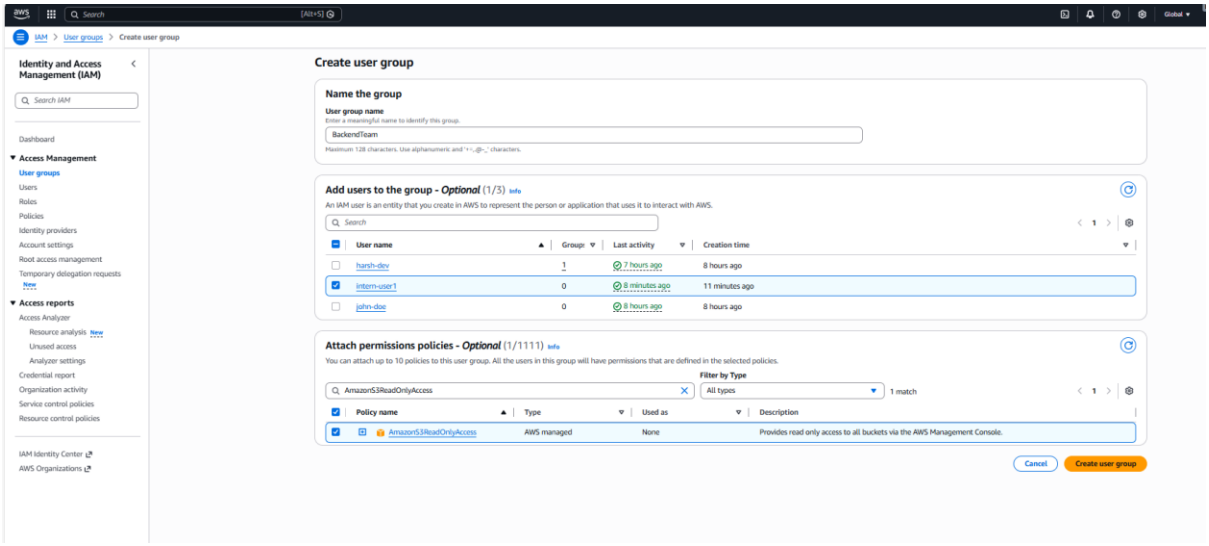
Understand why groups exist and how permissions flow.

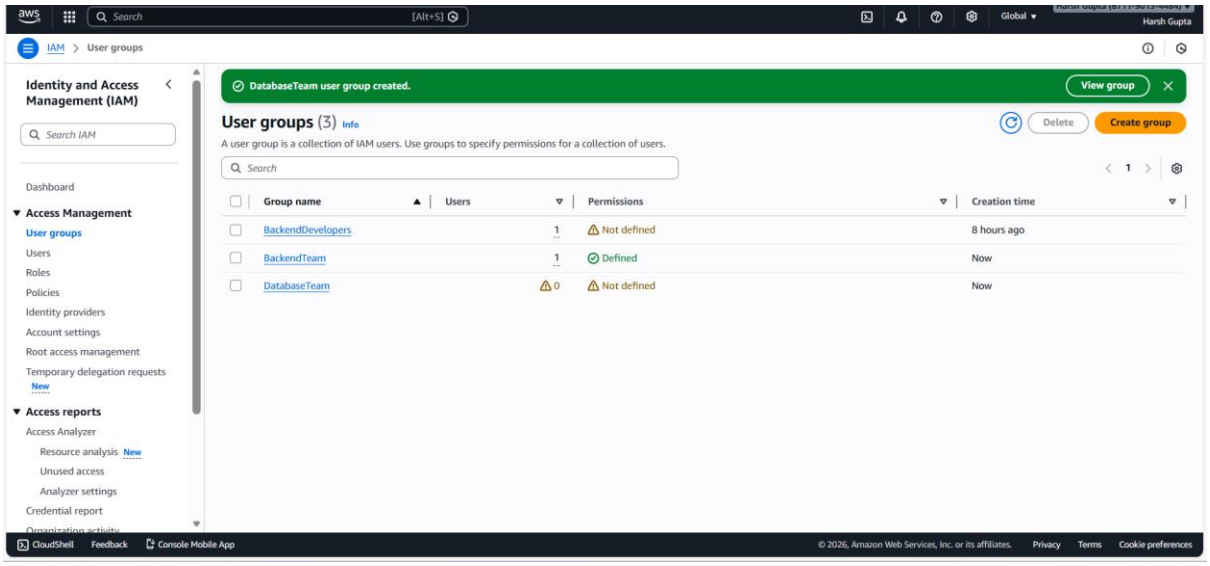
Tasks

1. Create two IAM groups:
   BackendTeam
   DatabaseTeam
2. Add intern-user1 to **BackendTeam**
3. Attach AWS managed policy:
   - AmazonS3ReadOnlyAccess to BackendTeam
4. Login as intern-user1 and:
   - List S3 buckets (should work)
   - Upload file (should fail)

Questions to Answer

- Why upload is denied?
- Where did the permission come from?






```
C:\Users\260016593>aws s3 ls
2026-01-27 11:26:24 harsh-backend-s3

C:\Users\260016593>
```

```
C:\Users\260016593>echo "Hi , Harsh" > welcome.txt

C:\Users\260016593>aws s3 cp welcome.txt s3://harsh-backend-s3/
upload failed: .\welcome.txt to s3://harsh-backend-s3/welcome.txt An error occurred (AccessDenied) when calling the PutObject operation: User: arn:aws:iam::671190134484:use
r/intern-user1 is not authorized to perform: s3:PutObject on resource: "arn:aws:s3:::harsh-backend-s3/welcome.txt" because no identity-based policy allows the s3:PutObject
action
```

Questions to Answer

● Why upload is denied?

- The upload is denied because the AmazonS3ReadOnlyAccess policy only grants read-only permissions for S3 resources. This means intern-user1 can perform actions like:

ListBucket to see all buckets and GetObject to download objects

But write operations such as PutObject (uploading files) are not included in this policy.

● Where did the permission come from?

We attached policy attached to BackendTeam group (AmazonS3ReadOnlyAccess). So intern-user1 is a member of BackendTeam and it inherits all policies attached to that group. Therefore, intern-user1's effective permissions are exactly what BackendTeam has ie, read-only access to S3.

ACTIVITY 3 – Custom Policy Using Visual Editor

🎯 Objective

Create a **least-privilege custom policy**

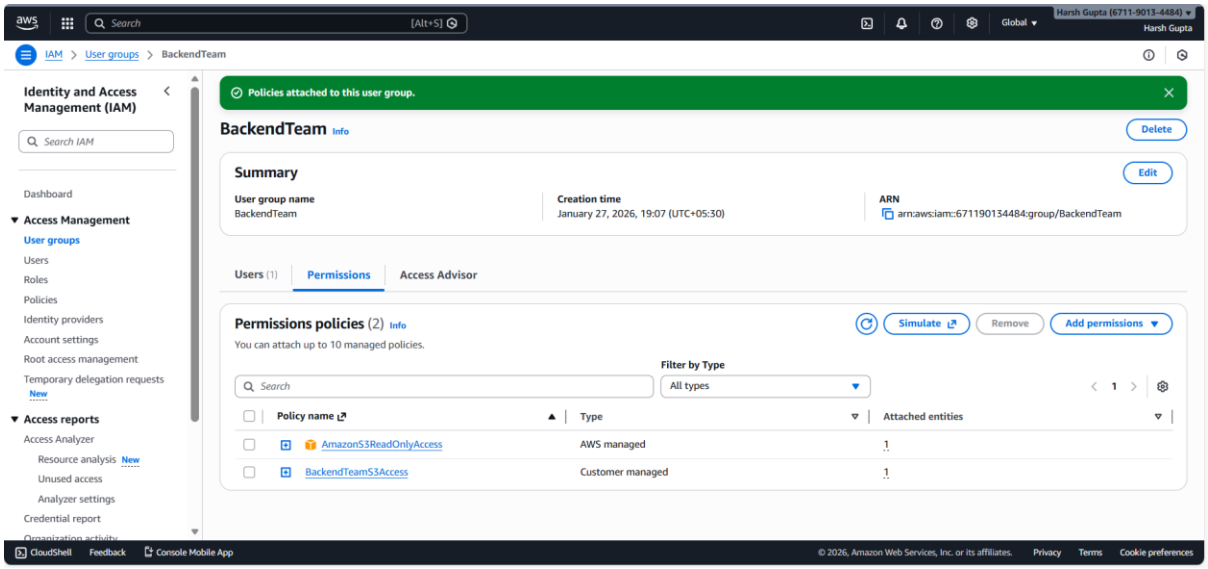
Scenario

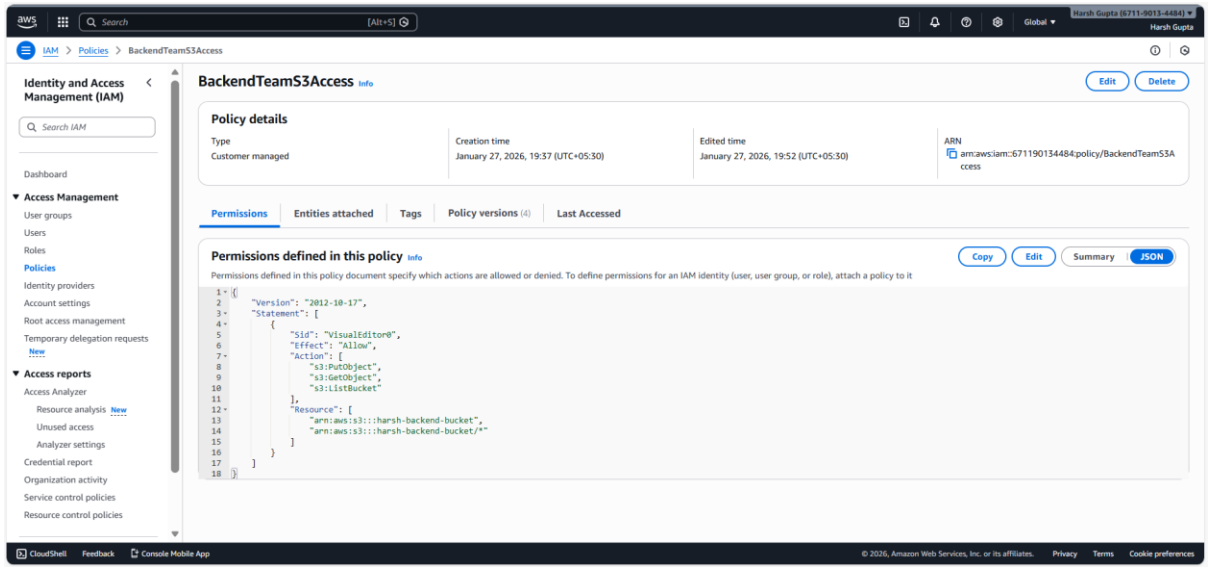
Backend team needs:

- Read & write objects in **one specific S3 bucket**
- No delete permission

Tasks

1. Create a policy using **Visual Editor**
2. Service: **S3**
3. Allow:
   - GetObject
   - PutObject
   - ListBucket
4. Restrict access to:
   my-backend-bucket
5. Attach policy to **BackendTeam**
6. Test:
   - Upload file → allowed
   - Delete file → denied






```
C:\Users\260016593>aws s3 cp welcome.txt s3://harsh-backend-bucket/
upload: .\welcome.txt to s3://harsh-backend-bucket/welcome.txt

C:\Users\260016593>aws s3 rm s3://harsh-backend-bucket/welcome.txt
delete failed: s3://harsh-backend-bucket/welcome.txt An error occurred (AccessDenied) when calling the DeleteObject operation: User: arn:aws:iam::671190134484:user/intern-u
ser1 is not authorized to perform: s3:DeleteObject on resource: "arn:aws:s3:::harsh-backend-bucket/welcome.txt" because no identity-based policy allows the s3:DeleteObject
action
```

ACTIVITY 4 – Custom Policy Using JSON Editor

🎯 Objective

Understand IAM policy JSON structure.

Scenario

Frontend team can:

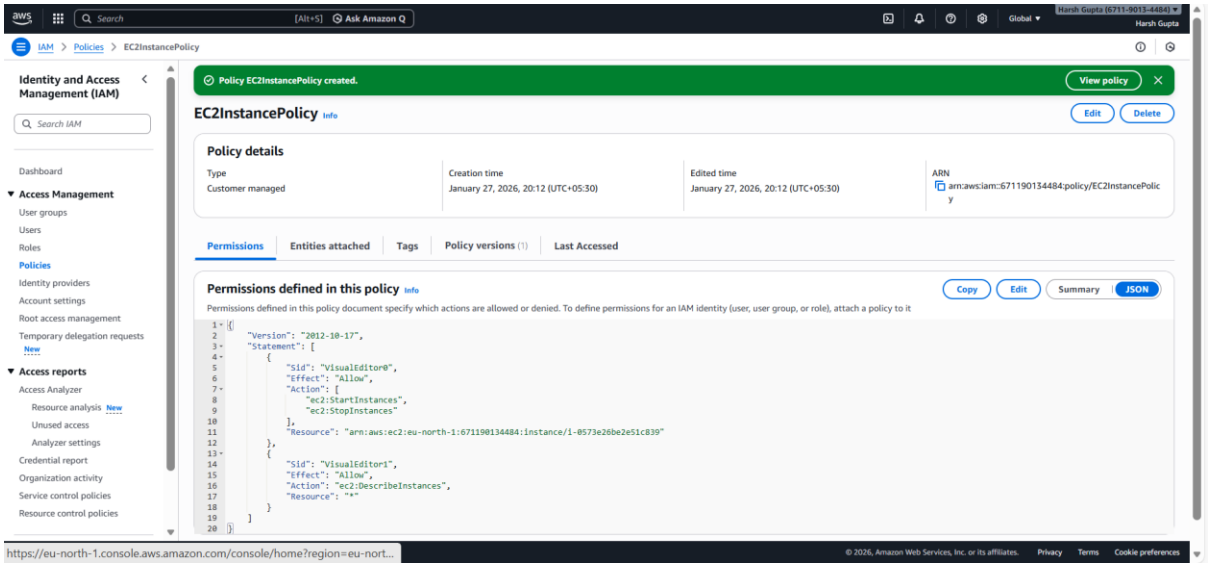
- Start/Stop only **one EC2 instance**
- Cannot access other instances

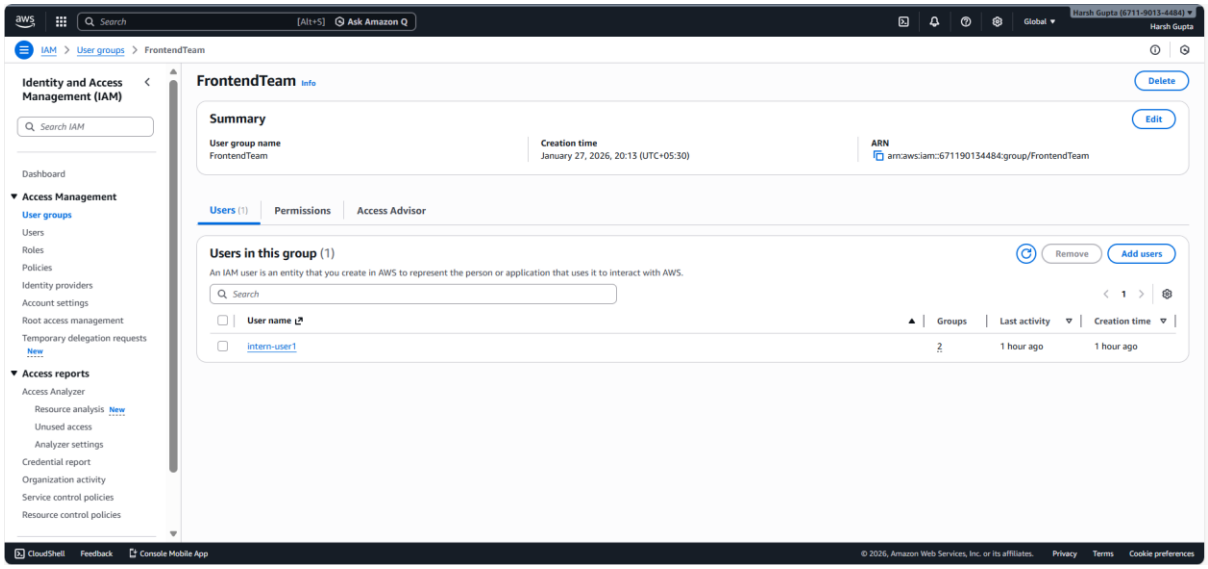
Tasks

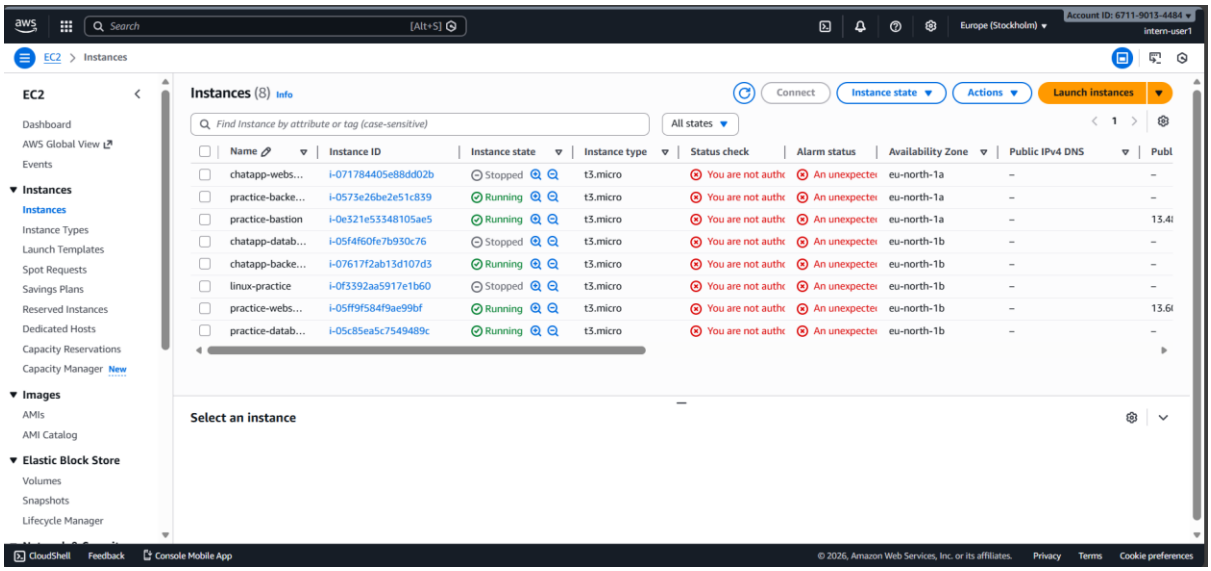
1. Create a policy using **JSON editor**
2. Allow actions:
    - ec2:StartInstances
    - ec2:StopInstances
    - ec2:DescribeInstances
3. Restrict resource to:
   One specific EC2 instance ARN
4. Create group:
   FrontendTeam
5. Attach policy
6. Test by logging in as frontend user

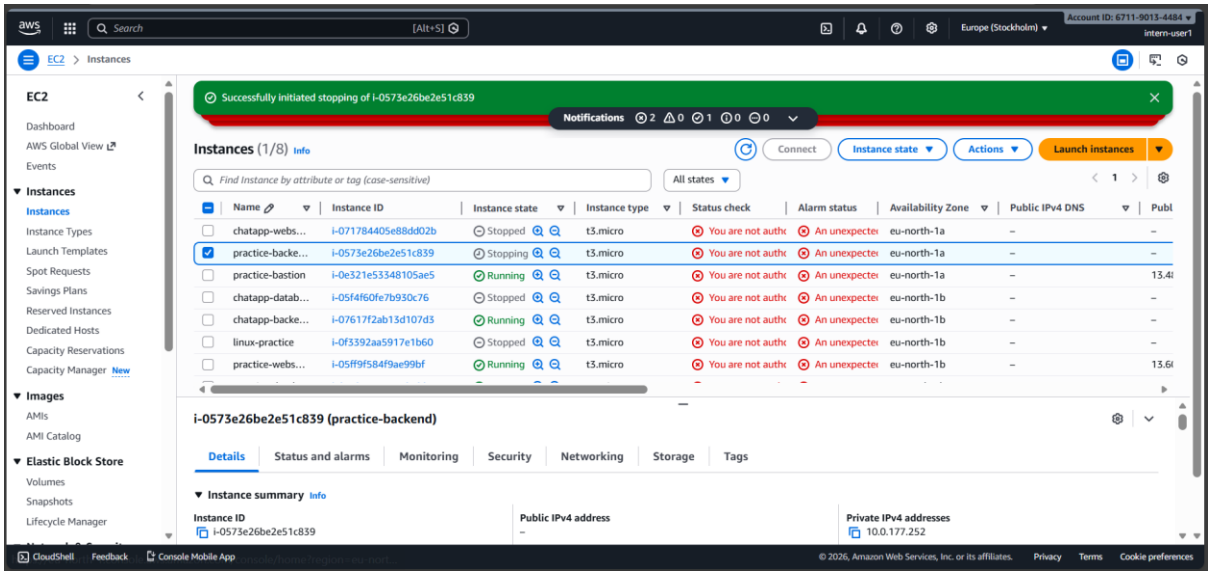
Expected Outcome

- Only one EC2 visible & controllable




IAM > User groups > FrontendTeam
Identity and Access Management (IAM)
FrontendTeam Info
Delete
Summary
Edit
User group name
FrontendTeam
Creation time
January 27, 2026, 20:13 (UTC+05:30)
ARN
arn:aws:iam::671190134484:group/FrontendTeam
Users (1)
Permissions
Access Advisor
Users in this group (1)
An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.
Remove
Add users
User name
Groups
Last activity
Creation time
intern-user1
2
1 hour ago
1 hour ago


EC2 > Instances
Instances (8) Info
Connect
Instance state
Actions
Launch instances
Select an instance


Successfully initiated stopping of i-0573e26be2e51c839
Instances (1/8) Info
i-0573e26be2e51c839 (practice-backend)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-0573e26be2e51c839
Public IPv4 address
Private IPv4 addresses
10.0.177.252