AWS IAM - Identity and Access Management

What is AWS IAM?

AWS IAM (Identity and Access Management) is a service that allows us to manage who can access AWS resources and what actions they can perform.

IAM helps in:

- Creating users
- Assigning permissions
- Securing AWS accounts

📌 IAM is a global service (not region-specific).

Why IAM is Important?

- Prevents unauthorized access
- Controls access to AWS resources
- Ensures security & compliance
- Required for every AWS service

📌 Without IAM, AWS usage is insecure.

IAM Basics

🔹 Authentication vs Authorization

Authentication → Who are you? (Login)

Authorization → What can you do? (Permissions)

🔹 Root User vs IAM User

Root User	IAM User
Full access	Limited access
Used only for setup	Used for daily work
High risk	Safer

📌 Never use root account daily.

2 ways to access AWS

1.Console access (GUI) ---> Email/Password (root user)

Username /Password (IAM user)

2.Programmatical access (CLI) ---> CLI, SDK's

1.we need to install AWSCLI software in Linux and windows (CMD).

2.we need to install SDKs for java, .NET, python etc.).

- Authentication can be done on programmatical access using keys (Access key and secret key).
- "aws configure" is used to configure AWSCLI on windows cmd and on Linux.
- Keys are user specific, individual IAM user have their own keys.
- It is not recommended to share the keys to anyone.
- Every IAM user can have max 2 set of keys.
- Once the keys are lost it is lost you can't get the same keys back. But we can regenerate then you will get the new keys.

IAM Identities

🔐 IAM Users

An IAM user represents a person or application that interacts with AWS.

Use cases:

- Admin
- Developer
- Tester

Best Practices:

- One user per person
- Enable MFA
- Avoid access keys unless required

🔐 IAM Groups

A group is a collection of IAM users.

Why use groups?

- Easier permission management
- Common access for many users

1.Group under group (nested group) are not possible.

2.It is possible to attach multiple policies to the IAM user and IAM group.

3.You can add or remove policies to the user and groups anytime.

🔐 IAM Roles

IAM roles provide temporary permissions.

Used by:

- EC2
- Lambda
- Other AWS services
- Cross-account access

📌 Roles do not have passwords.

- If you configure IAM roles, you no need to configure keys on machines.
- 1 EC2 instance can have only 1 role attached at a time. But 1 role can be attached to multiple EC2 instances at the same time.
- 2 AWS services will not communicate with each other by default, if you want them to communicate use IAM role.
- ex: - If EC2 wants to access S3 then we create the role and give permissions. Trusted entity = EC2 (To whom you are attaching role) Permissions = S3 (What kind of permissions the role should have).

Switch Role

Switch Role allows an IAM user to temporarily access another AWS account or role without creating a separate user in that account.

It uses IAM Roles + Trust Policy to provide temporary permissions.

Why Switch Role Is Used

- To access multiple AWS accounts securely
- To avoid creating multiple IAM users
- To follow least privilege and security best practices

How Switch Role Works (Simple Flow)

IAM User → Switch Role → Temporary Permissions → Access Resources

- No passwords are shared
- Temporary credentials are used
- Access expires automatically

Steps to Use Switch Role

1.Create IAM Role in target account.

2.Add trust policy to allow source account.

3.Login to source account.

4.Click Switch Role.

5.Enter account ID & role name.

Key Benefits

- Better security
- No long-term credentials
- Easy multi-account management

Switch Role allows users to temporarily assume permissions of another role or AWS account without creating separate IAM users.

IAM Policies

🔹 What is a Policy?

A policy is a JSON document that defines permissions. AWS has policy editor/policy generator these will help to generate JSON code automatically.

🔹 Policy Elements

- Version – policy language version

- Effect – Allow or Deny
- Action – AWS actions
- Resource – AWS resource
- Condition – optional rules

🔹 Policy Structure

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Action": "s3:ListBucket",
        "Resource": "*"
    }
    ]
}
```

🔹 Types of Policies

- AWS Managed Policies
- Customer Managed Policies
- Inline Policies

IAM Roles for AWS Services

🔹 Why Services Need Roles?

Services like EC2 cannot store credentials securely.

IAM roles:

- Provide temporary access
- Improve security

🔹 EC2 Role Example

- Attach S3 access role

- EC2 can access S3 without access keys

IAM Permissions & Security

🔹 Least Privilege Principle

Grant only required permissions, nothing extra.

🔹Password Policy

- Minimum length
- Complexity
- Rotation

🔹 MFA (Multi-Factor Authentication)

Adds an extra layer of security using OTP.

📌 Mandatory for root user.

IAM Tags

IAM Tags are key–value pairs that help you organize, identify, and manage IAM resources such as users, roles, and policies.

Key: Environment

Value: Dev

Why IAM Tags Are Important?

- Helps organize IAM resources
- Enables better access control
- Useful for automation and cost tracking
- Simplifies management in large AWS environments

Where IAM Tags Are Used?

IAM tags can be added to:

- IAM Users
- IAM Roles
- IAM Policies

IAM Tags in Access Control

Tags can be used in IAM policy conditions to control access.

Example use case: Allow access only to users with Department = Dev

Identity Provider (IdP)

An Identity Provider allows users to sign in to AWS using external identities instead of creating IAM users.

Examples:

- Active Directory
- Google
- Azure AD

📌 Used for Single Sign-On (SSO) in enterprises.

Identity providers allow federated access to AWS using external user identities.

IAM Credential Report

The IAM Credential Report provides details about:

- IAM users
- Password usage
- Access keys status
- MFA status

📌 Used for security auditing.

Why important:

Helps identify inactive users and unused credentials.

IAM Access Advisor

IAM Access Advisor shows:

- Which services a user or role has to access
- When those services were last accessed

📌 Helps apply least privilege.

Use case:

Remove unused permissions safely.

IAM Access Analyzer

IAM Access Analyzer identifies:

- Resources that are accessible from outside your AWS account
- Over-permissive policies

📌Improves security posture.

Commonly used with: S3, IAM roles, KMS

AWS Identity Center (SSO)

AWS Identity Center (formerly AWS SSO) provides:

- Centralized access management
- Single sign-on to multiple AWS accounts
- Integration with external identity providers

📌 Used in multi-account AWS environments.

Common IAM Mistakes:

- Giving admin access to everyone
- Not enabling MFA
- Hardcoding access keys
- Ignoring policy restrictions

Real-World IAM Scenarios

🔹 Scenario 1: EC2 Accessing S3 Securely

Problem: EC2 needs access to S3 without credentials

Solution: Attach IAM Role with S3 permissions to EC2

🔹 Scenario 2: Developer Read-Only Access

Problem: Developer should only view S3 objects

Solution: Attach read-only S3 policy to IAM group

🔹 Scenario 3: Restrict Console Access

Problem: Allow CLI access but block console login

Solution: Policy condition to deny console actions

🔹 Scenario 4: Enforce MFA

Problem: User should access AWS only with MFA

Solution: Add MFA condition in IAM policy

🔹 Scenario 5: Multiple Users, Same Permissions

Problem: 10 users need same access

Solution: Create IAM group and attach policy once

🔹 Scenario 6: Cross-Team Access

Problem: Dev team needs limited Prod access

Solution: Use IAM role with trust policy

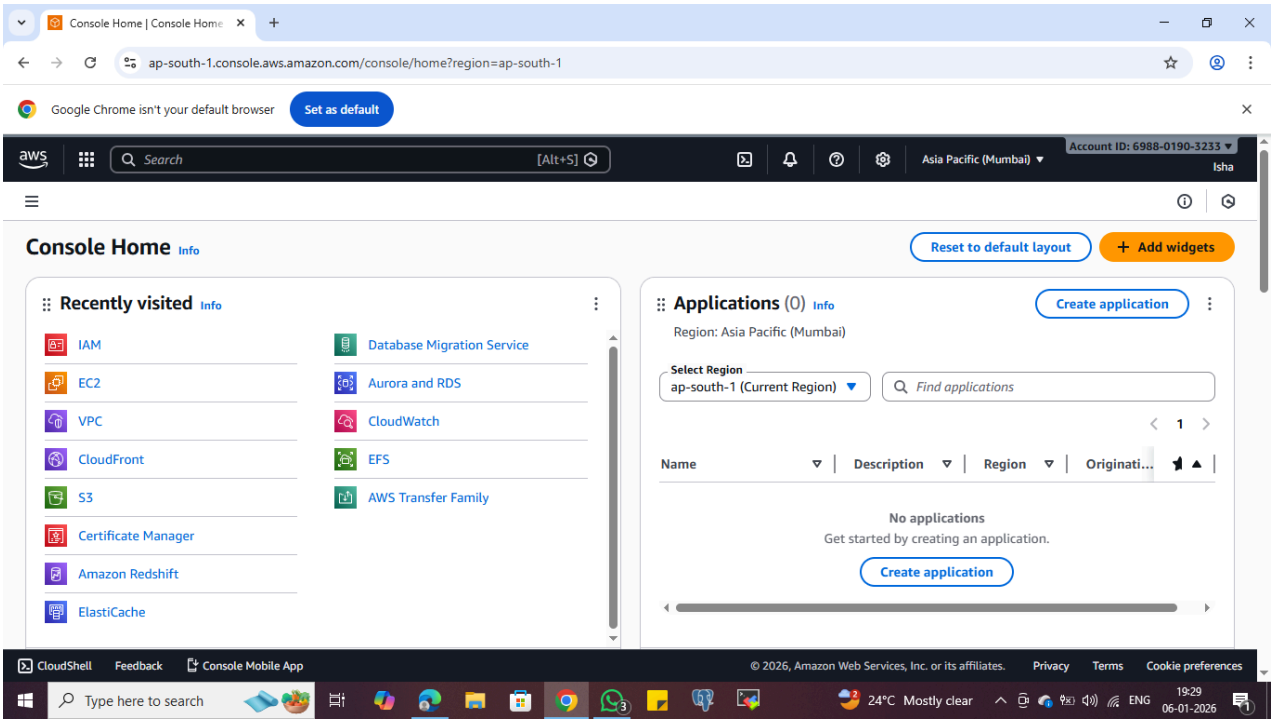
IAM Practical Work

For AWS console we have two accounts. One is root account and next one is IAM account. Always login with IAM account with username and password and then enter the MFA (I already said that how to enable MFA for accounts in previous posts) of IAM account to enter into console. Use only IAM account for daily use.

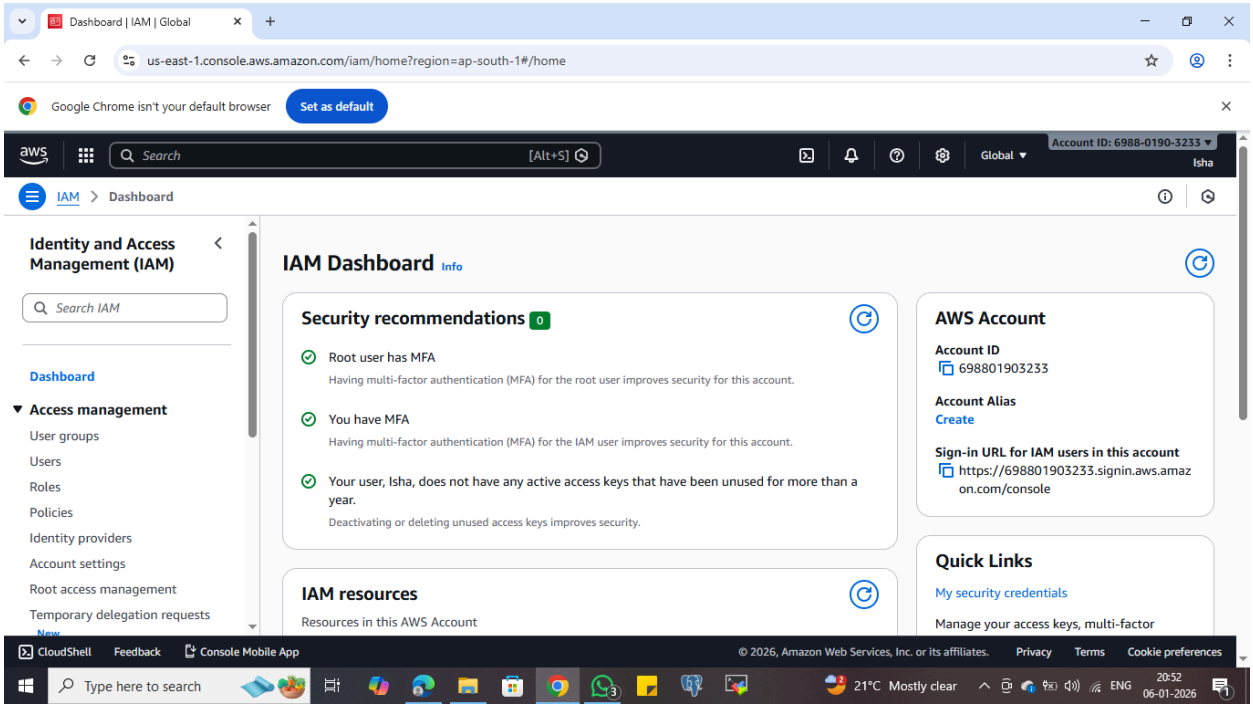
Practical 1: Create IAM User

Steps:

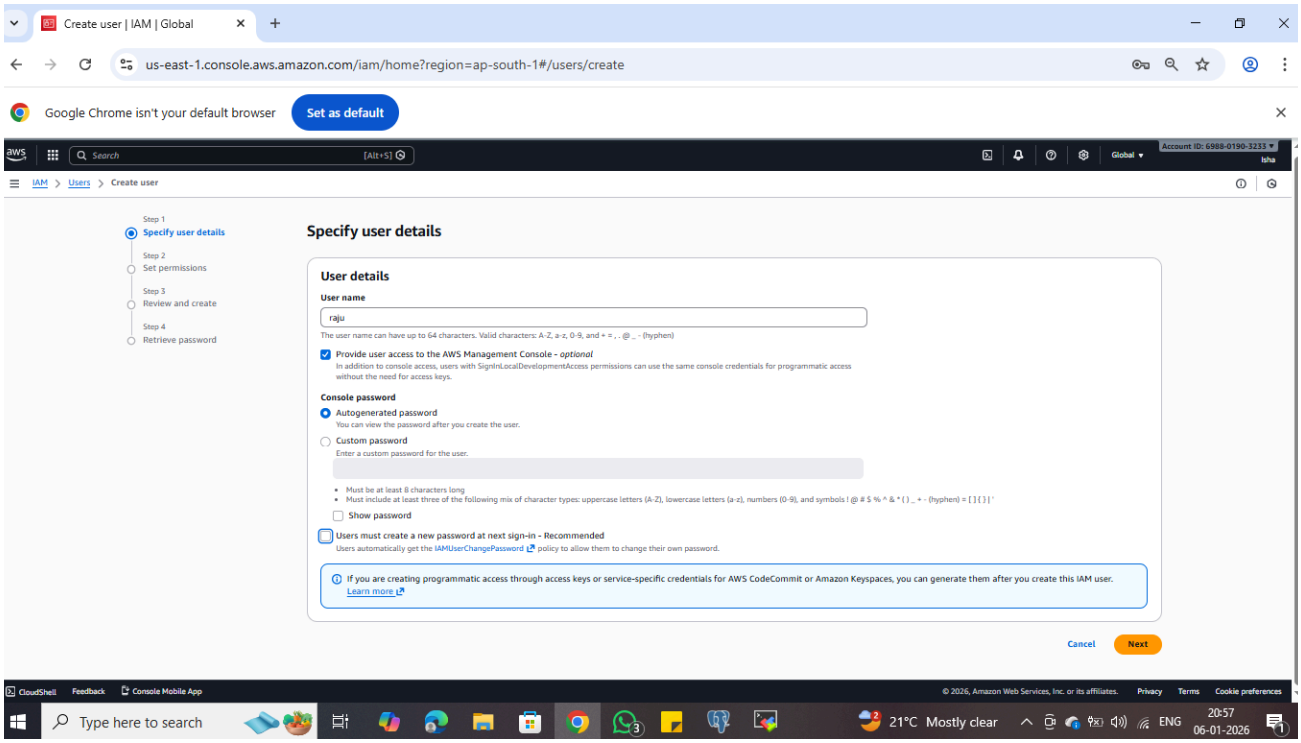
Go to AWS console. Here is the console...



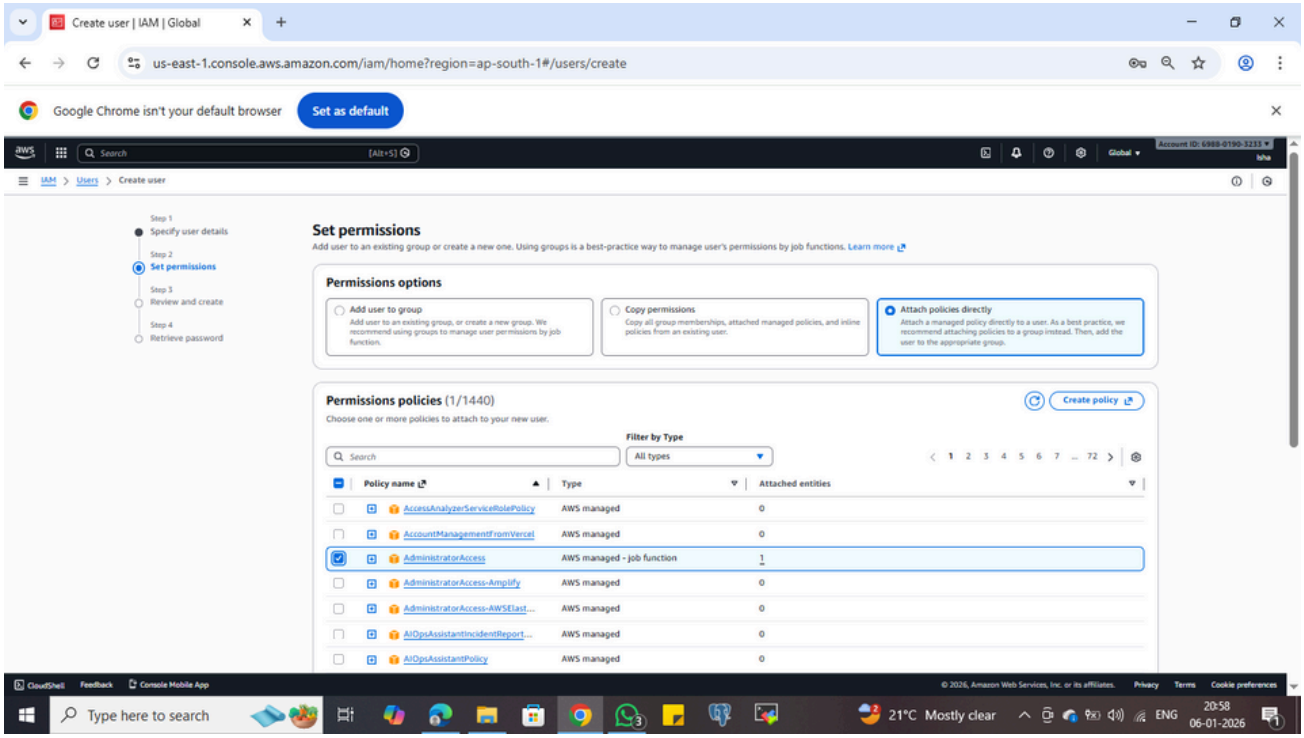
Now search for the IAM service...

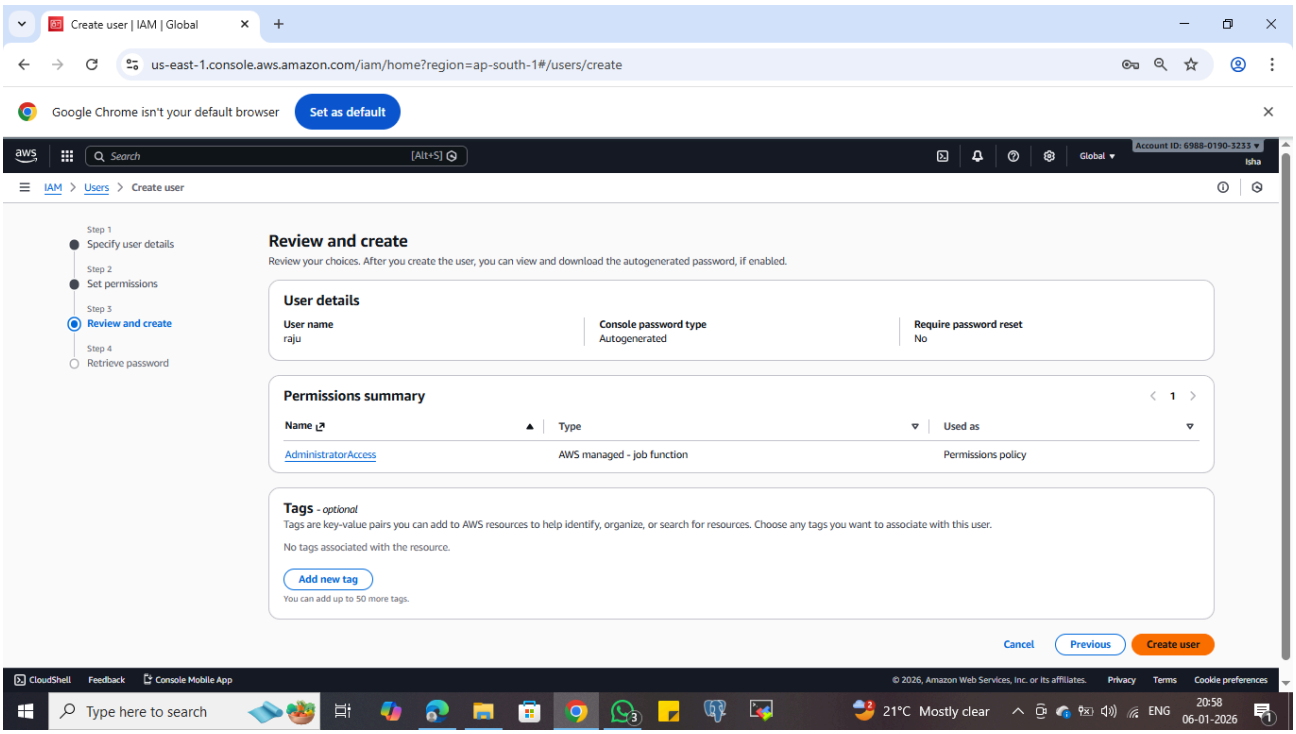


Go to users then select create user. Enter username and enable console access

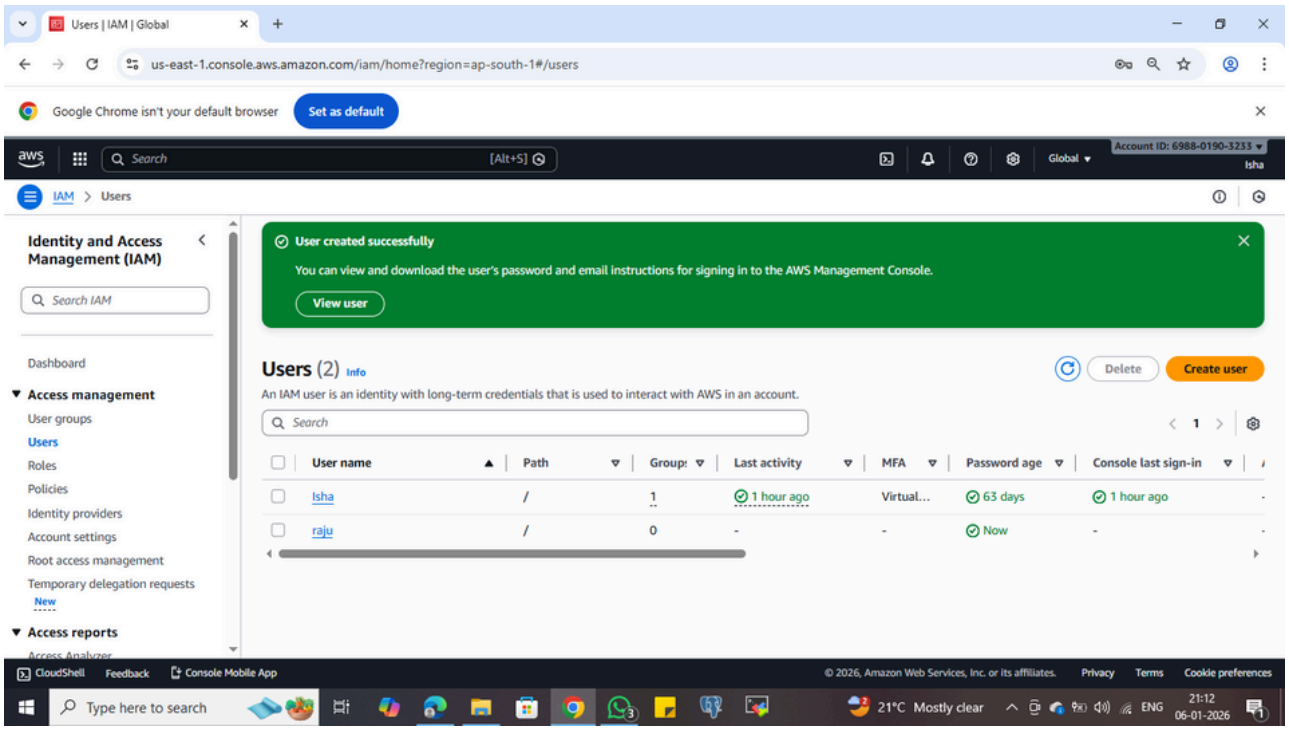


Then attach policy (Admin Access).





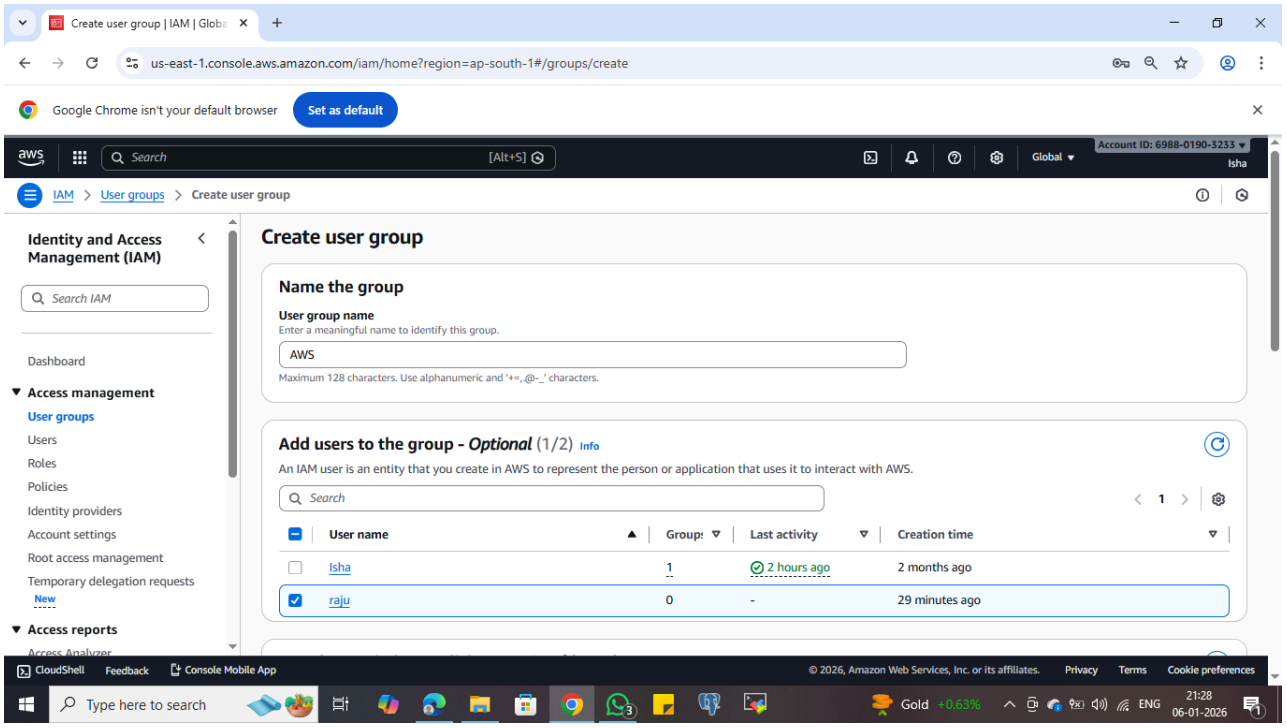
Then click on create user.



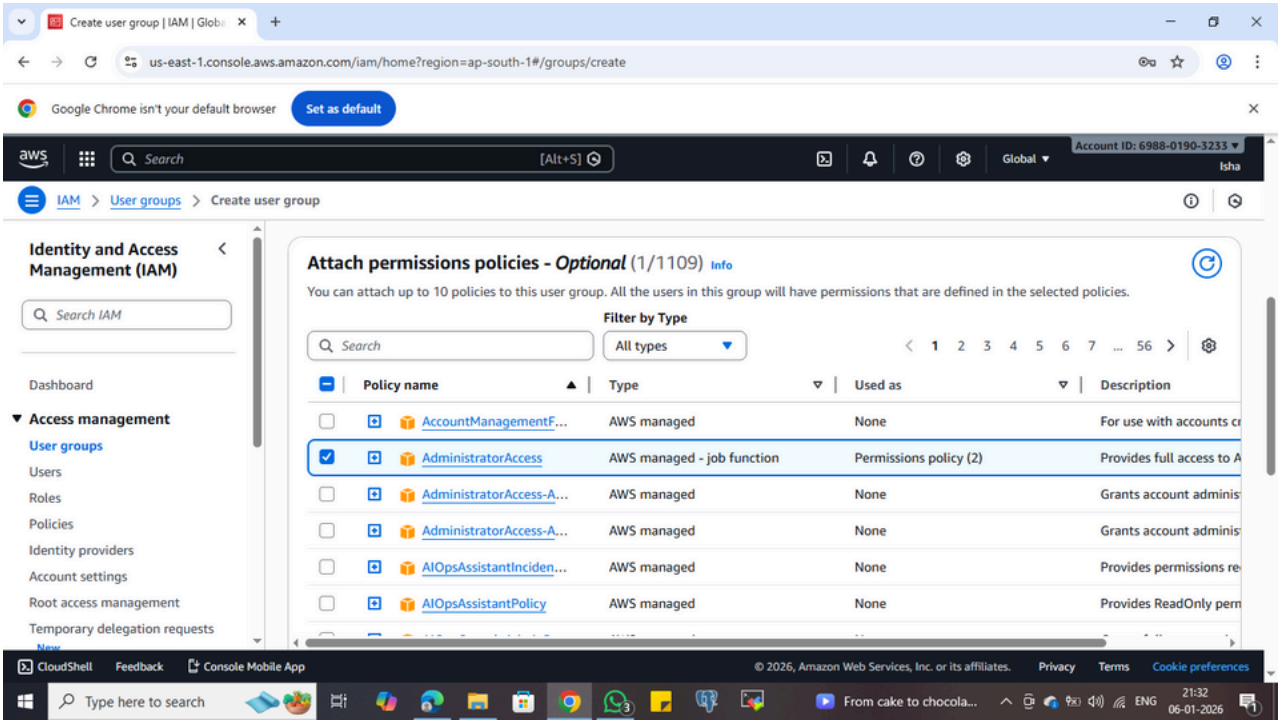
User created successfully.

Practical 2: Create IAM Group

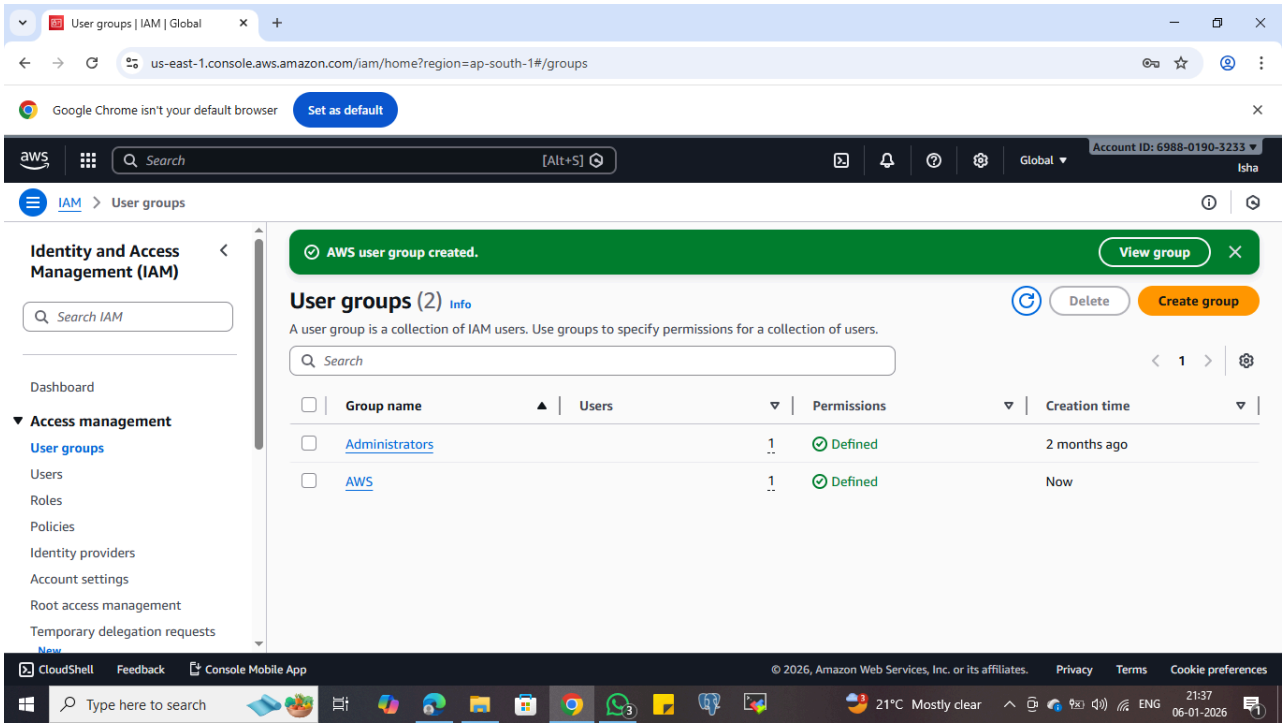
Go to IAM then select user groups then click create group. After that enter group name and add the user to the group.



Then attach the policies. Ex: I enable admin access. U can add any policies what u want.

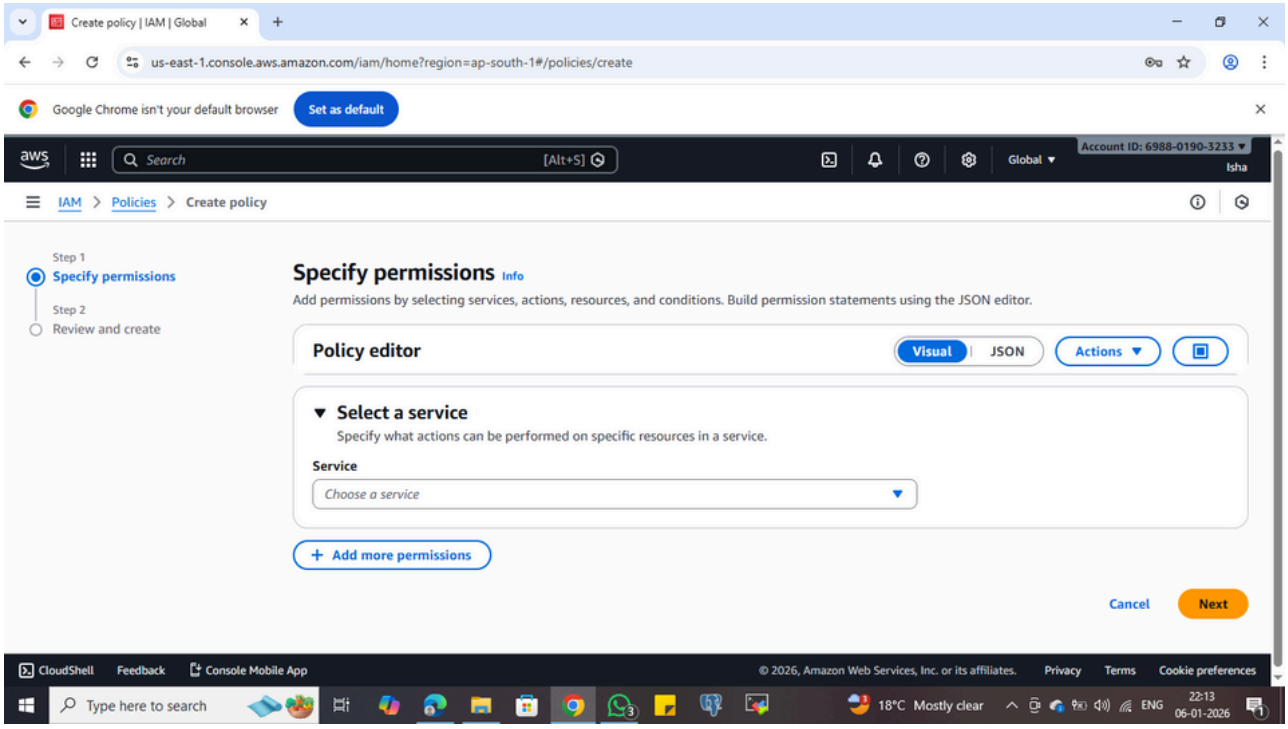


After that click create user group. Now group created successfully.

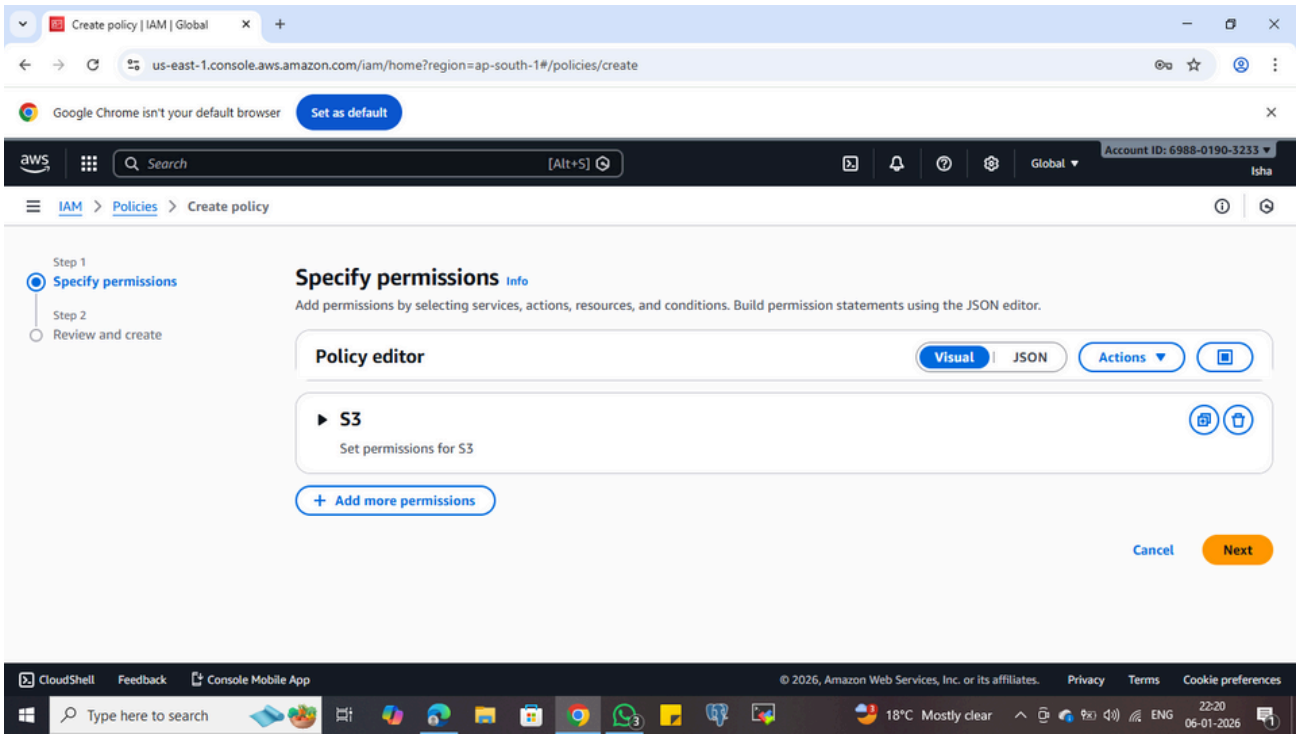


Practical 3: Create Policy without Json

Go to IAM then select policies then click create policy.



Now I want to create policy for S3 read access. We can create an S3 Read-Only custom policy using **visual editor**. Make sure visual editor tab is selected (default). Under Service, search S3**.** Select **S3**



Now expand s3 under that

Expand List

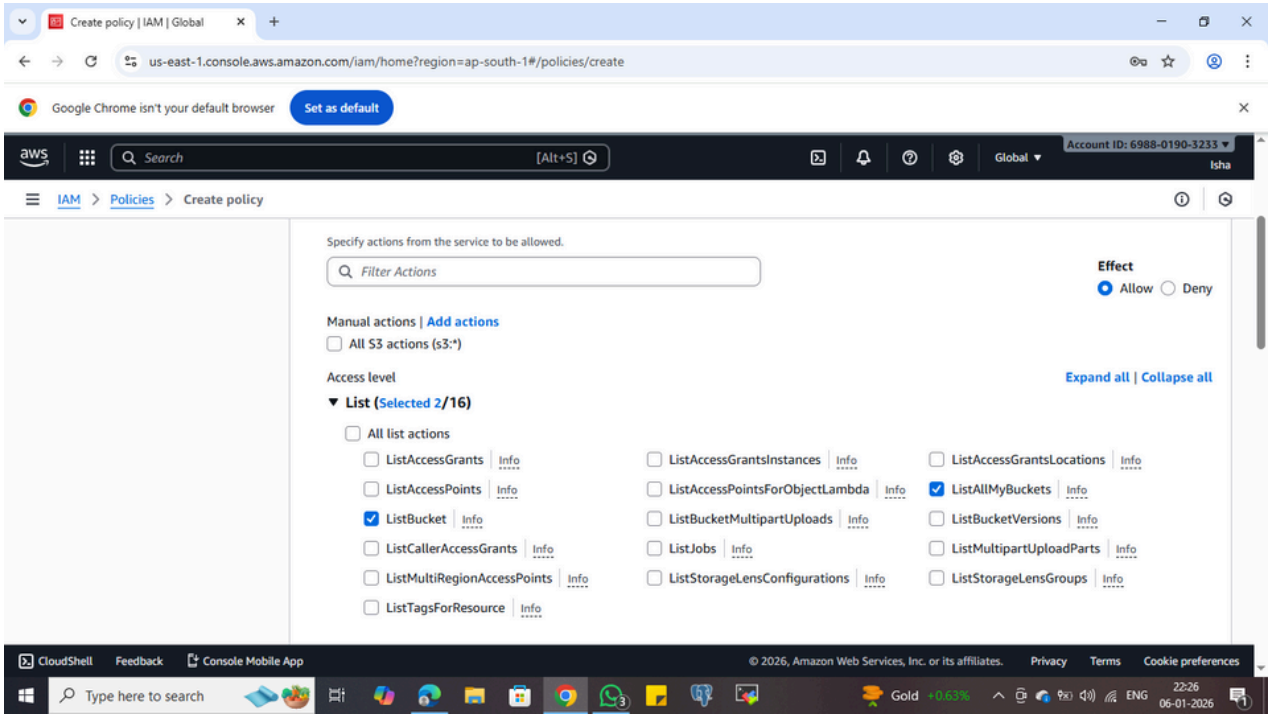
- ListAllMyBuckets
- ListBucket

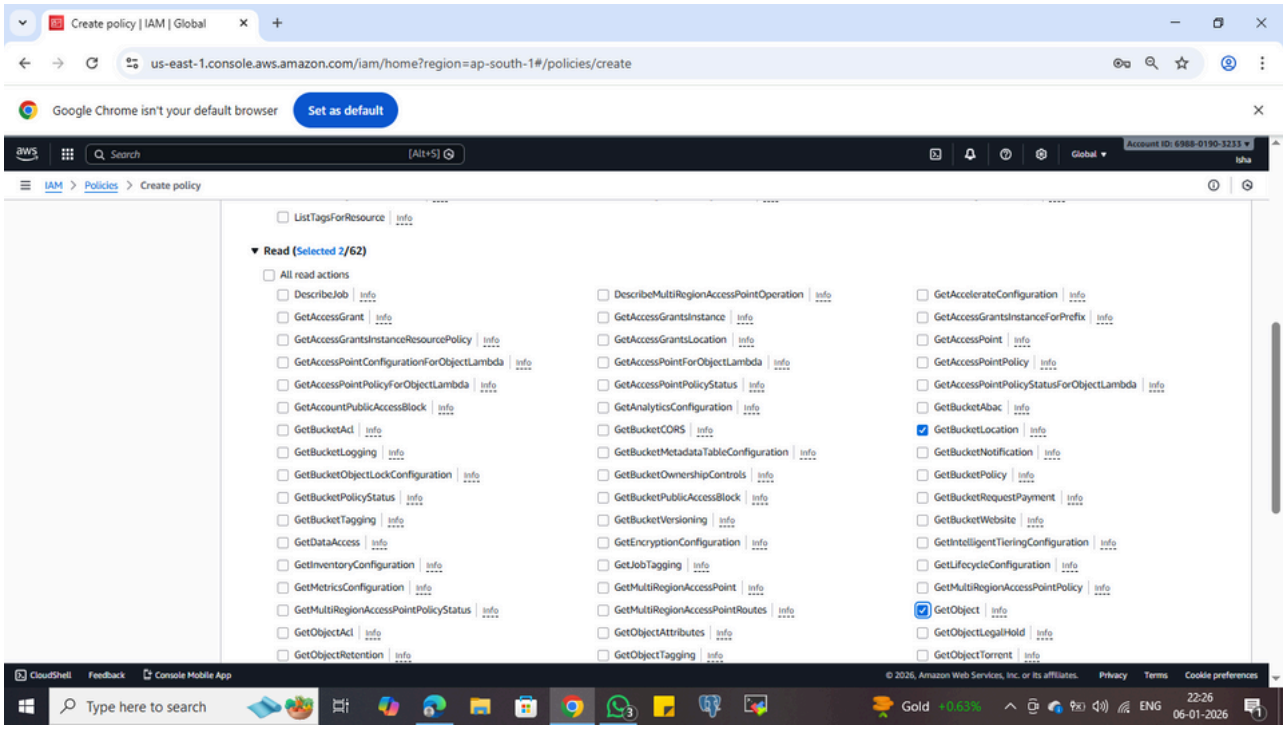
Expand Read

- GetObject

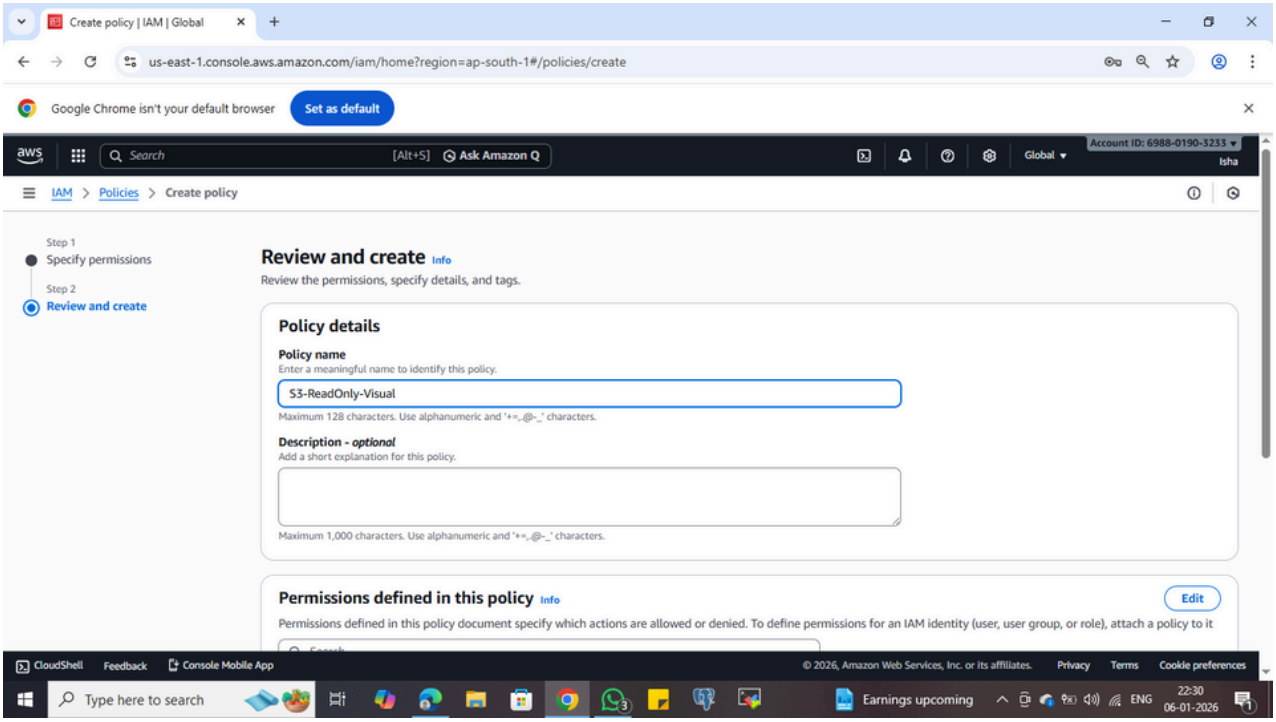
- GetBucketLocation

Do not select Write or Delete actions.



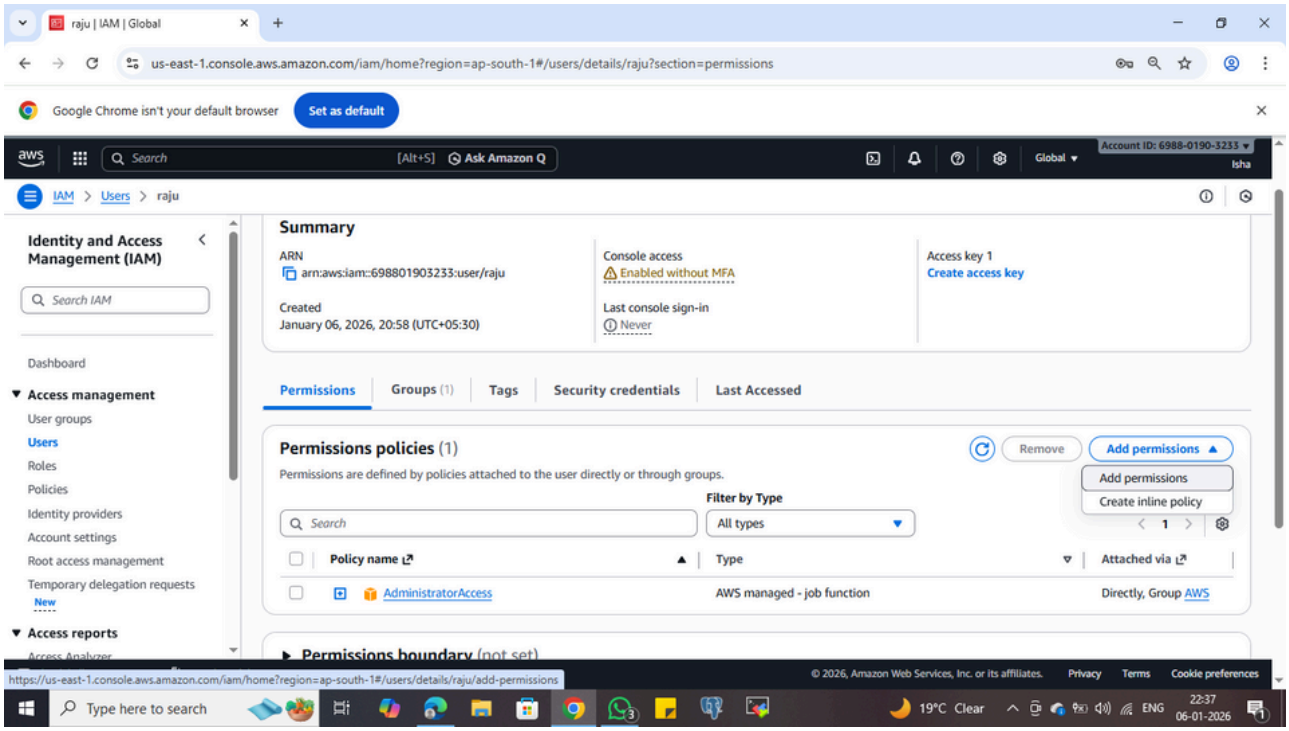


Now choose resources. Under resources select all. Then click next then set the name of the policy then click create policy.

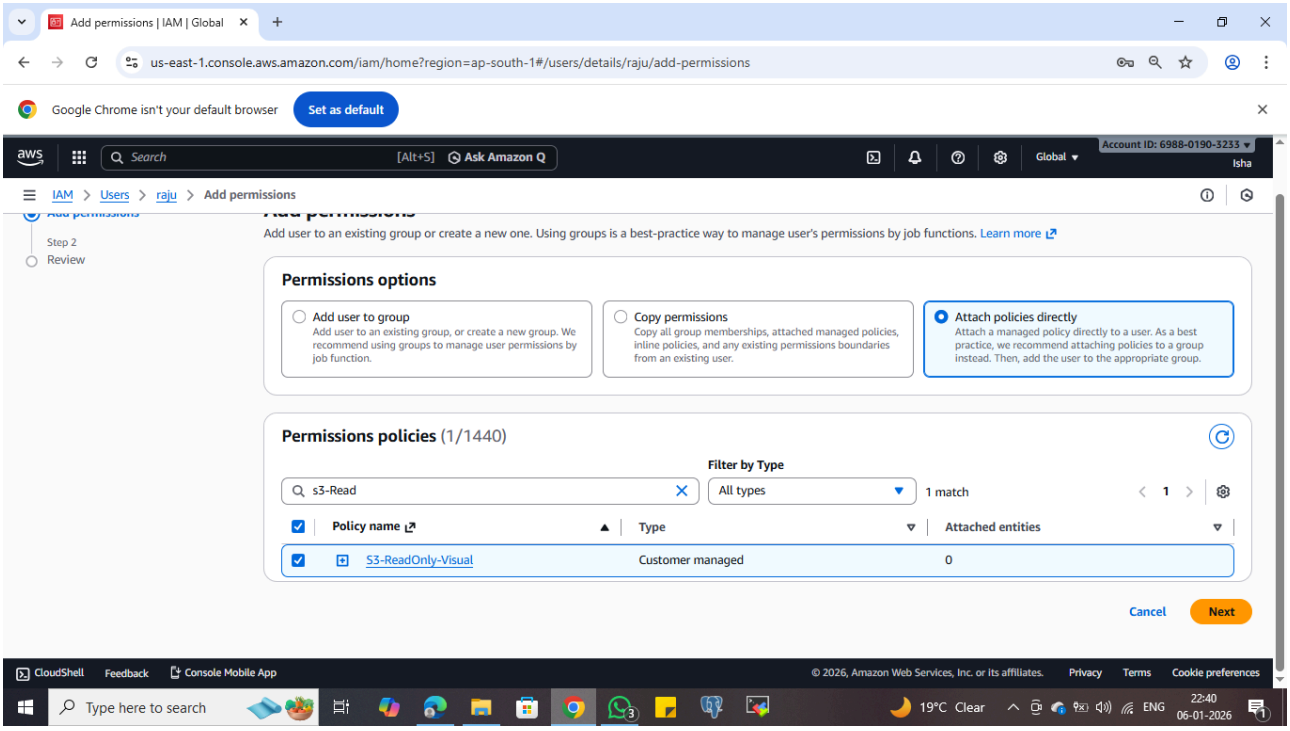


visual policy created successfully.

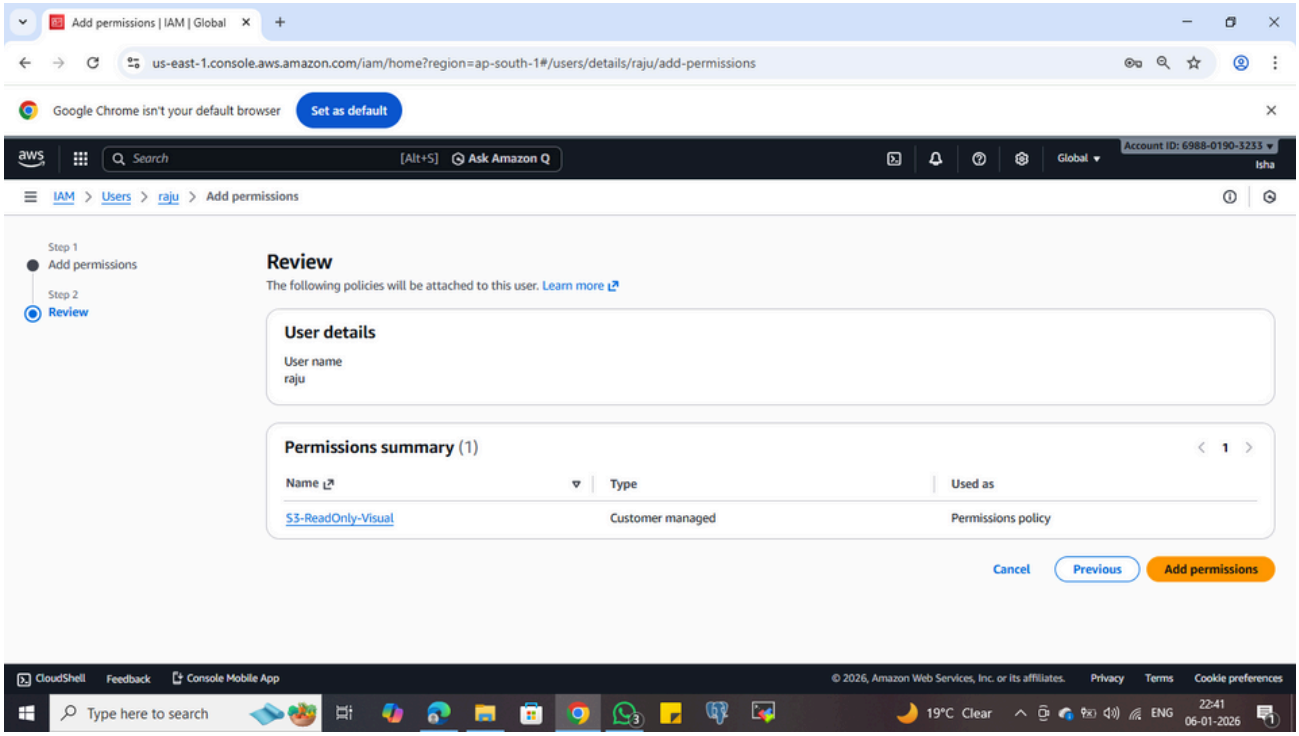
Now attach the policy to the user or group. Go to Users. Select your user then open Permissions.

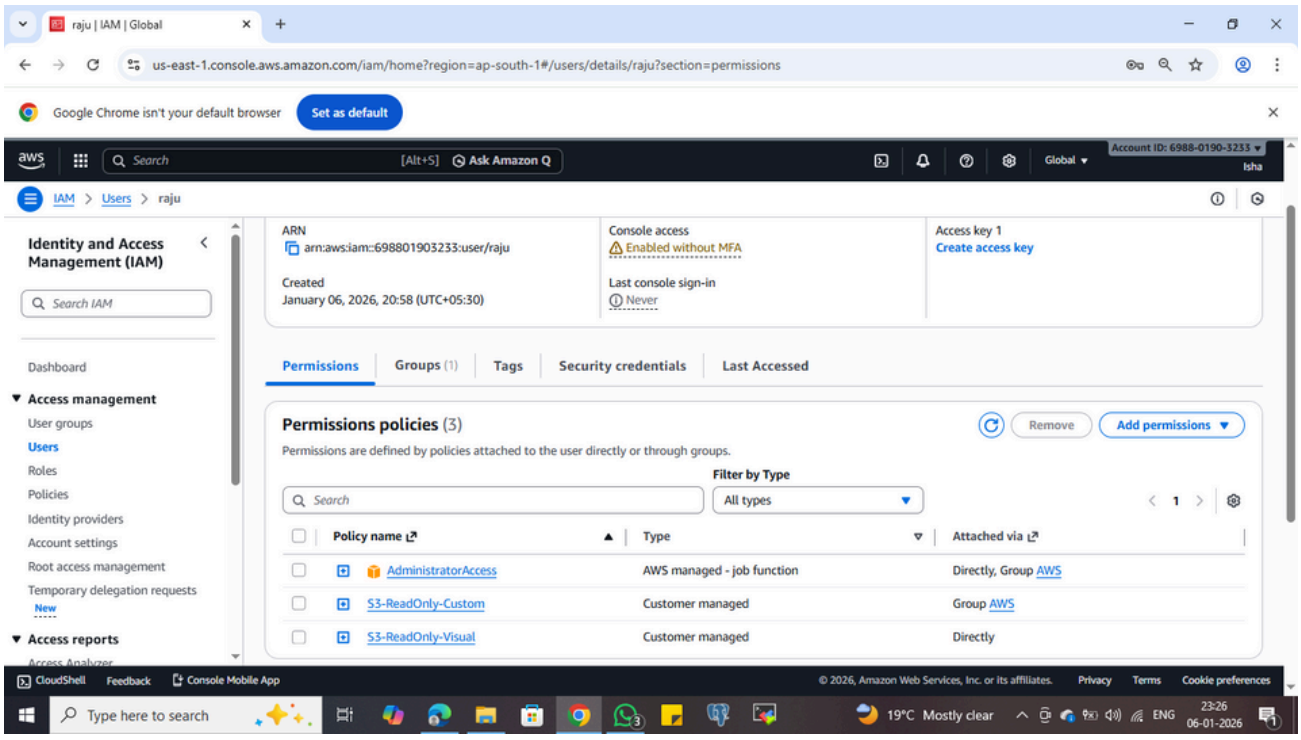


Click Add permissions. Select Attach policies.



Search S3-ReadOnly-Visual then click next. Now Select add permissions.

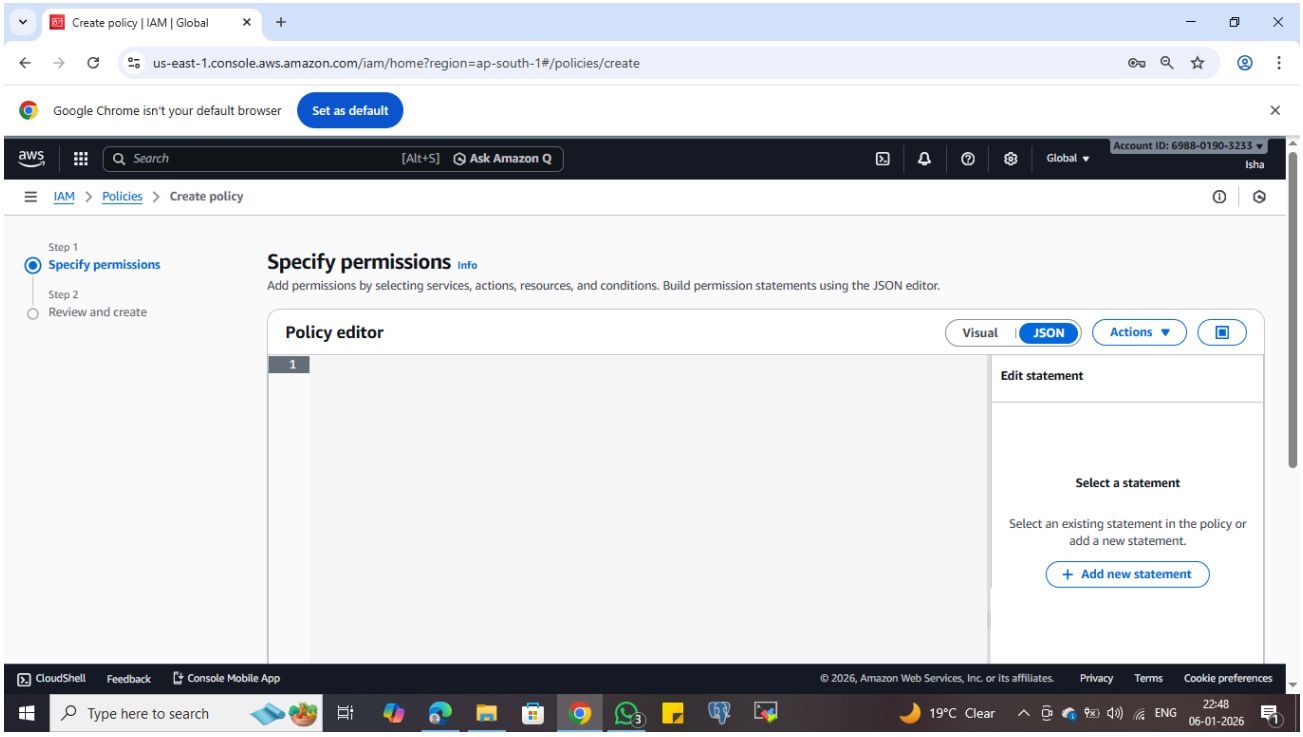




policy will be attached successfully

Practical 4: Create Custom Policy with Json

Go to IAM then select policies then click create policy. select JSON tab. You will see a default JSON template. Delete the existing content

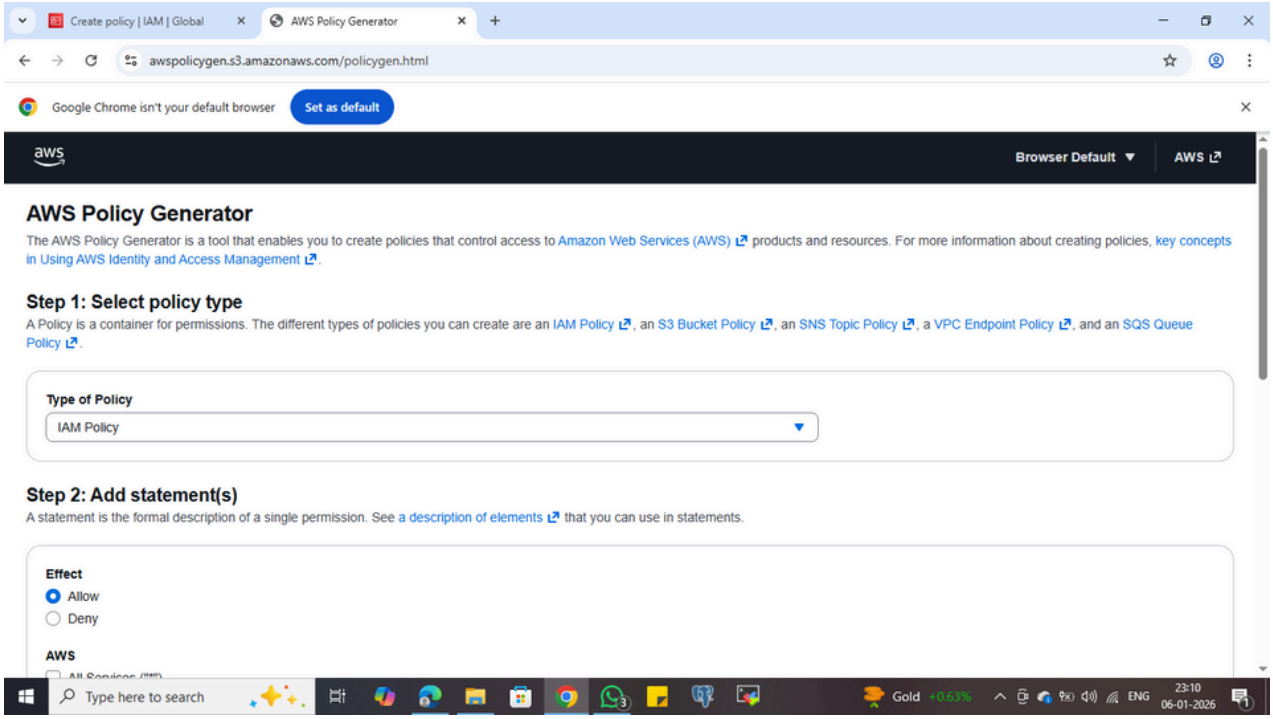


Now I want to create policy for S3 read access. We can create an S3 Read-Only custom policy using JSON template. For the code, AWS provides an official tool called the IAM Policy Generator, which lets you SELECT options and AUTO-GENERATE JSON.

Open the Policy Generator Link in browser

👉 https://awspolicygen.s3.amazonaws.com/policygen.html

Select Policy Type: IAM Policy

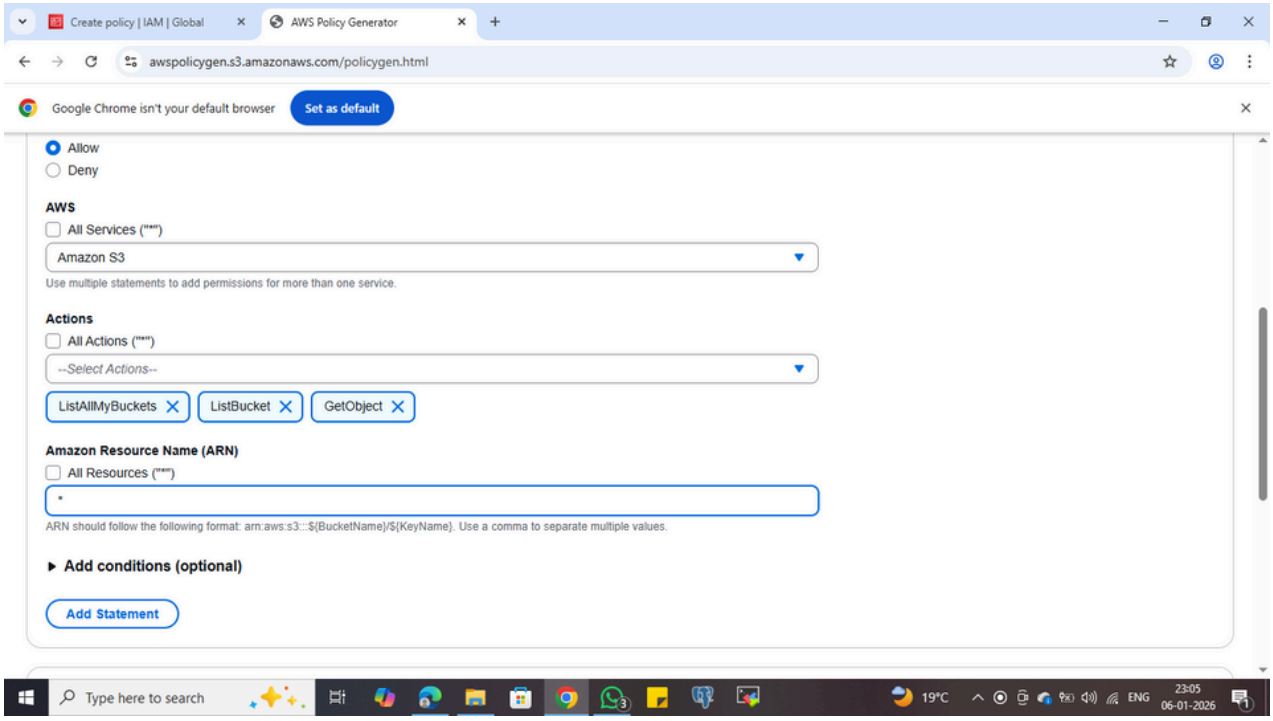


Select AWS Service: Amazon S3

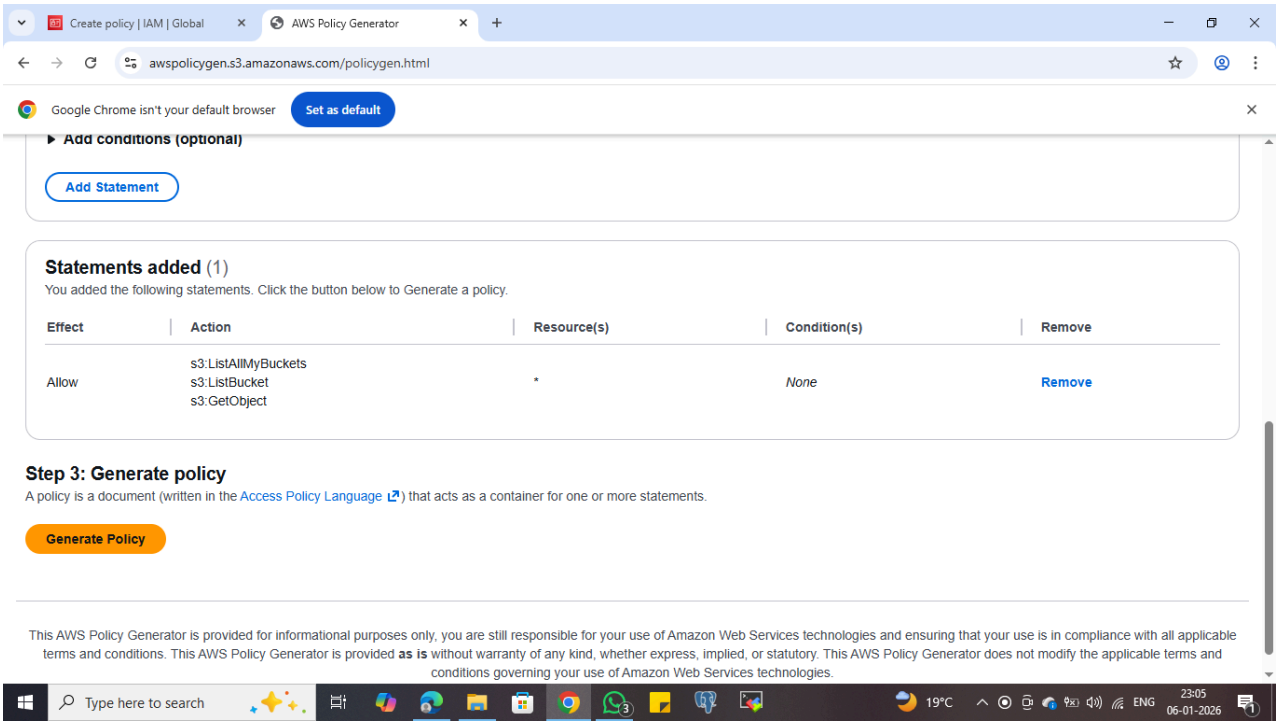
Select Actions → select:

- ✔ ListAllMyBuckets
- ✔ ListBucket
- ✔ GetObject

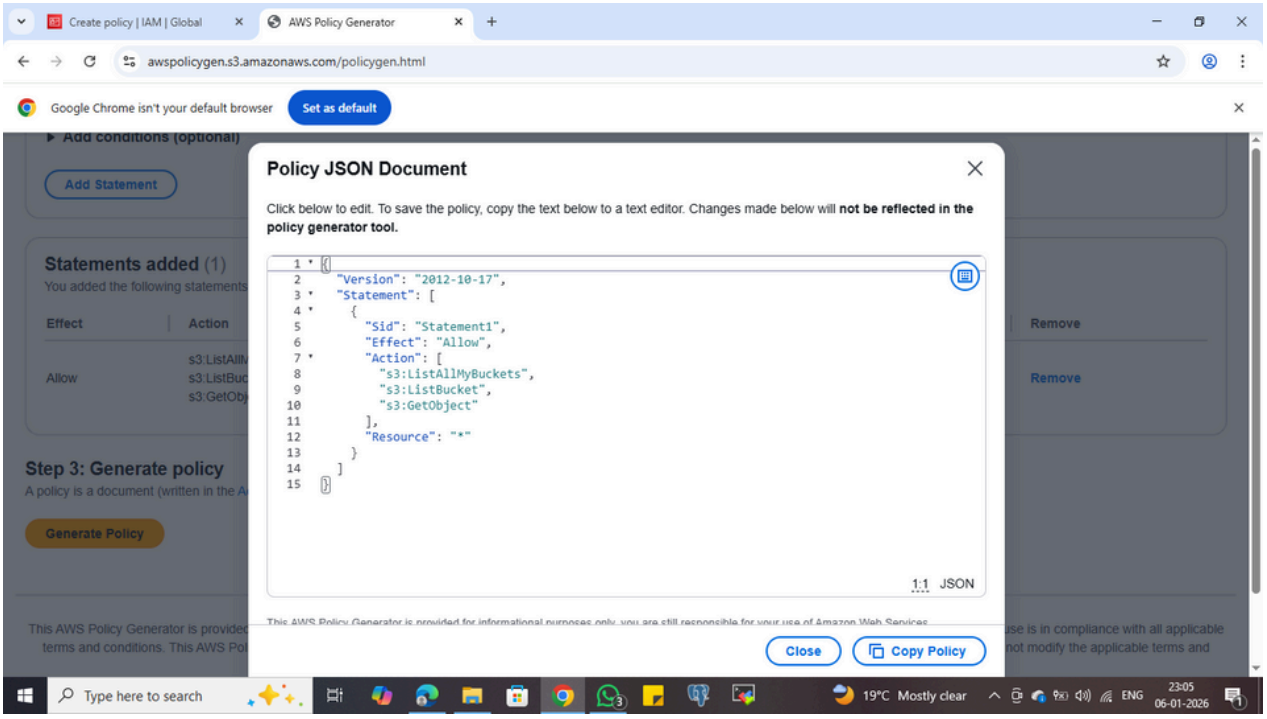
Specify Resource → Amazon Resource Name (ARN): For practice → *



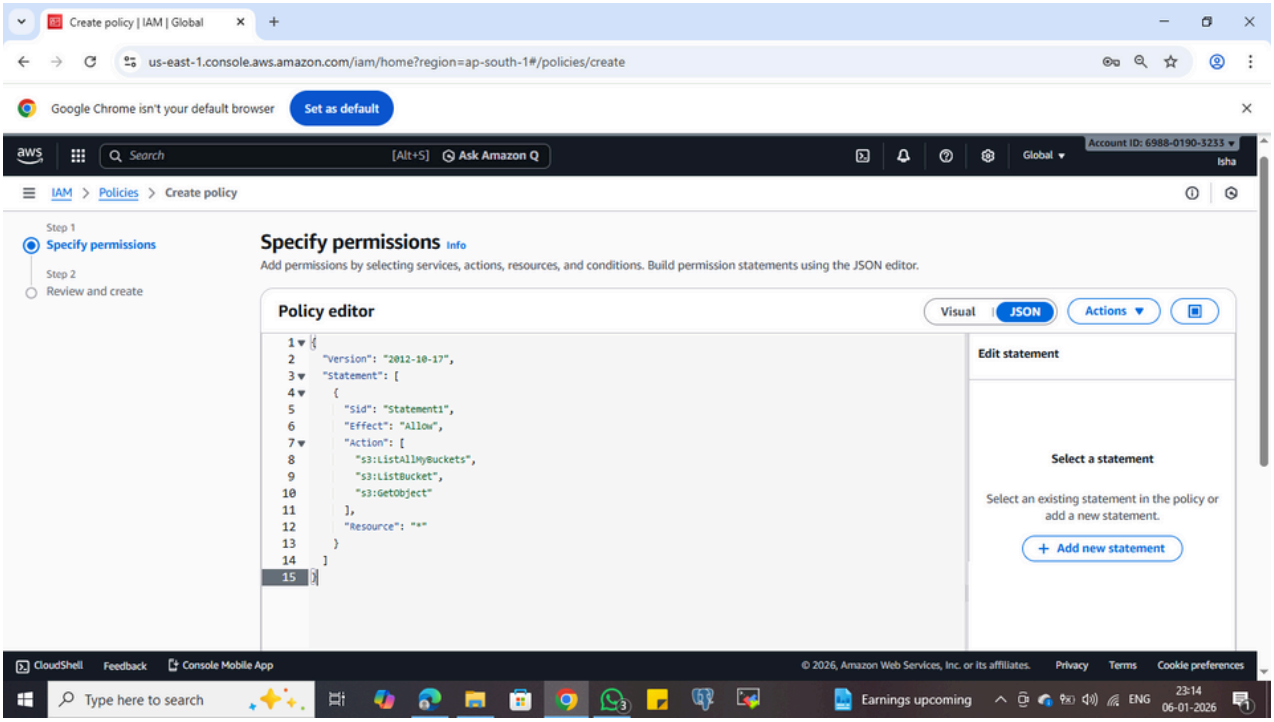
Click Add Statement



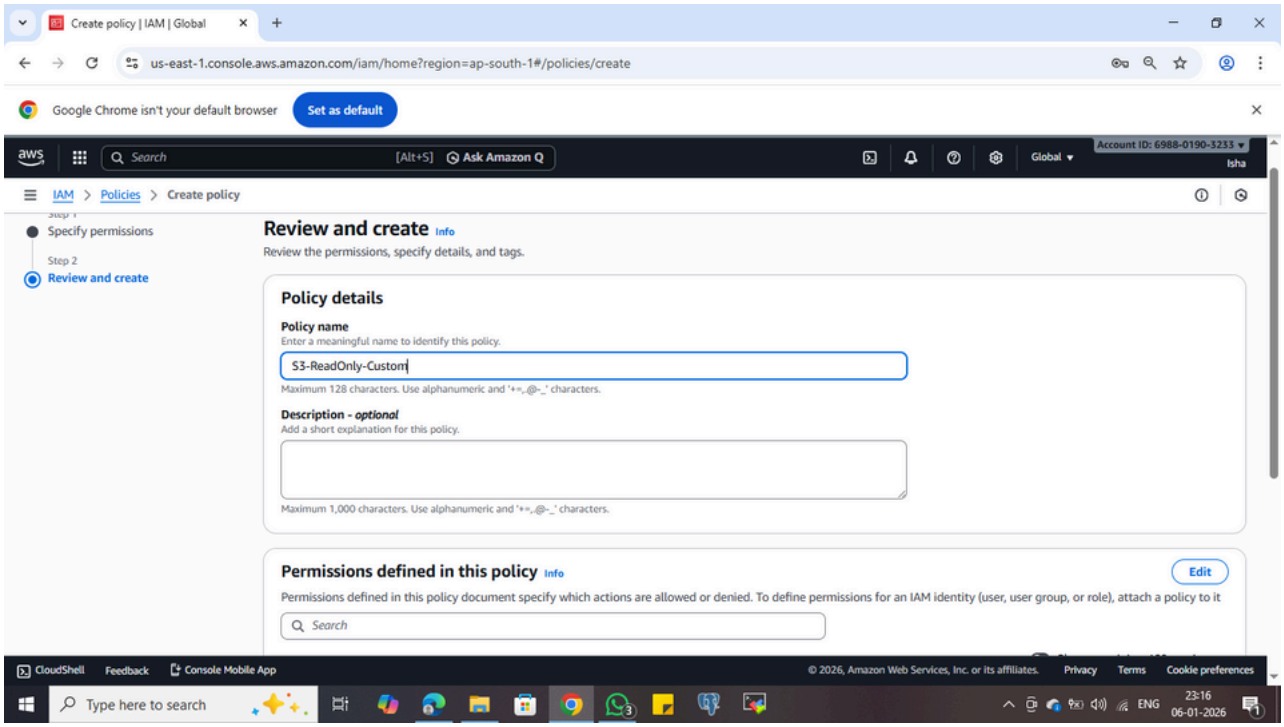
and then click the generate policy



copy the policy and paste it in the JSON tab



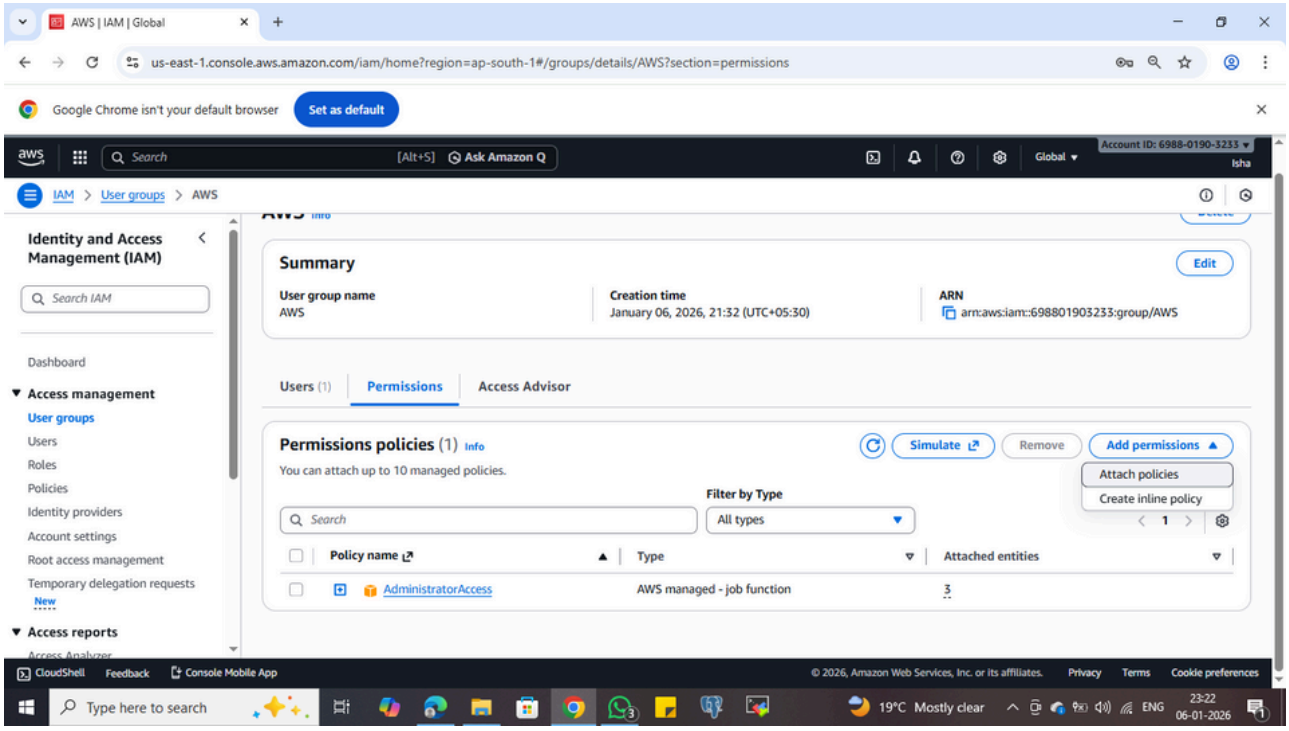
then next and set the policy name and click create policy.



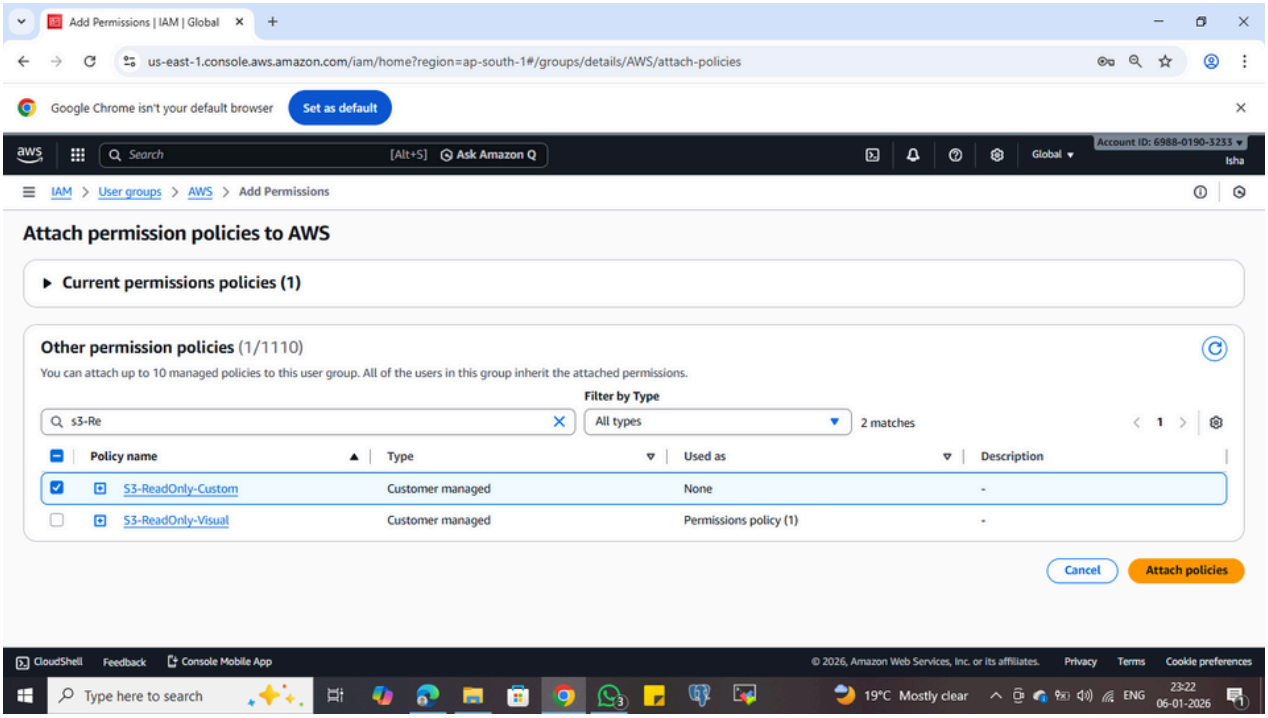
Now custom policy will be created successfully.

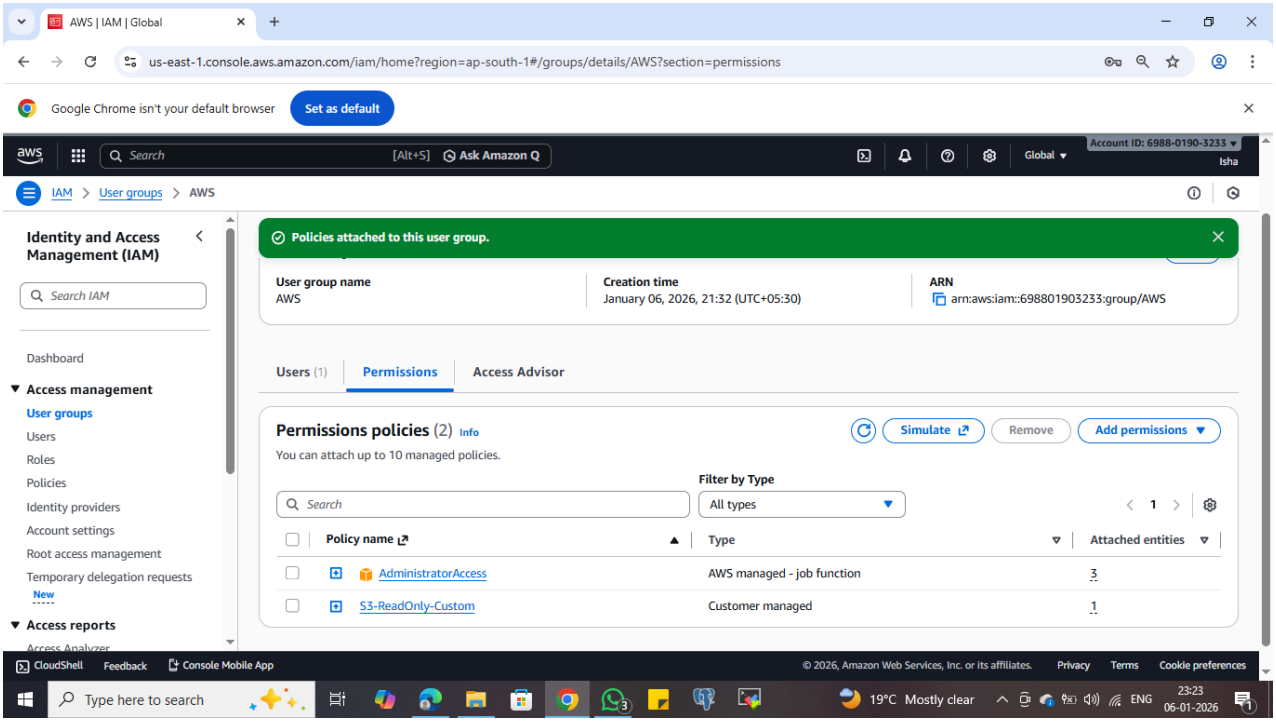
same as visual policy we can add this policy to the user or group. Now this time I add this policy to the group.

Go to IAM select your group**.** Open Permissions tab. Click Add permissions.



Choose Attach policies. Search S3-ReadOnly-Custom. Select the policy. Click Attach policies.





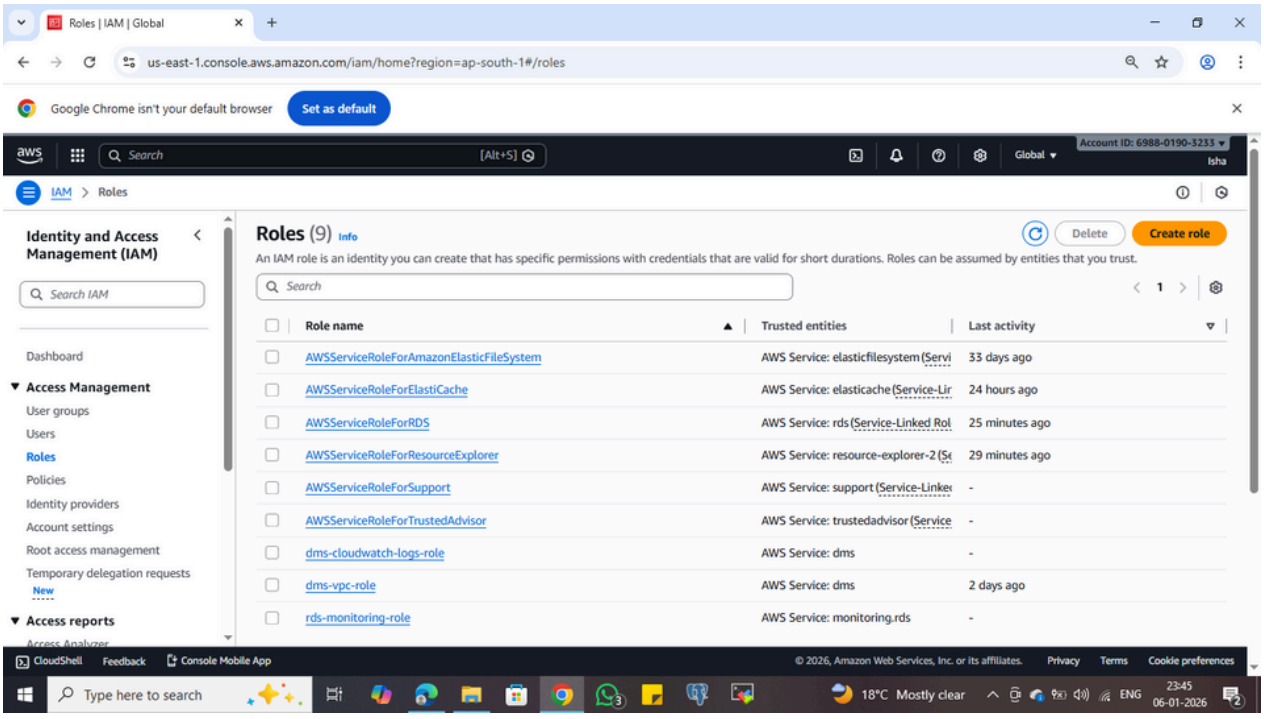
policy will be attached successfully.

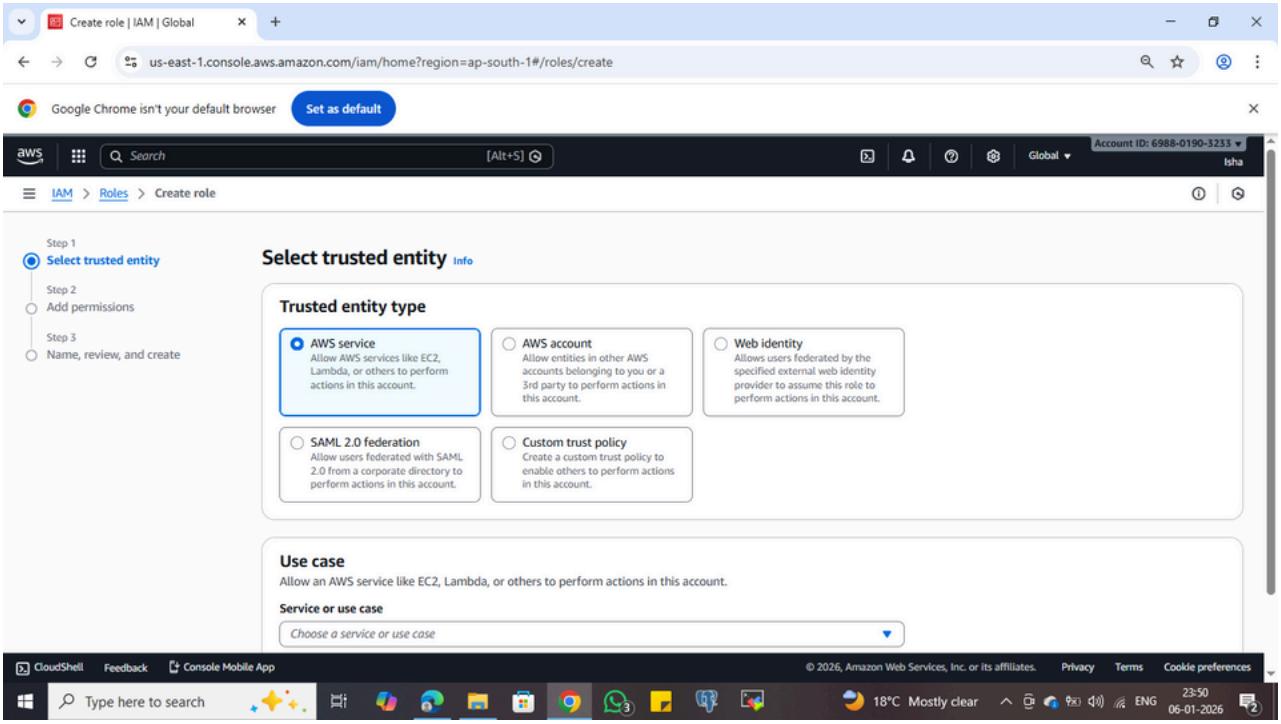
IAM policies can be created using either the JSON editor or the visual policy editor provided by AWS.

Practical 5: Create IAM Role for EC2

Role is used for communication between two services or two accounts. If we create a role and give permissions, then both can access each other.

Go to IAM then select roles then click create role





now we have to decide which one have to select to use

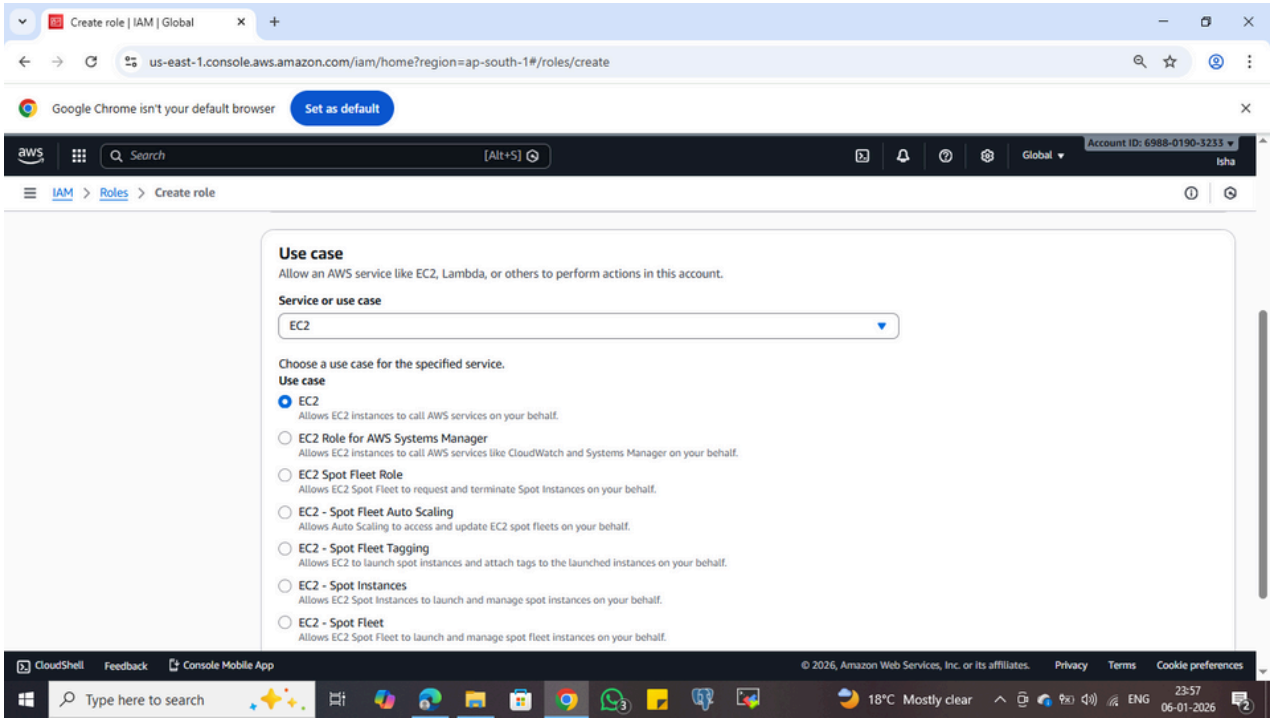
if we have to create a role for two services to communicate or access each other then we go with first one or else if we have to create a role for another account to access our account then we go with second one.

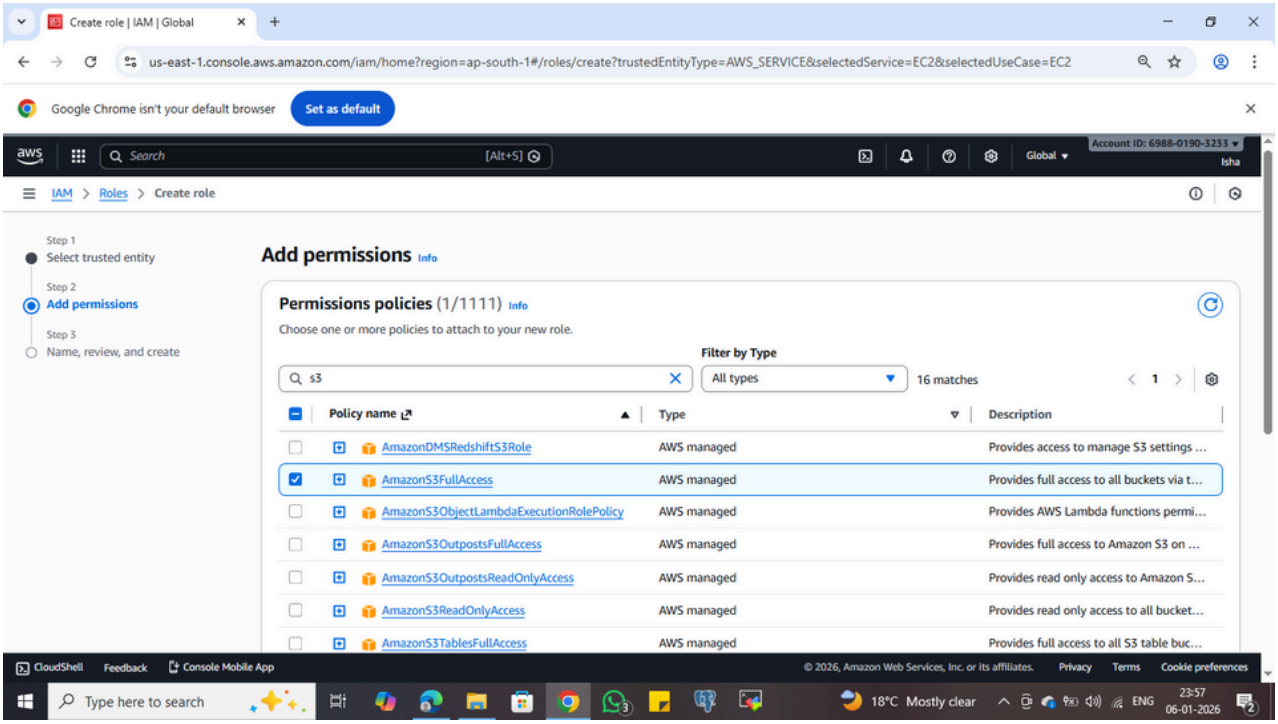
For ex, Ec2 wants to access s3 service now I'm selecting AWS service

then we have to mention trusted entity and permissions

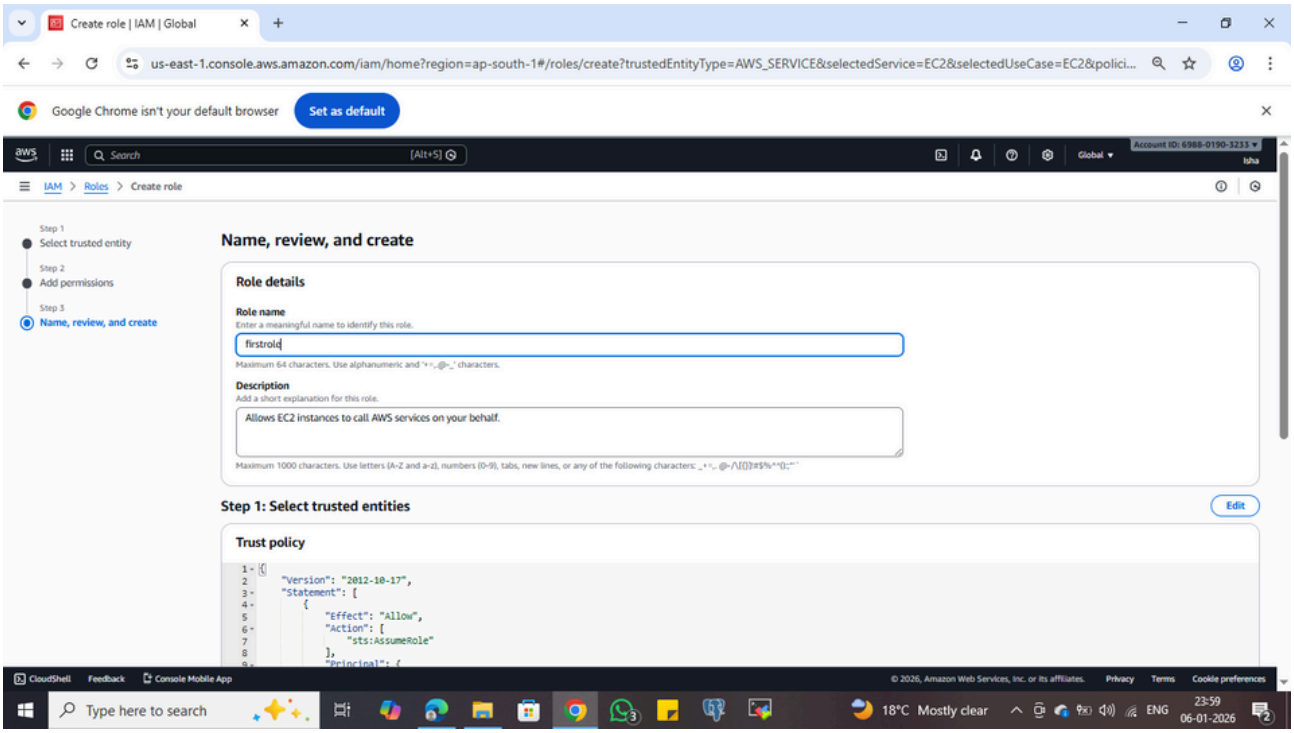
Trusted entity - which one have to attach

Permissions - to whom we are attach





Then set the role name and click create role. Then role is created successfully.



Then we have to attach this role to Ec2 to access the s3 service.

How I attach this role to Ec2 process I explained later after explanation about Ec2 service.

These are the simple basic practical's about IAM. I provide some tasks to do yourself to practice the IAM service.