Git Hub Link: https://github.com/charpitha-chavali/Security-in-Cloud-and-IOT/tree/main/06_Lab_1

1. Title:

Use IAM to implement user authentication in AWS. Set up rules to require Multi-Factor Authentication (MFA) for every user.

2. Objective:

The goal of this task is to establish secure user access in AWS by creating IAM users, assigning the appropriate permissions, and requiring Multi-Factor Authentication (MFA) for every account, thereby enhancing security against unauthorized access.

3. Problem Statement:

Different users with various roles require authorized access to AWS accounts. Even default credential-based authentication (password-only) is prone to phishing, brute-force attacks, and credential theft. To guarantee that only authorized users can safely access AWS services, IAM-based identity management must be implemented, and MFA must be enforced.

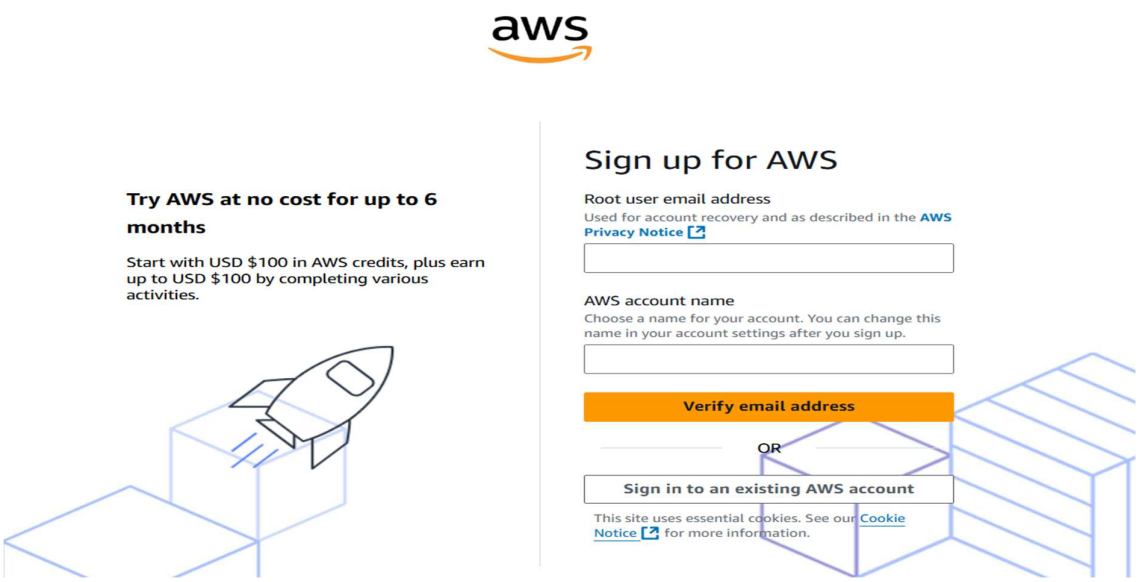
4. Implementation

AWS SIGN-UP GUIDE

Creating an AWS account involves several verification steps to ensure the account is linked to the correct user and can be billed properly. This document provides a clear, step-by-step explanation of the process for new users.

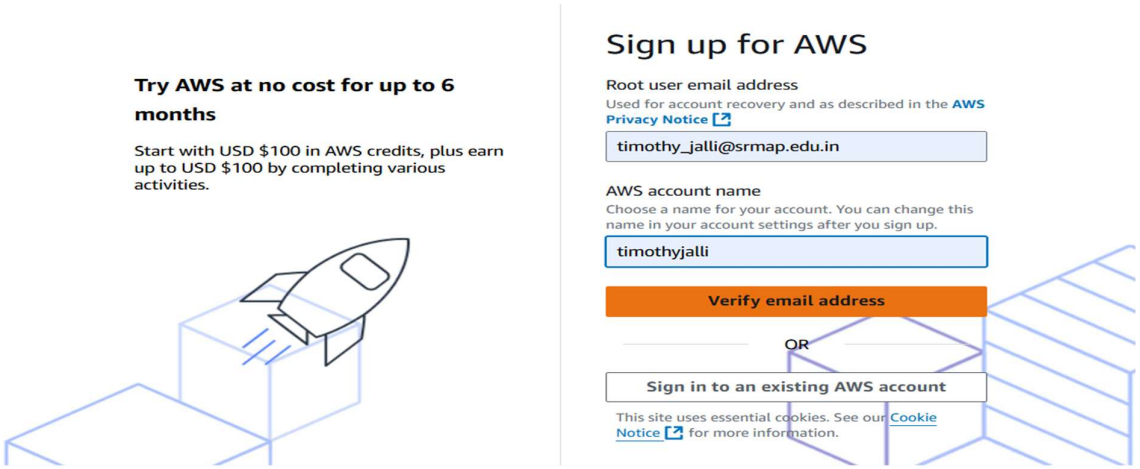
1. Visit the AWS Sign-Up Page:

To begin, visit the official AWS sign-up website and create an account.



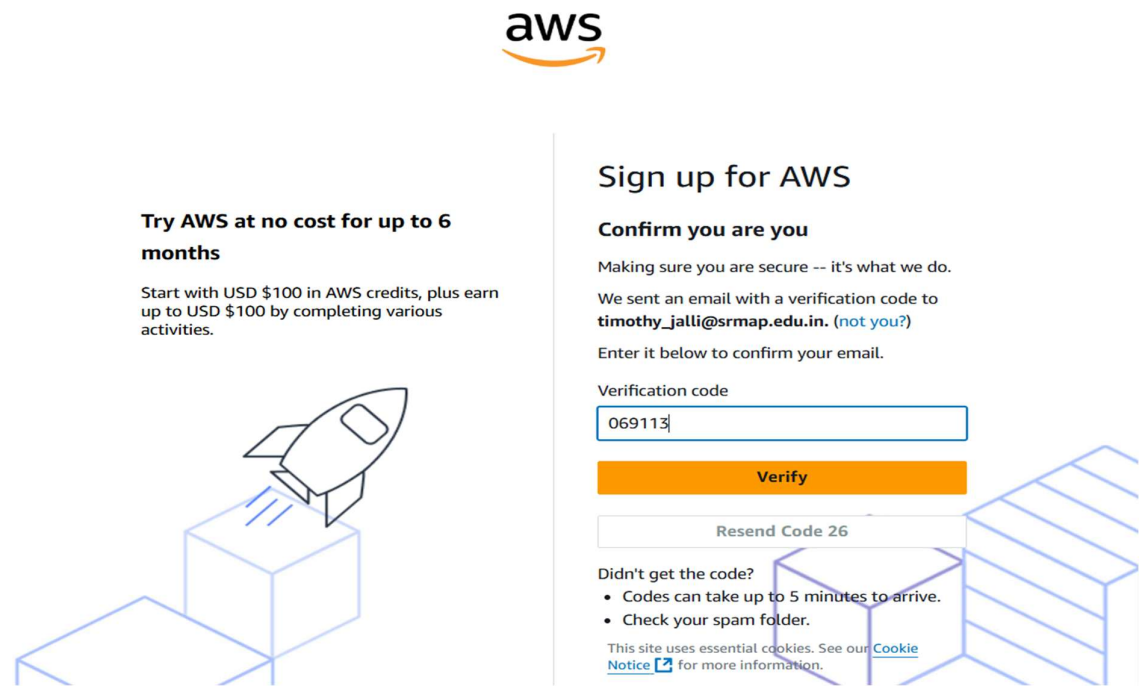
2. Enter Contact Information

Fill in your name, email as prompted.



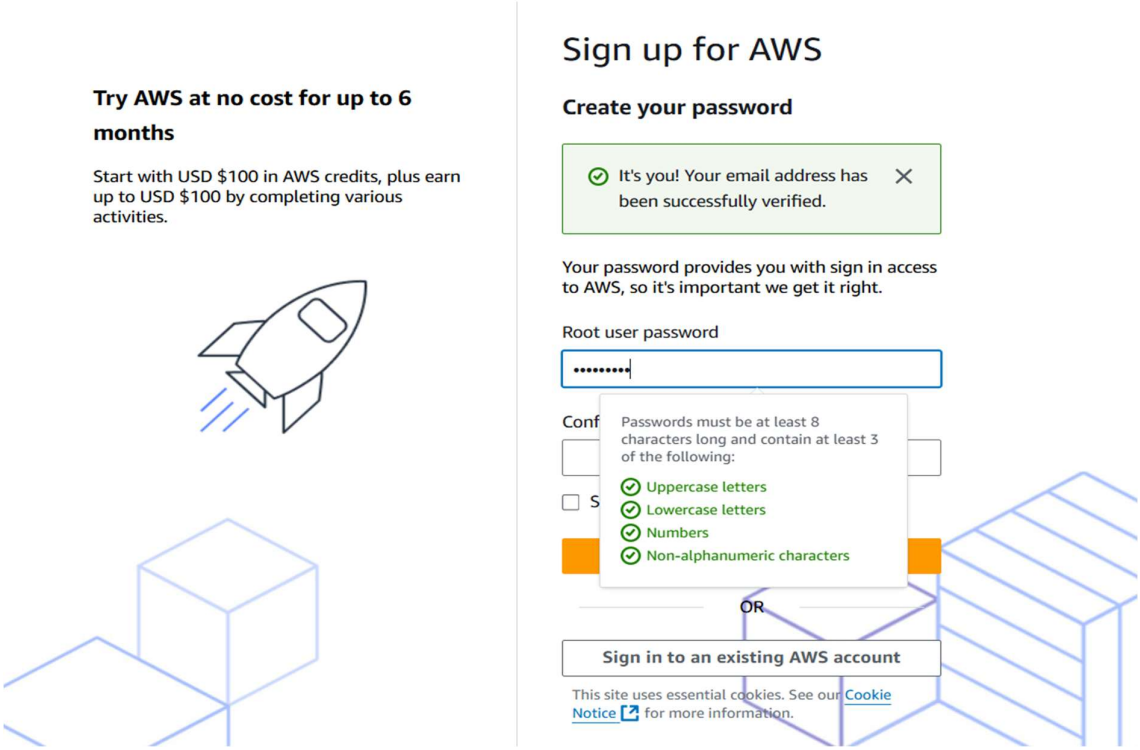
3. Verify Email

A verification code will be sent to your email address. Enter the code to proceed.



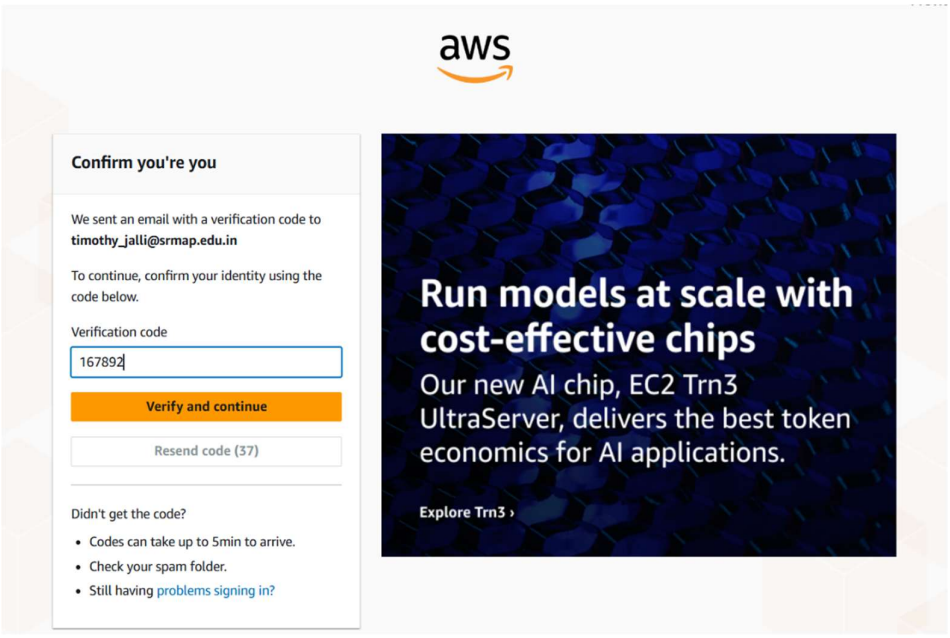
4. Create Root Password

Set a secure password following AWS password requirements.



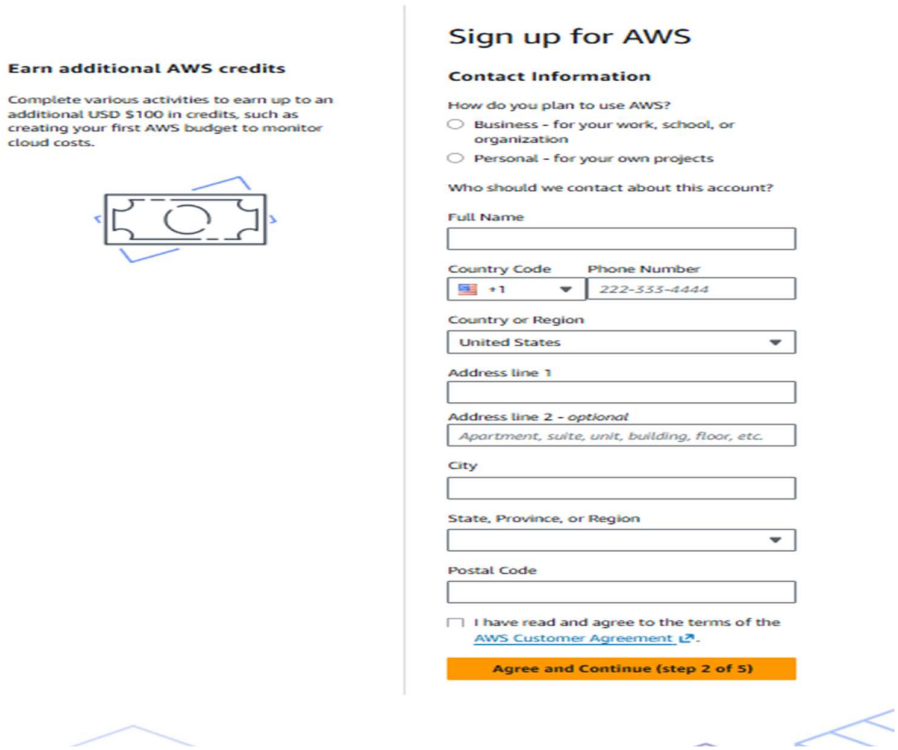
5. Identity Confirmation

Confirm your identity again using a verification code sent to your email.



6. Choose Account Type

Select Personal or Business based on usage.

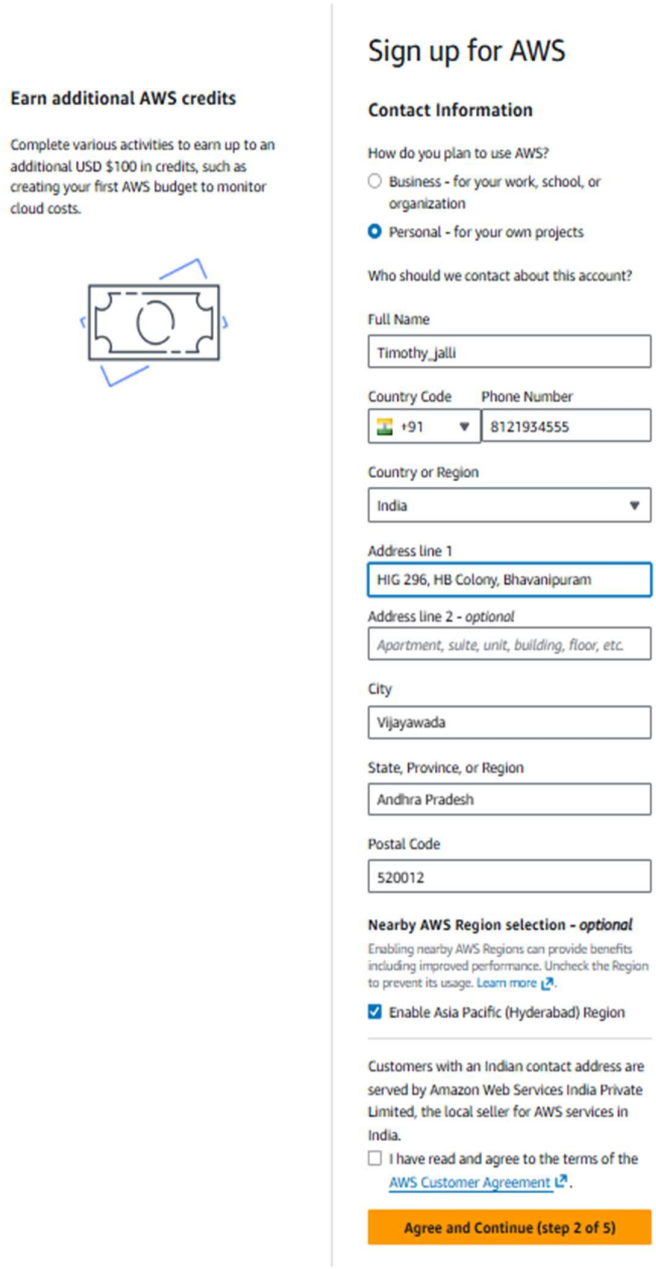


7. Provide Address & Phone Details

Enter:

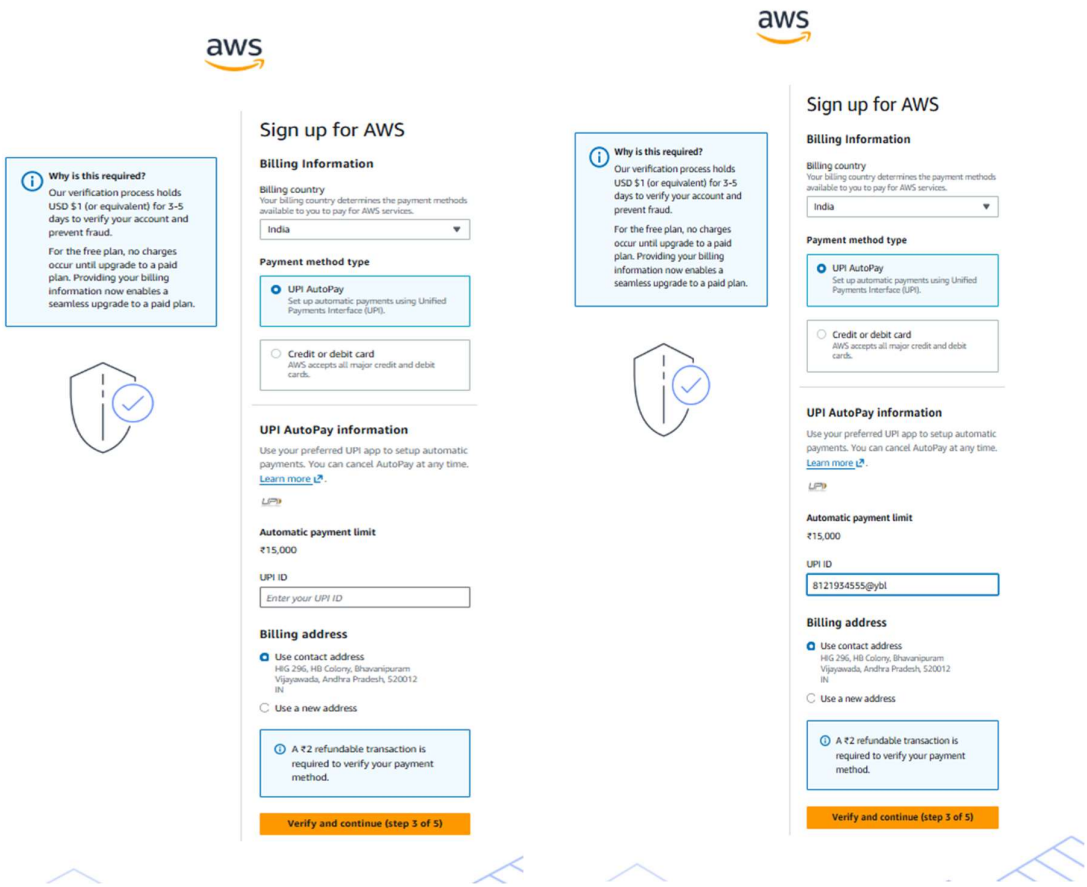
- Full name
- Country
- Address
- Phone number

Agree to terms to continue.



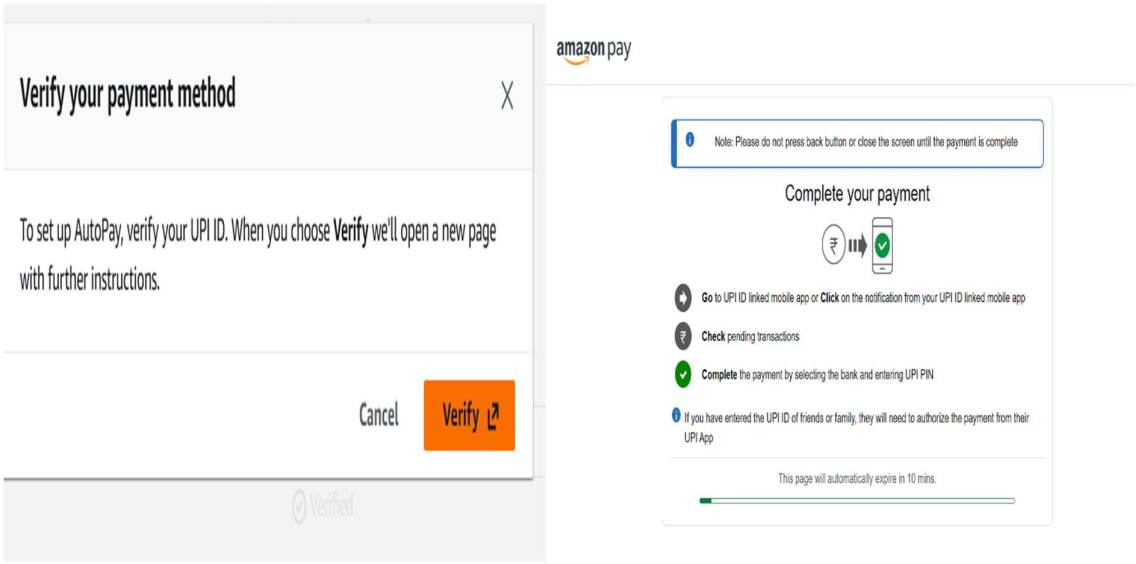
8. Payment Verification

Decide on a payment option. UPI Autopay is available for India.
Use your UPI app to verify payment after entering your UPI ID.



9. Account Successfully Created

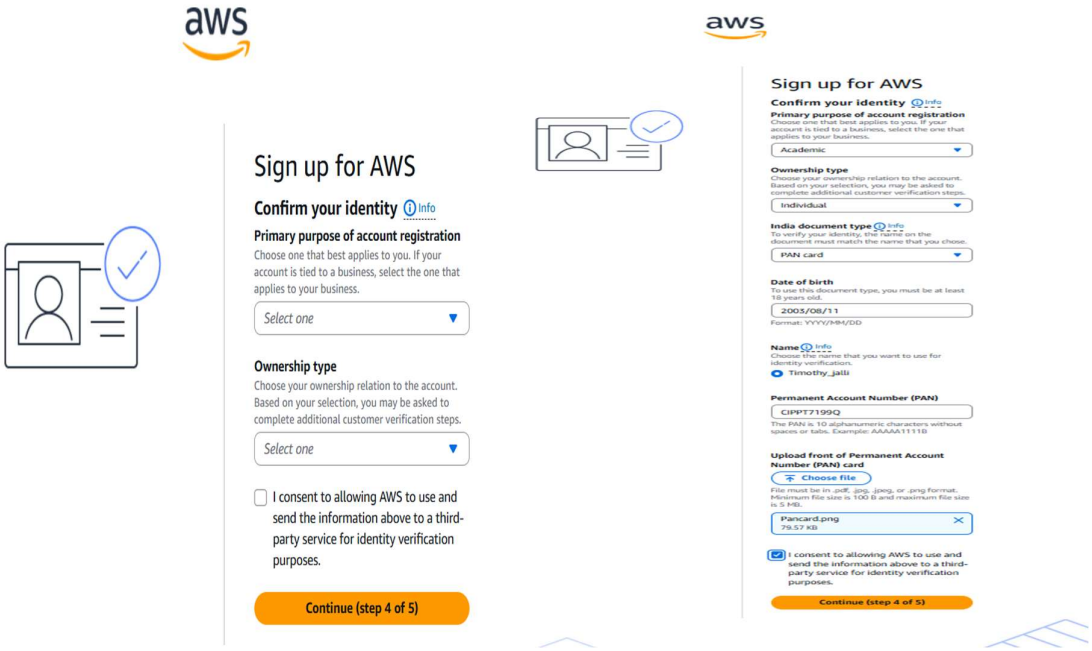
AWS verifies that your account was successfully created and authorized after completing UPI verification.



10.Complete Identity Verification (PAN)

You will be asked to verify your identity using a government-issued document such as PAN.
Steps:

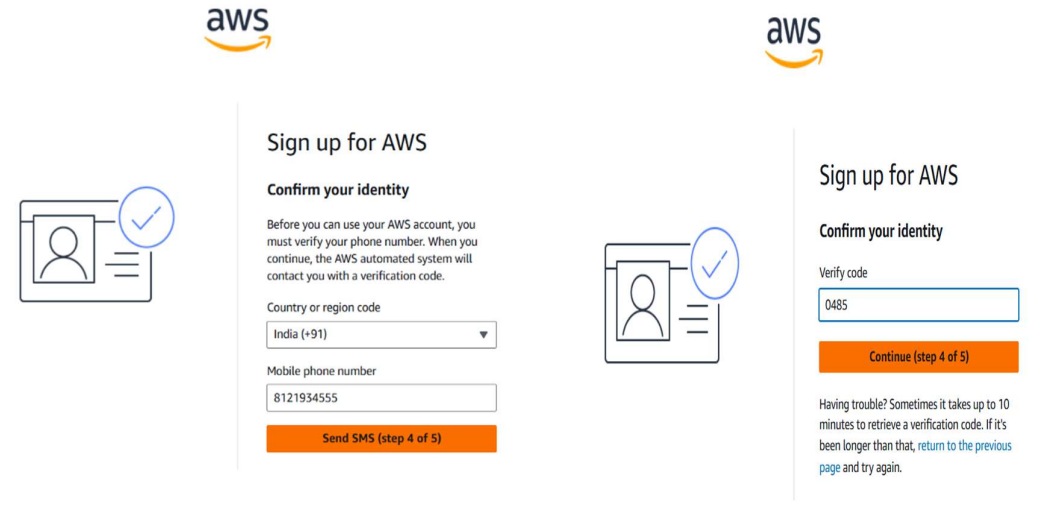
- Select Primary purpose (e.g., Academic)
- Select Ownership type (e.g., Individual)
- Select Document type (e.g., PAN Card)
- Enter Date of birth
- Upload front image of PAN card
- Provide consent and continue



11.Phone Number Verification

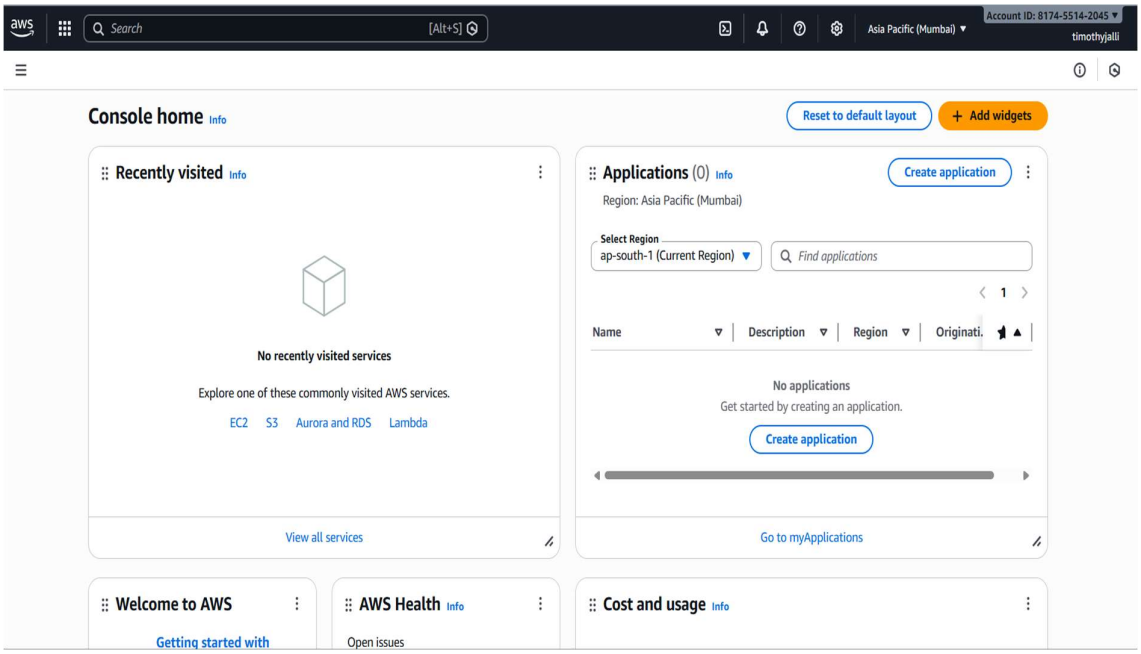
AWS will request your mobile number for verification.

- Enter country code
- Enter mobile number
- Click Send SMS
- Enter the OTP received and continue.



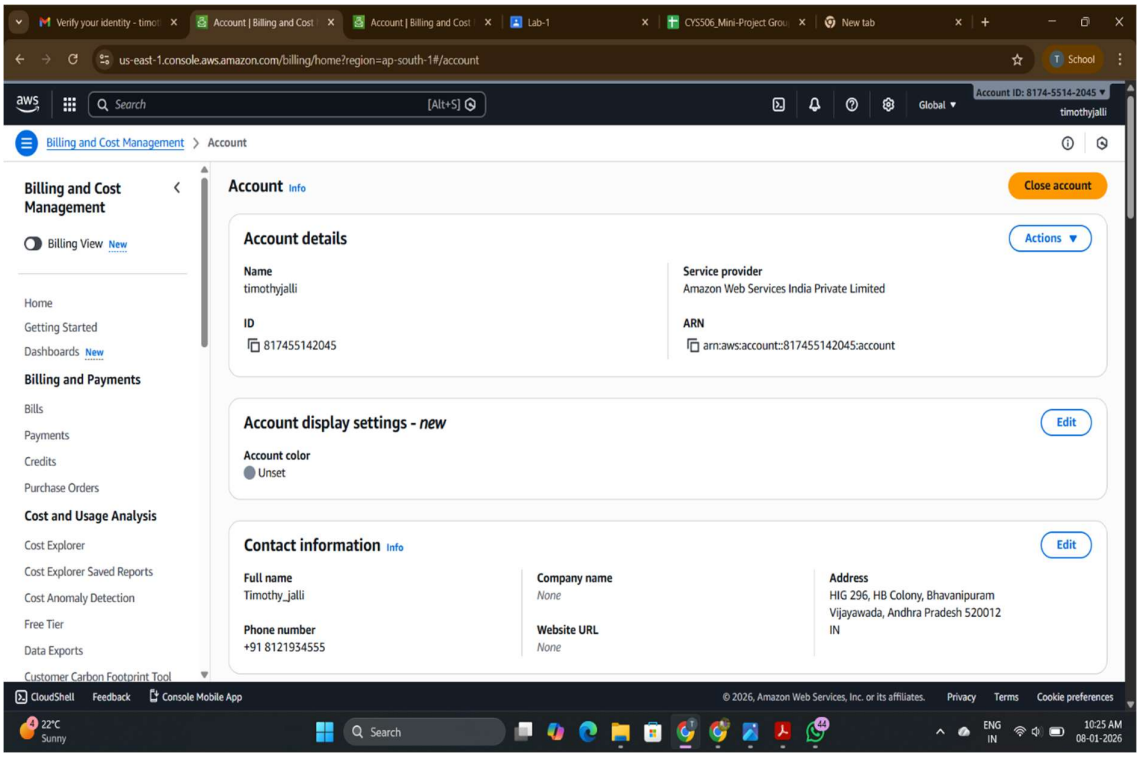
12.AWS Console Access

Once identity verification is completed, you can log in to AWS Management Console.
The console provides dashboards for services such as EC2, S3, Lambda, etc.



13.Billing & Account Management

After login, you can view your account details, billing, credits, payment methods, and address under Billing and Cost Management.

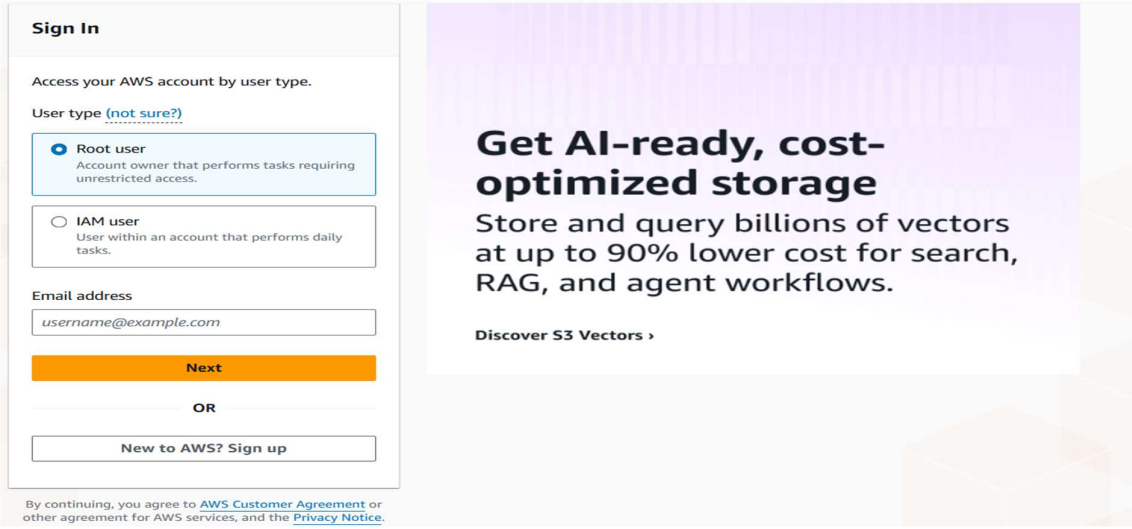


14.AWS Login Options

Once everything is set up, AWS gives you two ways to sign in:

- **Root User** – Full administrative access for managing billing, credentials, and critical configurations.
- **IAM User** – Restricted access for daily tasks without exposing sensitive account control.

This completes the AWS sign-up and onboarding flow.



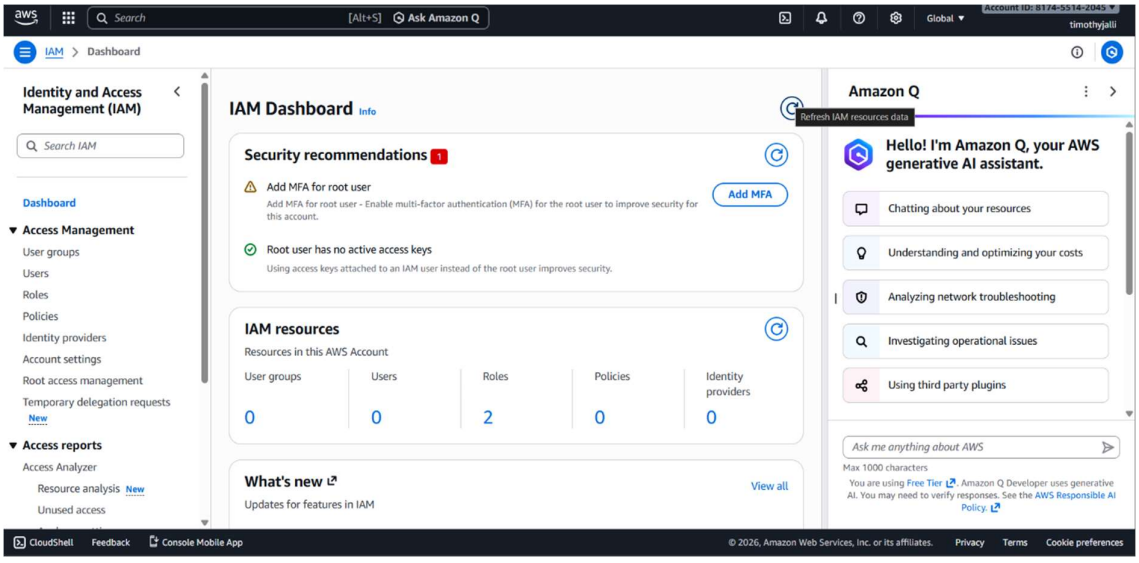
15.Turn On MFA

AWS asks you to turn on Multi-Factor Authentication (MFA) to make your account safer.
MFA means you need two things to log in:

- Your password
- A code from your phone

To turn it on:

1. Open IAM Dashboard
2. Click Add MFA
3. Choose the phone/authenticator option
4. Scan the QR code and enter the code



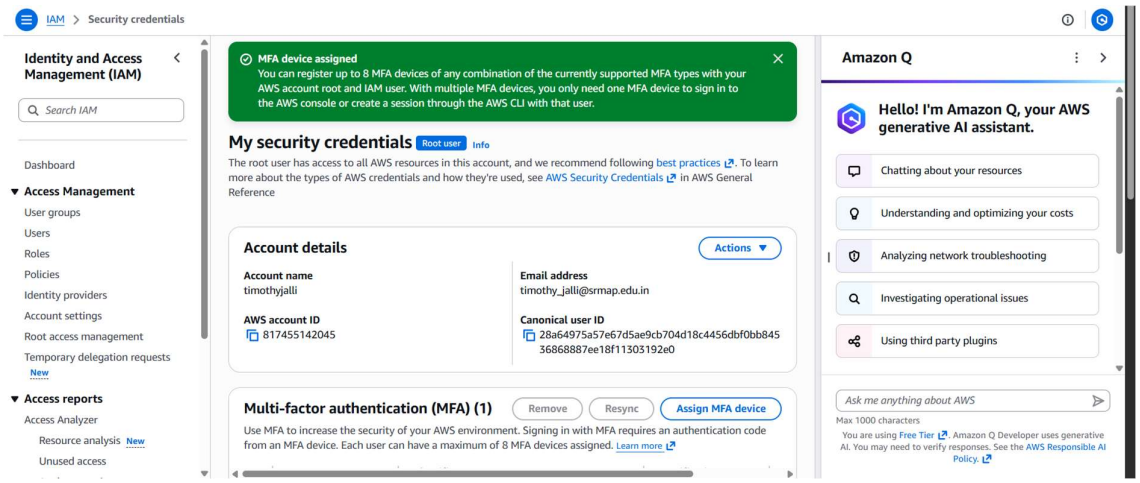
16.MFA Finished

MFA is now added to the root user.

From now on, when logging in as root you must enter:

- Password
- Phone code

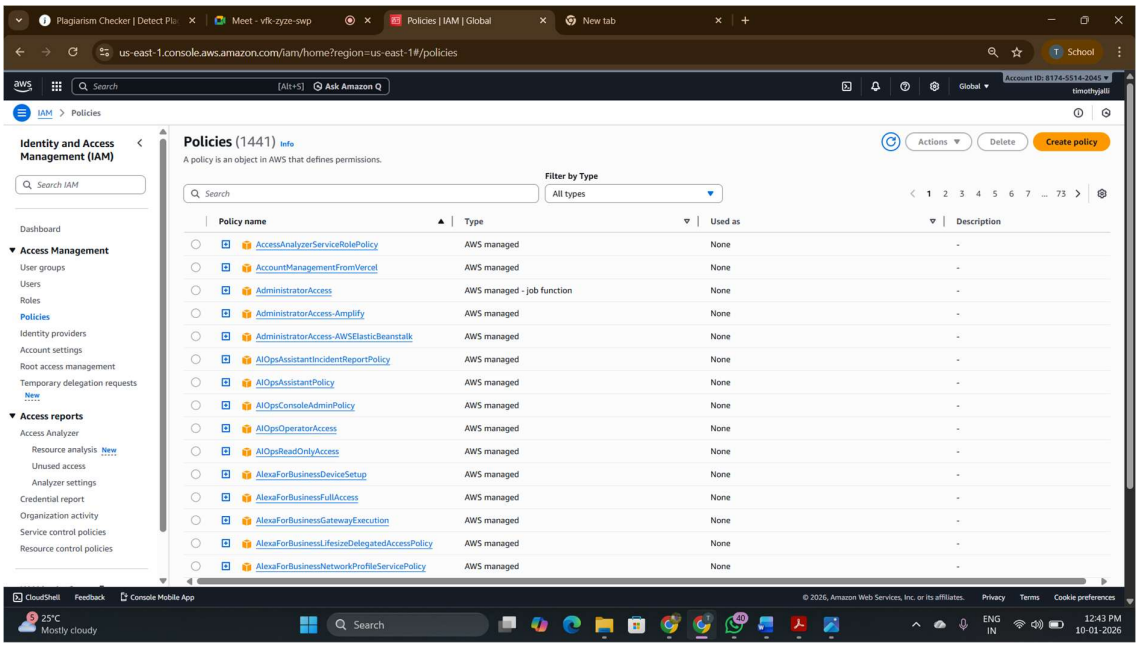
This keeps your AWS account more secure.



CREATING THE MFA ENFORCEMENT POLICY

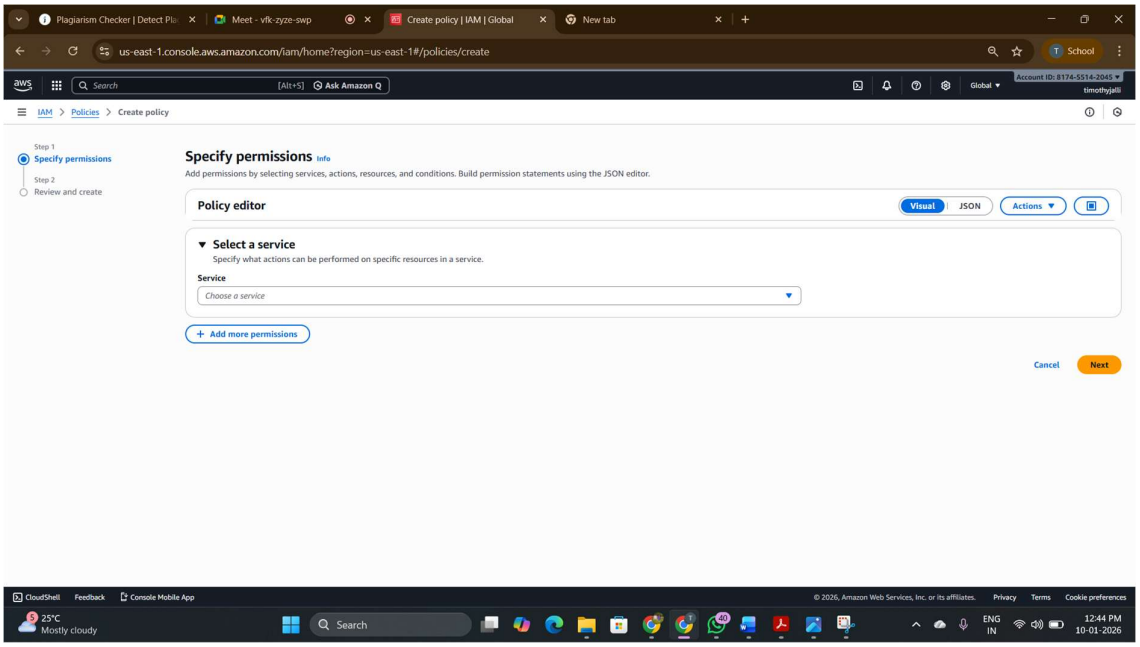
1. Open IAM Policies

From the AWS Management Console, I opened the **IAM** service and clicked on the **Policies** section to view existing IAM policies.



2. Create a New Policy

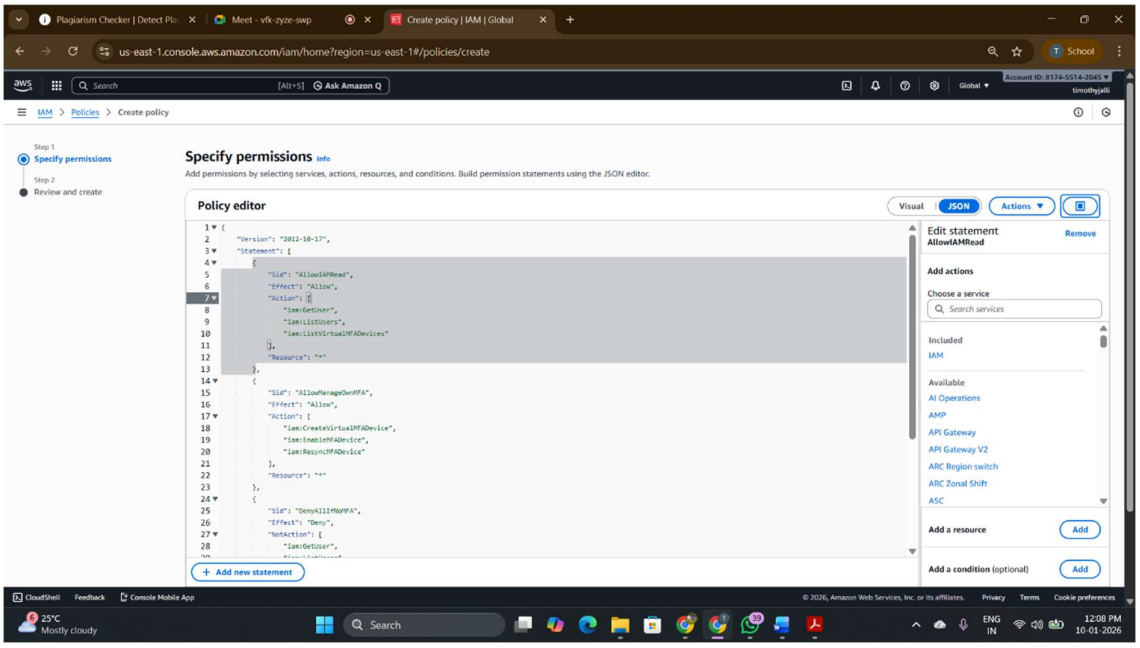
Next, I clicked on the **Create policy** button to start building a new custom IAM policy that will enforce MFA usage.



3. Specify Permissions Using JSON

On the **Specify permissions** page, I switched to the **JSON** tab. In the JSON editor, I added the JSON code that defines the permissions for MFA.

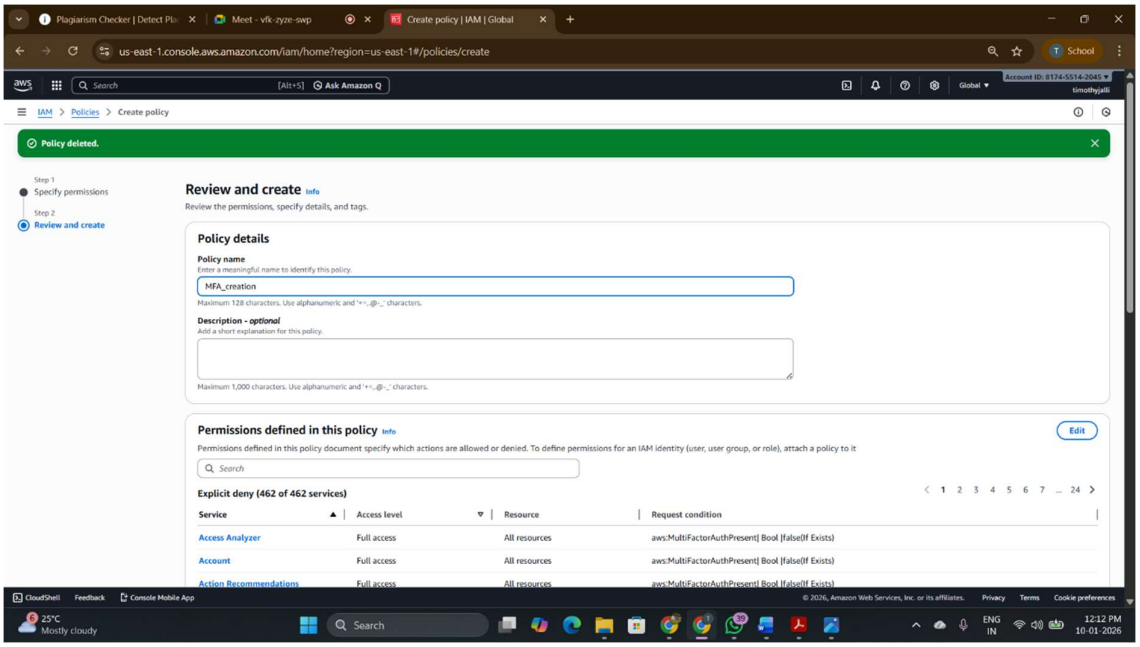
This JSON allows users to view their IAM details, configure their own MFA device, and denies access to other services if MFA is not enabled.



4. Review and Name the Policy

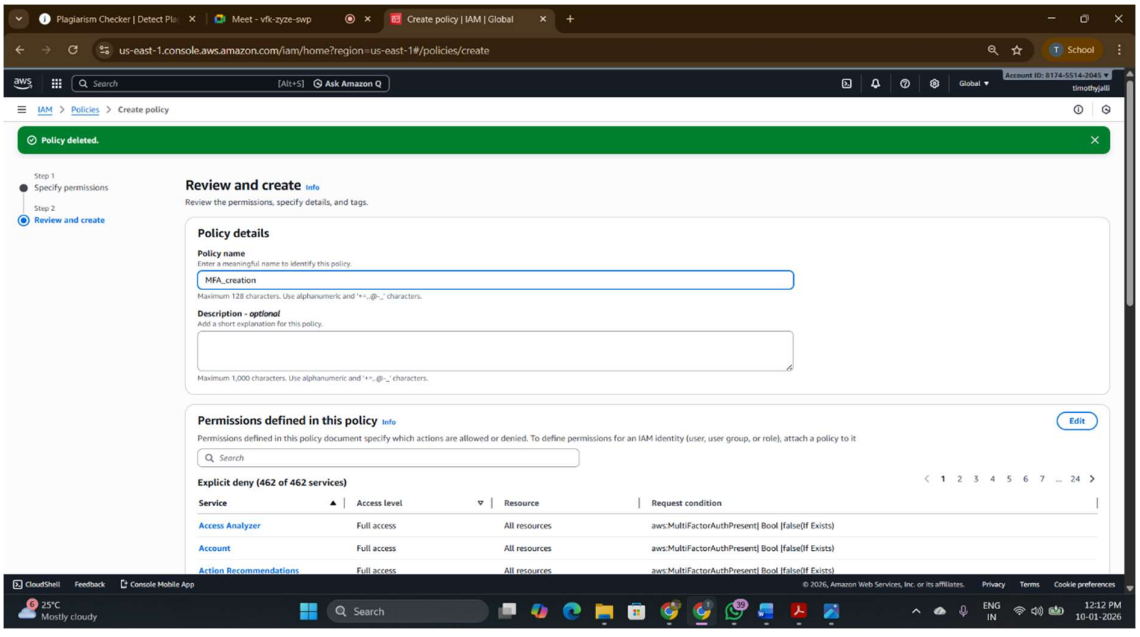
After adding the JSON, I clicked on **Next: Review**.

On the review page, I provided a name for the policy as **MFA_creation**. Adding a description was optional, so I left it blank.



5. Create the Policy

After reviewing the configuration, I clicked on **Create policy**. A confirmation message appeared at the top showing **Policy MFA_creation created**, which confirmed that the policy was successfully created.

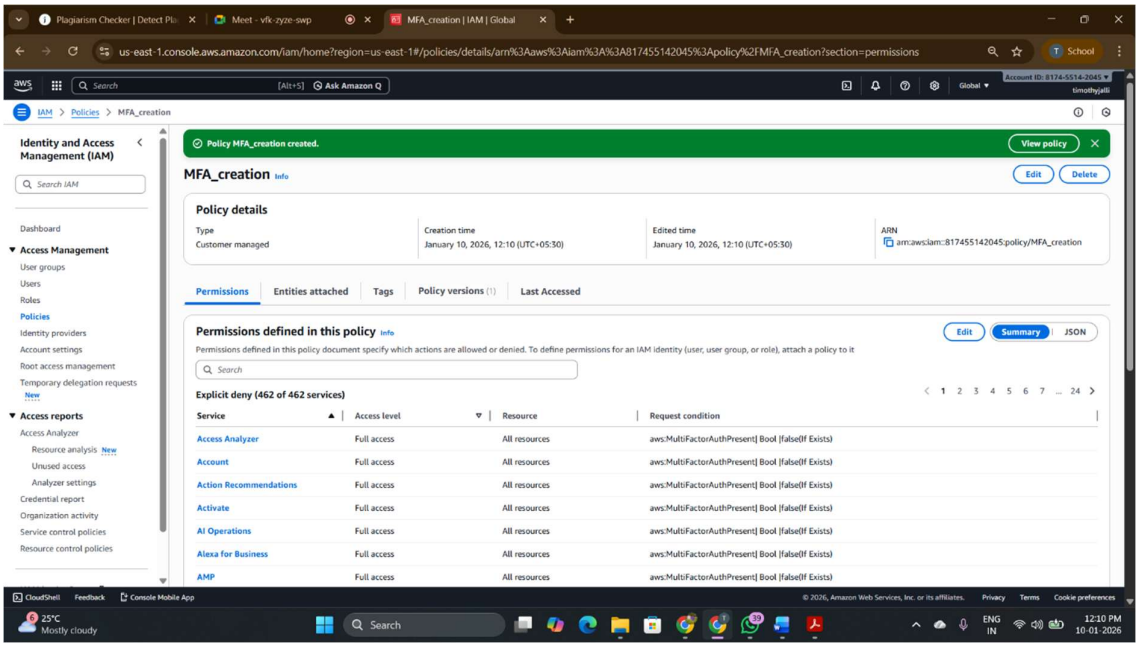


6. Verify Policy Details

I have accessed the policy subsequent to its creation in order to verify the information. The policy described the following information:

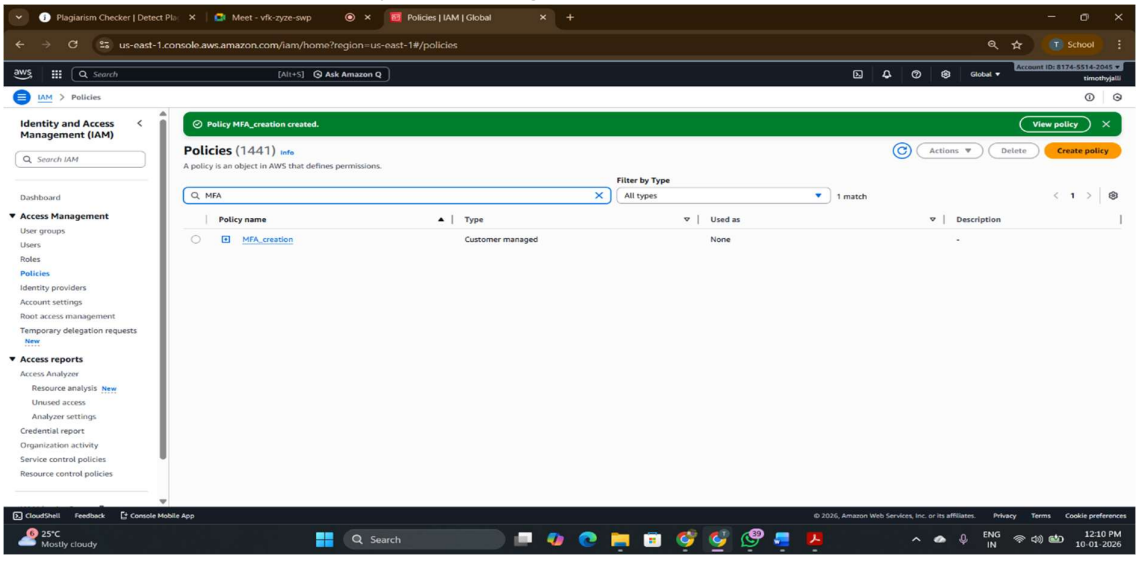
- Policy Name: MFA_creation
- Type: Customer-managed
- Permissions - Explicitly denies access to all services in case MFA is not activated.

The permissions tab listed the AWS services that would require MFA before users could access them.



7. Policy Ready for Use

Once the policy successfully created, it is now ready to be attached to IAM users or groups so that MFA becomes mandatory for accessing AWS resources.



5. Problems Faced During Development:

Various challenges arose during the implementation process. It took more time to sort out IAM policies and actually implement permissions between the different user roles. Also, there was confusion over the differentiation in the application of MFA between the root account and the normal IAM users. The application of MFA added to the delay because users had to pair their authenticator apps and manage time-based codes. After MFA was turned on, some users encountered some problems in logging in due to a lack of readily available verification codes. Also, some sections of the AWS console were not accessible until the correct policies were attached.

6. Conclusion:

Employing IAM in a configuration that implements MFA provides for added security when it comes to AWS environments. In the event of password compromise, unauthorized access is somewhat minimized because a second verification step is still required. IAM further allows detailed access control to be specified, including which AWS actions a principal-which may be an end user in a business or an application-can and cannot do. This prevents both unintentional error and intentional misuse. Overall, it follows the well-documented best practice of cloud security, protecting the critical cloud resource.