

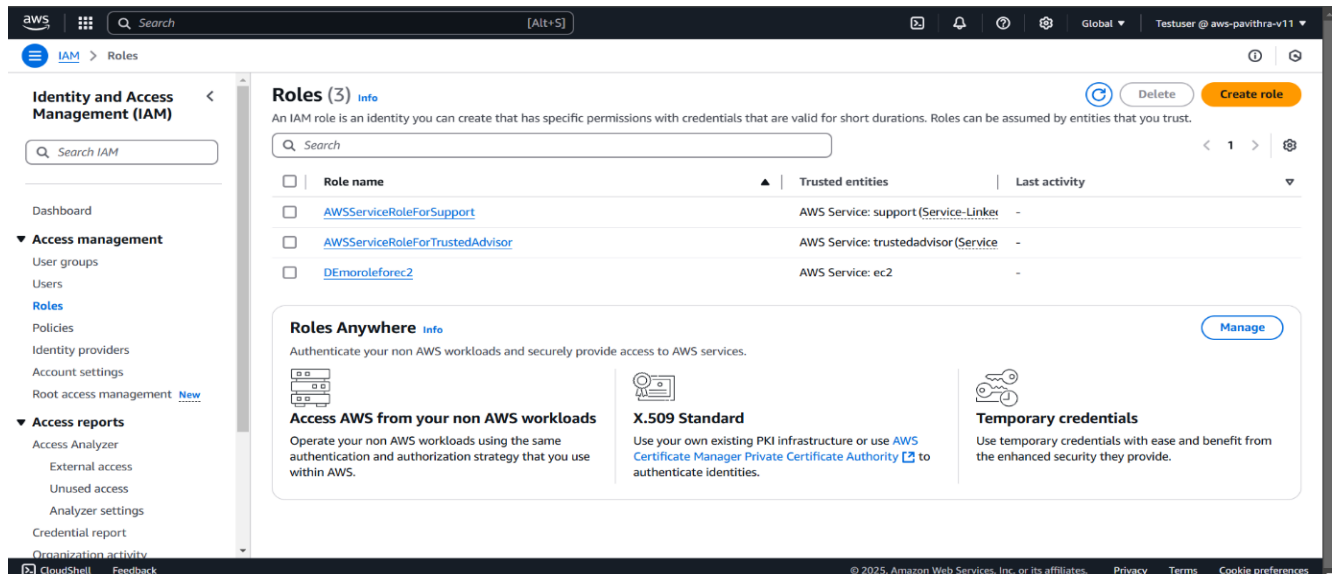
Securing AWS Environment for Workload Migration

Part 1: Create IAM Roles for Developers

IAM roles allow controlled access to AWS resources without sharing credentials.

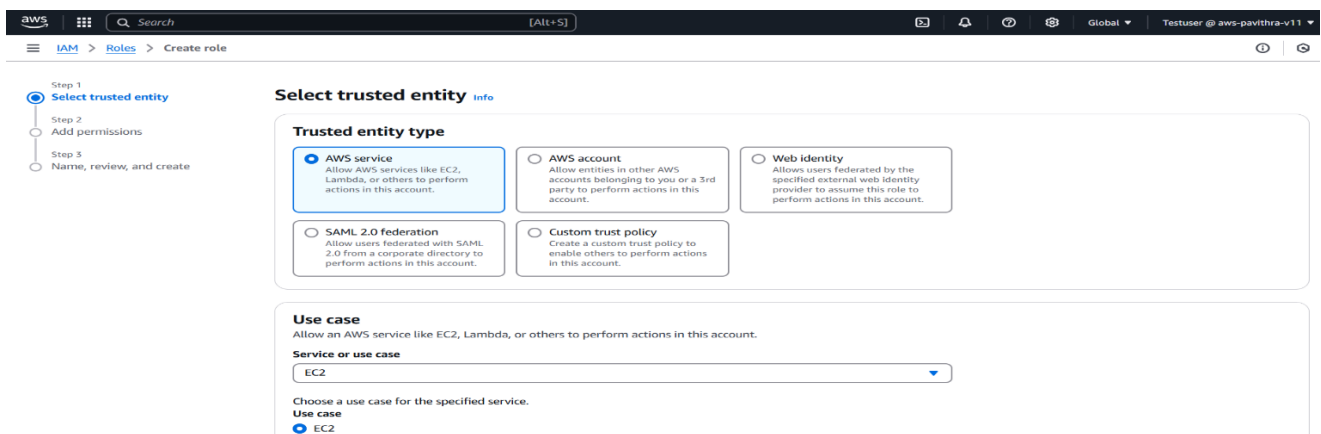
Step 1.1: Navigate to IAM Roles

1. Log in to the AWS Management Console.
2. Open the **IAM Dashboard** and select **Roles** from the navigation pane.
3. Click **Create role**.



Step 1.2: Define Trusted Entity

1. Choose a trusted entity type:
 - For applications: **AWS service** (e.g., EC2).
 - For users: **Another AWS account** or **IAM users**.
2. Click **Next**.



Step 1.3: Attach a Policy

1. Select an existing policy or create a custom policy:
 - For developers, attach **AmazonS3ReadOnlyAccess** for read-only access to S3.
 - For sensitive workloads, attach **AdministratorAccess** (restricted use).

2. Click **Next** and provide a **Role name** (e.g., DeveloperRole).

Step 1

Step 2

Step 3

Select trusted entity

Add permissions

Name, review, and create

Add permissions

Info

Permissions policies (2/1026)

Info

Choose one or more policies to attach to your new role.

Filter by Type

All types

20 matches

Q administrator

X

Policy name

Type

Description

☒

AdministratorAccess

AWS managed - job function

Provides full access to AWS services an...

☐

AdministratorAccess-Amplify

AWS managed

Grants account administrative permissi...

☐

AdministratorAccess-AWSElasticBeanstalk

AWS managed

Grants account administrative permissi...

☐

AmazonAPIGatewayAdministrator

AWS managed

Provides full access to create/edit/dele...

☐

AmazonSecurityLakeAdministrator

AWS managed

Provides full access to Amazon Securit...

☐

AWS-SSM-DiagnosisAutomation-Administrati...

AWS managed

Provide permission for Diagnosing issu...

☐

AWS-SSM-DiagnosisAutomation-Operational...

AWS managed

Provides permissions for operational a...

☐

AWS-SSM-RemediationAutomation-Administ...

AWS managed

Provide permission for Remediating iss...

aws

Search

[Alt+S]

Global

Testuser @ aws-pavithra-v11

IAM

Roles

Create role

Step 1

Step 2

Step 3

Select trusted entity

Add permissions

Name, review, and create

Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

Developer_role

Maximum 64 characters. Use alphanumeric and '+', '@', '_' characters.

Description

Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+@-./\[]!#\$%^&*(){};:'"`,~.:

Step 1: Select trusted entities

Edit

Trust policy

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "ec2:DescribeInstances"
8       ]
9     }
10  ]
11 }
```

aws

Search

[Alt+S]

Global

Pavithra_Athmanathan

IAM

Roles

Step 1.4: Review and Create

1. Verify details and click **Create role**.

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

Roles

Policies

Roles (4)

Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Search

1

Role name

Trusted entities

Last activity

☐

[AWSServiceRoleForSupport](#)

AWS Service: support

(Service-Linker)

-

☐

[AWSServiceRoleForTrustedAdvisor](#)

AWS Service: trustedadvisor

(Service)

-

☐

[DEmoroleforec2](#)

AWS Service: ec2

-

☐

[Developer_role](#)

AWS Service: ec2

-

Part 2: Define IAM Policies

IAM policies restrict or grant access to AWS services.

Step 2.1: Create a Custom Policy

1. Navigate to **IAM Policies** and click **Create policy**.

Define permissions in JSON format:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "s3:*",  
      "Resource": "arn:aws:s3:::example-bucket/*"  
    }  
  ]  
}
```

- 2.
3. Click **Review policy**, provide a name (e.g., S3FullAccess), and save.

The screenshot shows the AWS IAM console interface for creating a new policy. The breadcrumb navigation indicates the path: IAM > Policies > Create policy. The page is divided into two main sections: a left sidebar with step indicators and a main content area.

Step 1: Specify permissions is the active step. Below it, **Step 2: Review and create** is visible. The main content area is titled **Specify permissions** with an **Info** icon. A sub-header reads: "Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor."

The **Policy editor** is the central focus, featuring a **Visual** tab and an active **JSON** tab. The JSON editor displays the following policy document:

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": "s3:*",  
7       "Resource": [  
8         "arn:aws:s3:::spavi-bckt",  
9         "arn:aws:s3:::pavi-bckt/*"  
10      ]  
11    }  
12  ]  
13 }
```

On the right side of the editor, the **Edit statement** panel is visible. It contains the heading **Select a statement** and the instruction: "Select an existing statement in the policy or add a new statement." Below this instruction is a button labeled **+ Add new statement**.

condition. For details, choose **Show remaining**. [Learn more](#)

Permissions defined in this policy [Info](#) [Edit](#)

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it

Allow (1 of 437 services) Show remaining 436 services

Service	Access level	Resource	Request condition
S3	Limited: List, Permissions management, Read, Write, Tagging	BucketName string like [example-bucket, ObjectPath] string like [All]	None

Add tags - optional [Info](#)

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tags.

[Cancel](#) [Previous](#) [Create policy](#)

Step 1 Specify permissions

Step 2 **Review and create**

Review and create [Info](#)

Review the permissions, specify details, and tags.

Policy details

Policy name
Enter a meaningful name to identify this policy.

Maximum 128 characters. Use alphanumeric and '+@_.' characters.

Description - optional
Add a short explanation for this policy.

Maximum 1,000 characters. Use alphanumeric and '+@_.' characters.

Permissions defined in this policy [Info](#) [Edit](#)

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it

Policy Pavithra_policy created. [View policy](#)

Policies (1324) [Info](#)

A policy is an object in AWS that defines permissions.

Filter by Type 1 match

	Policy name	Type	Used as	Description
<input type="radio"/>	Pavithra_policy	Customer managed	None	-

Identity and Access Management (IAM)

Dashboard

Access management

- User groups
- Users
- Roles
- Policies**
- Identity providers
- Account settings
- Root access management [New](#)

Access reports

Step 2.2: Attach Policy to a Role or User

1. Go to **Roles** or **Users** in IAM.
2. Attach the newly created policy (e.g., S3FullAccess) to the required entity.

aws [Search] [Alt+S] Global Testuser @ aws-pavithra-v11

IAM > Users > Testuser > Add permissions

Step 1 Add permissions
Step 2 Review

Add permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

- ☐ Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- ☐ Copy permissions
Copy all group memberships, attached managed policies, inline policies, and any existing permissions boundaries from an existing user.
- ☒ Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1323)

Filter by Type: All types 1 match

Search: pav

<input checked="" type="checkbox"/>	Policy name	Type	Attached entities
<input checked="" type="checkbox"/>	Pavithra_policy	Customer managed	0

Cancel Next

aws [Search] [Alt+S] Global Testuser @ aws-pavithra-v11

IAM > Users > Testuser

Identity and Access Management (IAM)

Search IAM

Dashboard

▼ Access management

- User groups
- Users**
- Roles
- Policies
- Identity providers
- Account settings
- Root access management [New](#)

▼ Access reports

- Access Analyzer
- External access
- Unused access
- Analyzer settings
- Credential report
- Organization activity

1 policy added

Permissions Groups (1) Tags (1) Security credentials Last Accessed

Permissions policies (7)

Permissions are defined by policies attached to the user directly or through groups.

Filter by Type: All types

Search

<input type="checkbox"/>	Policy name	Type	Attached via
<input type="checkbox"/>	AdministratorAccess	AWS managed - job function	Group User
<input type="checkbox"/>	AdministratorAccess-Amplify	AWS managed	Group User
<input type="checkbox"/>	AdministratorAccess-AWSElasticBeanstalk	AWS managed	Group User
<input type="checkbox"/>	AmazonS3FullAccess	AWS managed	Group User
<input type="checkbox"/>	Pavithra_policy	Customer managed	Directly
<input type="checkbox"/>	PowerUserAccess	AWS managed - job function	Group User
<input type="checkbox"/>	S3fullaccess	Customer managed	Directly

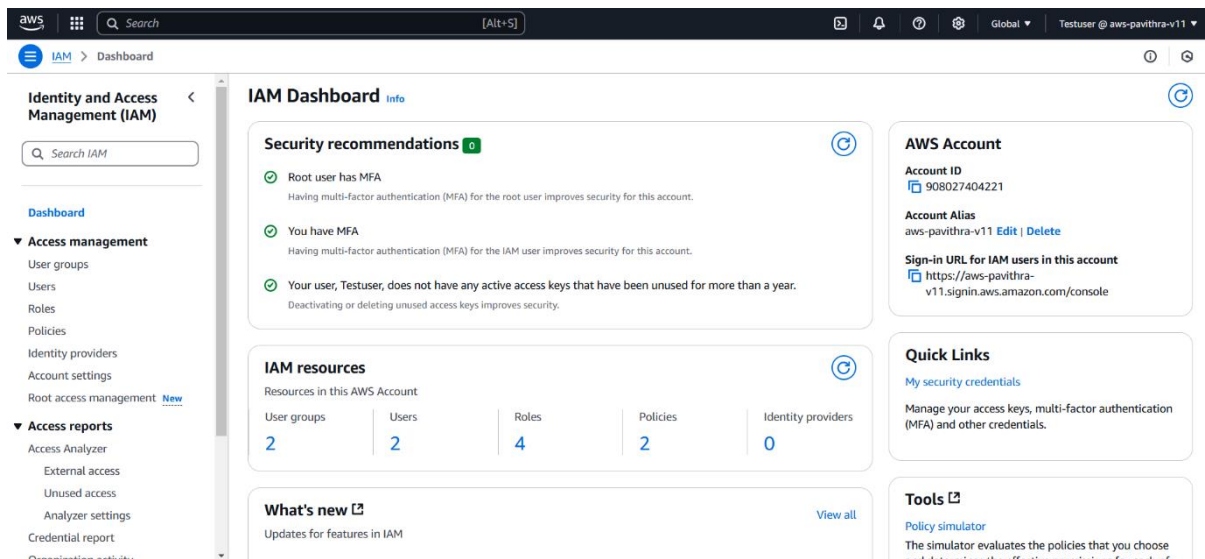
Remove Add permissions

Part 3: Secure Access with MFA

MFA adds an extra layer of security to AWS accounts.

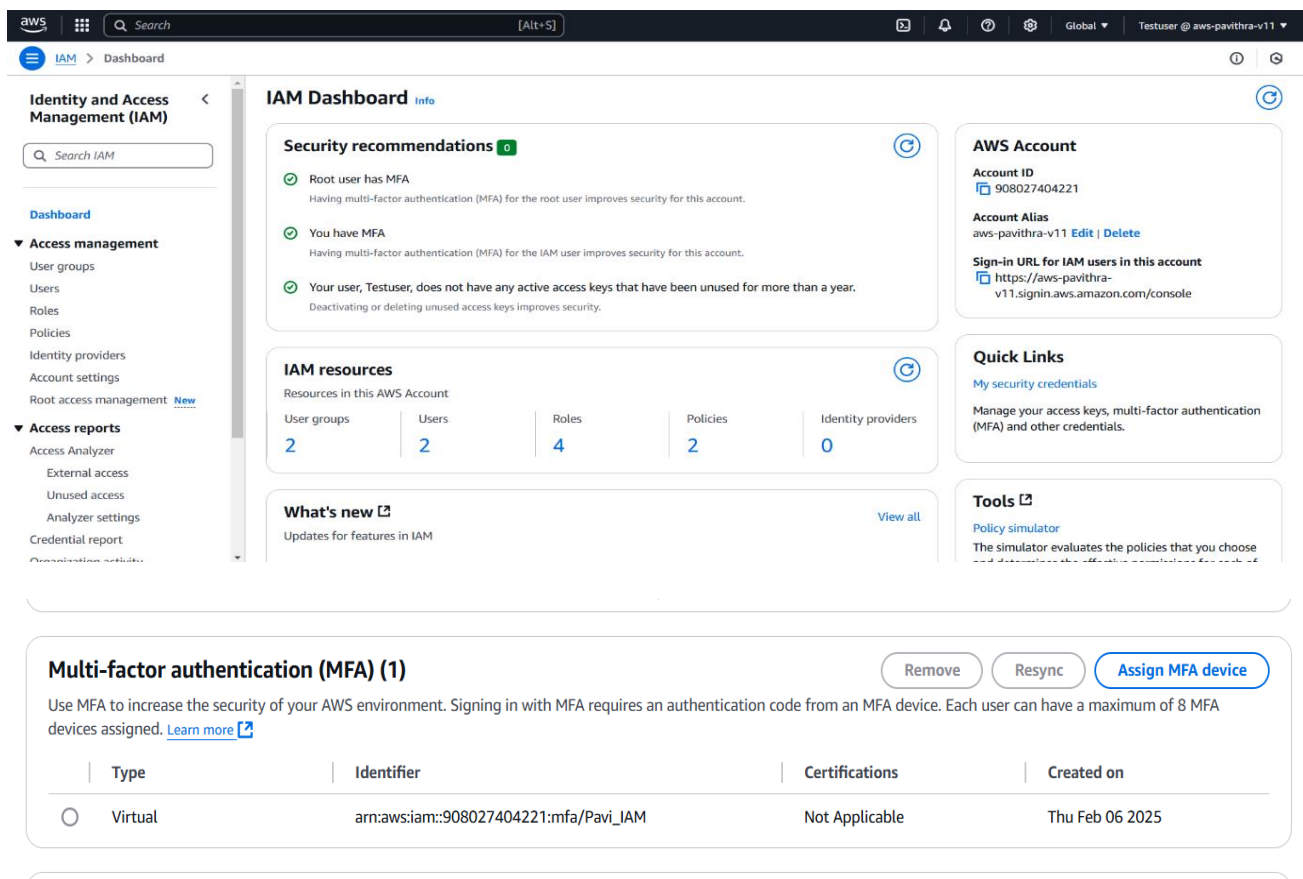
Step 3.1: Enable MFA for Root User

1. Log in as the root user and navigate to **My Security Credentials**.
2. Under **Multi-Factor Authentication (MFA)**, click **Activate MFA**.
3. Choose a device type (e.g., virtual MFA using Google Authenticator).
4. Scan the QR code, input the codes generated, and save.



Step 3.2: Enable MFA for IAM Users

1. Go to the **IAM Dashboard** and select **Users**.
2. Choose a user and click **Security credentials**.
3. Under **MFA**, click **Manage** and follow the steps to activate MFA.



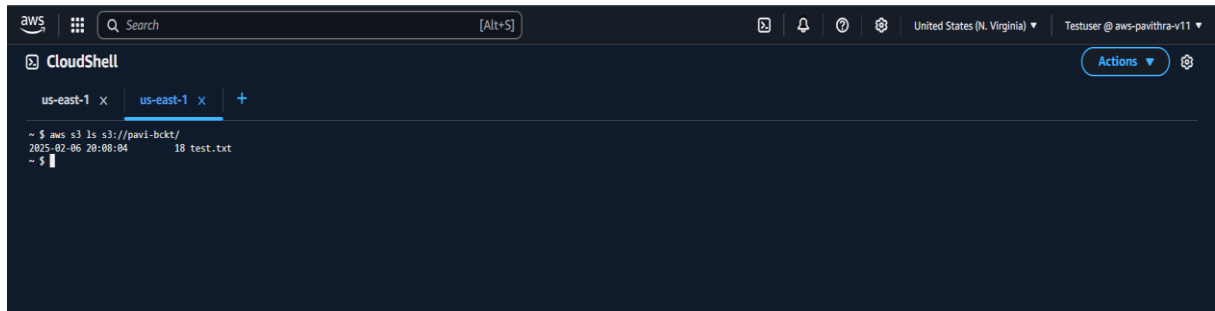
Part 4: Test Access

1. Log in as the developer to the AWS Management Console or AWS CLI.
2. Verify the roles and policies applied:

Run CLI commands such as:

```
aws s3 ls s3://example-bucket
```

- Ensure access matches the permissions defined in the policies.



2. Test MFA by logging in with the generated codes.

