

## 1. Create a New IAM User:

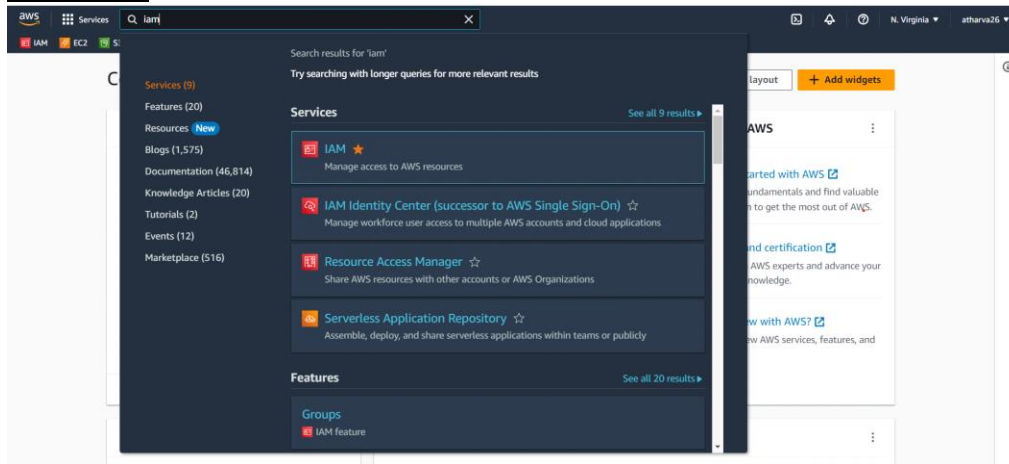
Create a new IAM user with programmatic access.

Assign appropriate permissions to the user based on their role or responsibilities.

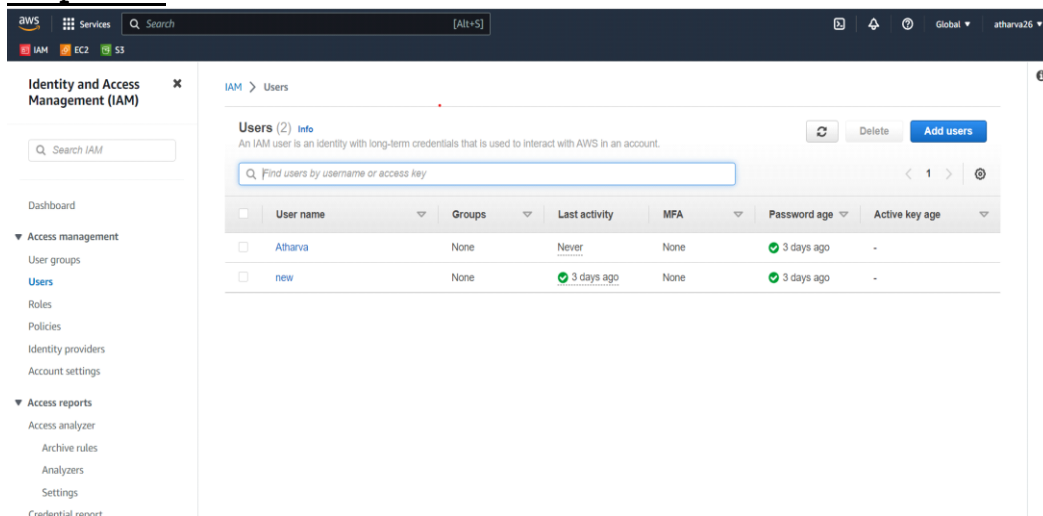
Generate and securely provide the user's access key and secret access key.

-->> creating i am user:-

**step 1:-**search and open iam service on aws



**step 2:-** Next click on users and next click on add user



**step 3 :-**Then gave user name then click on provides users access to the aws management console and in it select i want to create an iam user Gave a password to user by click on custom password  
password ex.:- Atharva@26+

aws

Services

Search

iam

ec2

s3

[Alt+S]

Global

atharva26

Set permissions

Step 3

**Review and create**

Step 4

Retrieve password

**User details**

User name atharva2002	Console password type Custom password	Require password reset No
--------------------------	------------------------------------------	------------------------------

**Permissions summary**

< 1 >

Name	Type	Used as
AdministratorAccess	AWS managed – job function	Permissions policy

**Tags - optional**

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

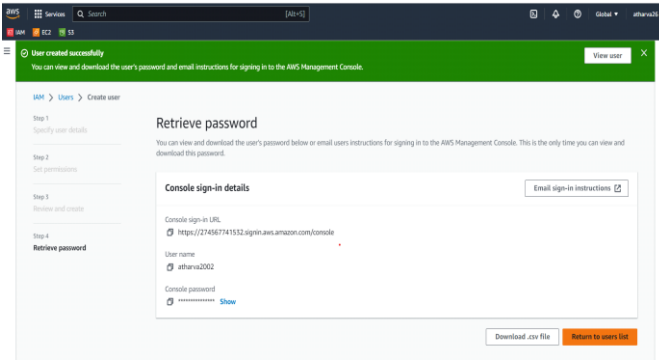
You can add up to 50 more tags.

Cancel

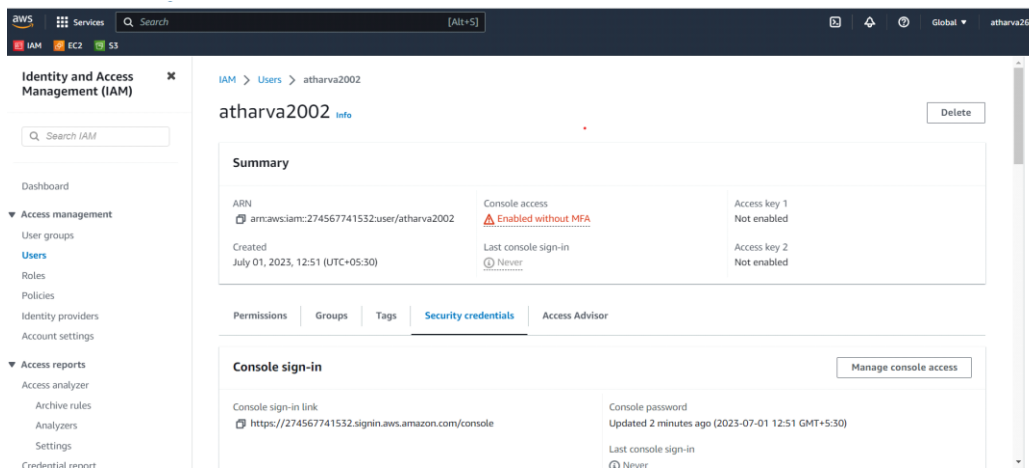
Previous

Create user

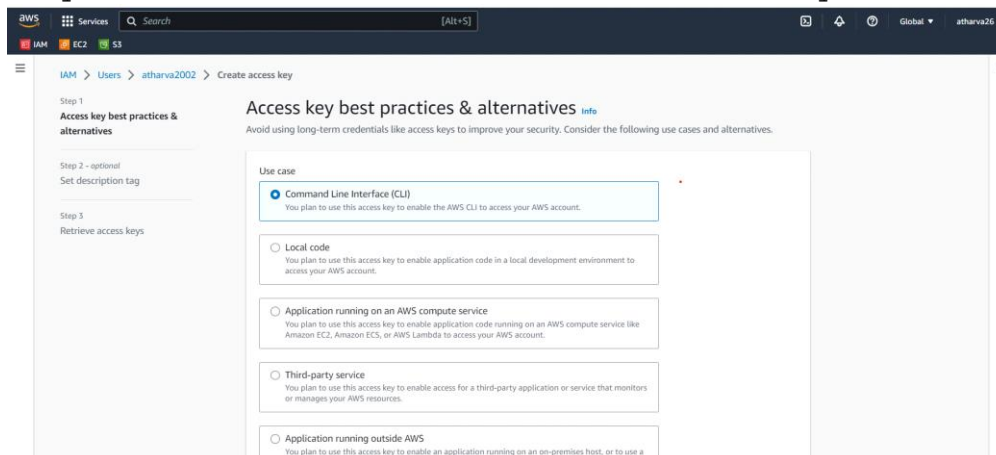
**step 5:-** Then check details you entered and if you want to add tags then you can add it. and click to create user.

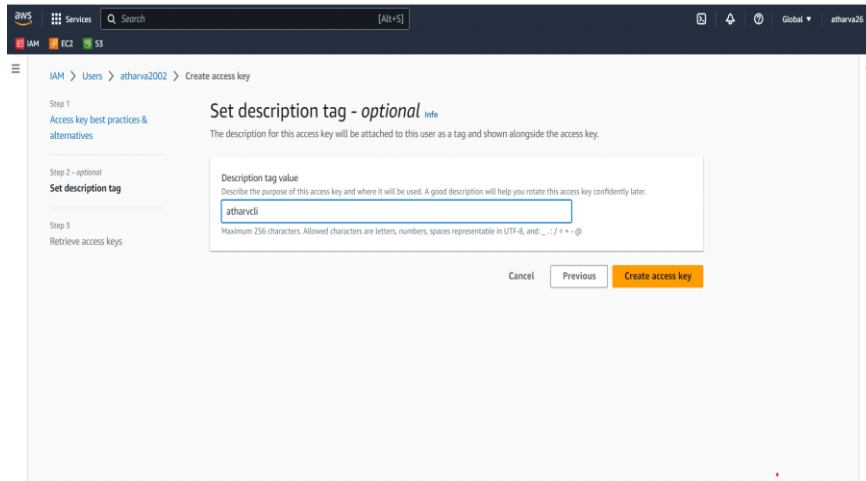


**step 6:-** Then go to user and go to security credentials go to access key click on create access key.

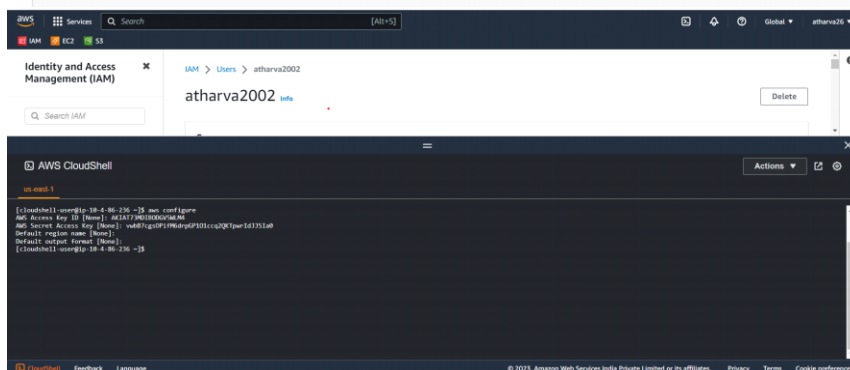
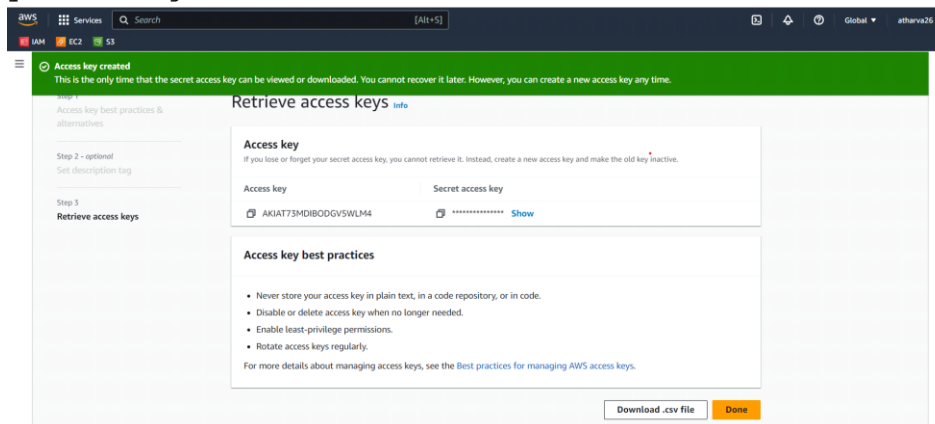


**step 7:-** Then choose option command line interface add description tag if you want and then click on create access key.





**step 8 :-** here please download that both keys because it will not be retained after that open cli cloudshell and type command aws configure after that it will ask for both keys just enter both keys and then enter enter and you will get access of that user on cli.



## 2. Configure IAM Roles:

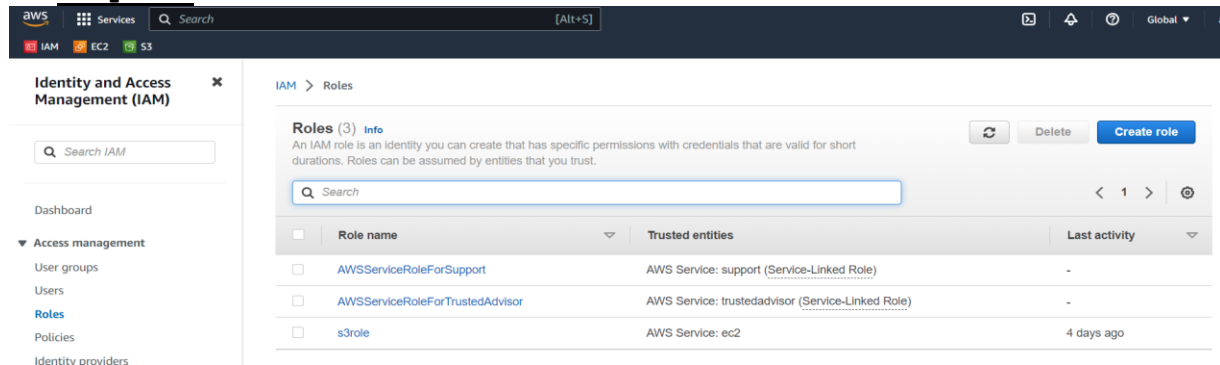
Create an IAM role for EC2 instances or Lambda functions with specific

permissions.

Attach policies to the role that grant necessary permissions to access AWS resources.

Assign the role to EC2 instances or Lambda functions.

➔ **Step 1:-**Go to IAM service click on role and then click on create role



**Step 2:-**Select aws service and select which service do you want ec2 or lambda and then click on next

**Trusted entity type**

☒ **AWS service**  
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

☐ **AWS account**  
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

☐ **Web identity**  
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

☐ **SAML 2.0 federation**  
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ **Custom trust policy**  
Create a custom trust policy to enable others to perform actions in this account.

**Use case**  
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

**Common use cases**

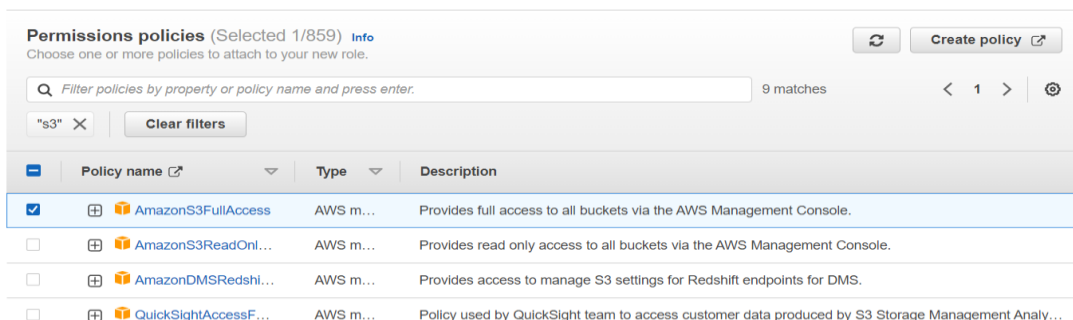
☒ **EC2**  
Allows EC2 instances to call AWS services on your behalf.

☐ **Lambda**  
Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

**Step 3:-**Then attach policy to that role. And click on next.

Add permissions [Info](#)



**Step 4:-**Then give name to the role and add tag if you want then click on create role.

## Name, review, and create

### Role details

#### Role name

Enter a meaningful name to identify this role.

atharvarole

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 alphanumeric and '+', '@', '-' characters.

### Step 2: Add permissions

Edit

#### Permissions policy summary

Policy name	Type	Attached as
AmazonS3FullAccess	AWS managed	Permissions policy

### Tags

#### Add tags - optional

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 tag

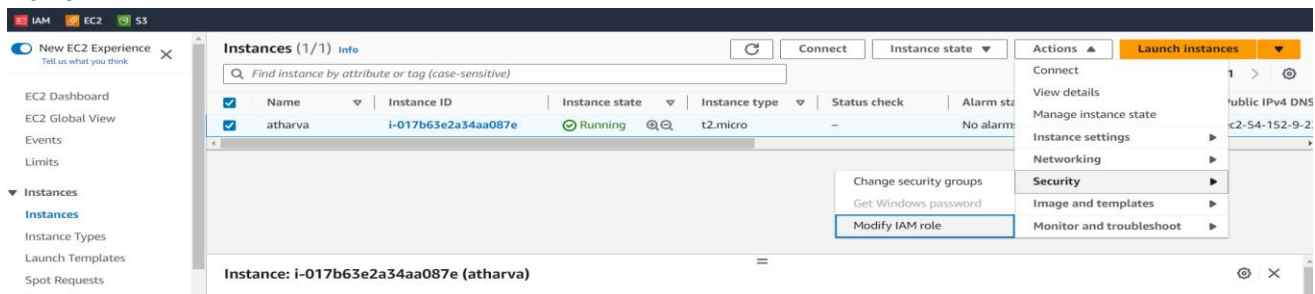
You can add up to 50 more tags.

Cancel

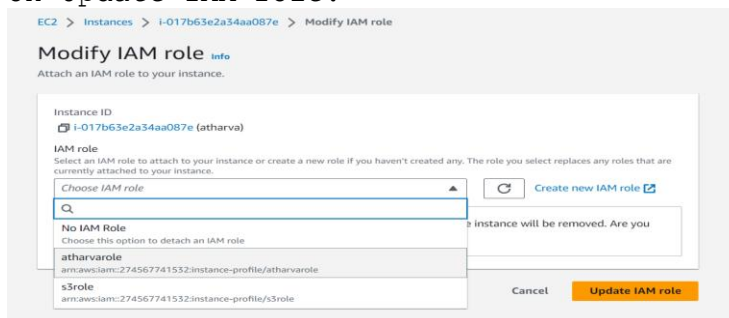
Previous

Create role

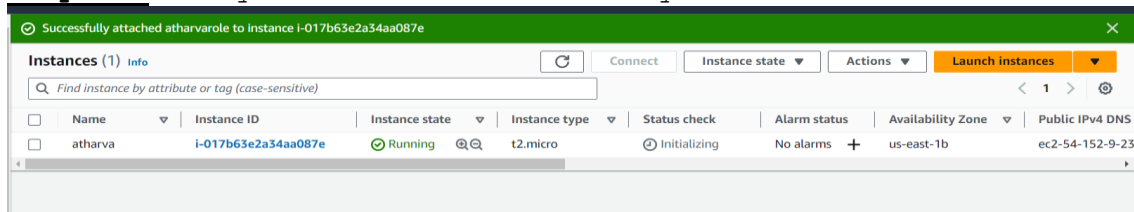
**Step 5:-** After that go to ec2 instance on which you want to apply role select that instance click on actions then security then select option Modify IAM role



**Step 6:-** Then select IAM role that we create in previous step and then click on Update IAM role.



**Step 7:-** now your role is successfully attached to the instance.



### 3. Implement Multi-Factor Authentication (MFA):

Enable MFA for IAM users to provide an additional layer of security.

Guide users on how to set up MFA devices (such as virtual MFA apps or hardware tokens).

Test the MFA configuration to ensure it functions correctly.

→ **Step 1:-** select user from IAM service on which you want apply MFA click on that user.

IAM > Users

Users (3) Info							
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.							
Find users by username or access key							
	User name	Groups	Last activity	MFA	Password age	Active key age	
<input type="checkbox"/>	Atharva	None	◁	None	✓ 4 days ago	-	
<input type="checkbox"/>	atharva2002	None	◁	None	✓ Yesterday	✓ Yesterday	

**Step 2:-** click on enable MFA

IAM > Users > atharva2002

atharva2002 Info Delete

Summary

ARN

arn:aws:iam::274567741532:user/atharva2002

Created

July 01, 2023, 12:51 (UTC+05:30)

Console access

Enabled without MFA

Access key 1

AKIAT73MDIBODGV5WLM4 - Active

Never used. Yesterday old.

Access key 2

Not enabled

Permissions Groups Tags (1) Security credentials Access Advisor

Enable MFA

**Step 3:-** then enter one meaningful name and select option like authenticator app or security key or hardware totp token and click on next I choose app

**Select MFA device**

**Specify MFA device name**

Device name  
Enter a meaningful name to identify this device.

atharva

Maximum 128 characters. Use alphanumeric and '+', '.', '@', '=', '\_' characters.

**Select MFA device** [Info](#)

Select an MFA device to use, in addition to your username and password, whenever you need to authenticate.

☒ **Authenticator app**  
Authenticate using a code generated by an app installed on your mobile device or computer.

☐ **Security Key**


**Step 4:-** then open app and scan QR or enter that secret key which is shown in below picture and after that enter 2 MFA codes then click on next. MFA is enabled.

**Set up device**

**Set up your authenticator app**  
A virtual MFA device is an application running on your device that you can configure by scanning a QR code.

1 Install a compatible application such as Google Authenticator, Duo Mobile, or Authy app on your mobile device or computer.  
[See a list of compatible applications](#)

2 Scan the QR code or enter the secret key.

QR code: 

Secret key: `MXBULR8GWC6ZLV4I6AM  
XC663JGL4FX4OWBY663  
7HCU5P5CPKJ2PFL4SU54  
7LHM`

app, choose  
the, then use  
Alternatively,  
Show secret key

**MFA device assigned**  
You can register up to 8 MFA devices of any combination of the currently supported MFA types with your AWS account root and IAM user. With multiple MFA devices, you only need one MFA device to sign in to the AWS console or create a session through the CLI with that user.

**Summary**

ARN	Console access	Access key 1
arn:aws:iam::274567741532:user/atharva2002	Enabled with MFA	AKIA73MD800GVSWM4 - Active (Never used, yesterday old)
Created July 01, 2025, 12:51 (UTC+05:30)	Last console sign-in (Never)	Access key 2 Not enabled

#### 4. Create IAM Policies:

Write a custom IAM policy that allows or denies specific actions on AWS resources.

Associate the policy with the appropriate IAM users, groups, or roles. Test the policy to verify that the desired access control is enforced.

→ **Step 1:-** Click on policy and create policy in IAM service

**Identity and Access Management (IAM)**

**Policies (1104)** [Info](#)

A policy is an object in AWS that defines permissions.

Filter policies by property or policy name and press enter.

Policy name	Type	Used as	Description
AdministratorAccess	AWS managed - job function	Permissions policy (2)	Provides full access
PowerUserAccess	AWS managed - job function	None	Provides full access
ReadOnlyAccess	AWS managed - job function	None	Provides read-only
AWSCloudFormationReadOnlyAccess	AWS managed	None	Provides access to
CloudFrontFullAccess	AWS managed	None	Provides full access
AWSCloudHSMFullAccess	AWS managed	None	Provides full access

**Step 2:-** Select which permission you want to add or deny.



IAM > Policies > Create policy

Step 1  
Specify permissions

Step 2  
Review and create

Policy editor

Visual JSON Actions

▼ Select a service  
Specify what actions can be performed on specific resources in a service.

Search

Popular services

Auto Scaling CloudFront EC2 IAM  
Lambda RDS S3 SNS

+ Add more permissions

Cancel Next

**Step 3:-** I chose ec2 policy and I want to deny it for all ec2 service that's why I clicked on all service and then click on switch to deny.

▼ EC2  
Allow 1 Actions

Specify what actions can be performed on specific resources in EC2.

▼ Actions allowed  
Specify actions from the service to be allowed.

Filter Actions

Switch to deny permissions

Manual actions | Add actions

☒ All EC2 actions (ec2:\*)

Access level

Expand all | Collapse all

► List (Selected 168/168)  
► Read (Selected 31/31)  
► Write (Selected 403/403)  
► Permissions management (Selected 5/5)  
► Tagging (Selected 2/2)

**Step 4:-** after click to switch to deny scroll down and there you can select it for all or for specific resources I want apply it on all resources so I select all and then click on next.

Policy editor

▼ EC2  
deny 1 Actions

▼ Resources  
Specify resource ARNs for these actions.

☐ Specific ☒ All

► Request conditions - optional  
Actions on resources are allowed or denied only when these conditions are met.

+ Add more permissions

Security: 0 Errors: 0 Warnings: 0 Suggestions: 0

Cancel Next

**Step 5:-** give the name to that policy and add tag if you want to add and then click on create policy.

### Policy details

#### Policy name

Enter a meaningful name to identify this policy.

atharvapolity

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

#### Description - optional

Add a short explanation for this policy.

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

### 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 tag

You can add up to 50 more tags.

[Cancel](#)

[Previous](#)

[Create policy](#)

**Step 6:-** policy is created and created policy is shown in policy section as mentioned in the below pictures.

[IAM](#) > [Policies](#)

The screenshot shows the AWS IAM console's Policies page. At the top, it says 'Policies (1105)' with an 'Info' link. Below this is a search bar and a table of policies. The table has columns for Policy name, Type, Used as, and Description. Two policies are visible: 'atharvapolicy' (Customer managed) and 'AdministratorAccess' (AWS managed - job function).

Policy name	Type	Used as	Description
<a href="#">atharvapolicy</a>	Customer managed	None	
<a href="#">AdministratorAccess</a>	AWS managed - job function	Permissions policy (2)	Provides full access to AWS services a...

**Step 7:-** then go to user on which you want to apply the policy. Then click on add permission.

[IAM](#) > [Users](#) > [atharva2002](#)

[atharva2002](#) [Info](#)

[Delete](#)

The screenshot shows the user 'atharva2002' in the IAM console. It displays a 'Summary' section with details like ARN, console access, and access keys. Below this is a 'Permissions' section with a table of attached policies. The 'Add permissions' button is visible in the top right corner of the permissions section.

**Summary**

ARN <a href="#">arn:aws:iam::274567741532:user/atharva2002</a>	Console access Enabled with MFA	Access key 1 AKIAT73MDIBODGV5WLM4 - Active <a href="#">Never used, Yesterday old.</a>
Created July 01, 2023, 12:51 (UTC+05:30)	Last console sign-in <a href="#">Never</a>	Access key 2 Not enabled

**Permissions policies (1)**

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

Filter by Type: All types

[Add permissions](#) [Create inline policy](#) [Add permissions](#)

**Step 8:-** then click on attached policy directly then search your created policy and then click on next.

The screenshot shows the 'Add permissions' wizard in the AWS IAM console. It has two steps: 'Add permissions' and 'Review'. In the 'Add permissions' step, there are three options: 'Add user to group', 'Copy permissions', and 'Attach policies directly'. The 'Attach policies directly' option is selected. Below this is a table of 'Permissions policies (1/1106)' with columns for Policy name, Type, and Attached entities. The 'atharvapolicy' is listed under 'Customer managed' policies.

**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 (1/1106)**

Filter by Type: Customer managed 1 match

Policy name	Type	Attached entities
<a href="#">atharvapolicy</a>	Customer managed	0

[Cancel](#) [Next](#)

**Step 9:-** check details and click on add permission.

## Review

The following policies will be attached to this user. [Learn more](#)

**User details**

User name  
atharva2002

**Permissions summary (1)** < 1 >

Name	Type	Used as
atharvapolicy	Customer managed	Permissions policy

Cancel Previous Add permissions

**Step 10 :-** check that applied policy works or not so logged into the user atharva2002 and check that this user will get EC2 access or not.

Resources

You are using the following Amazon EC2 resources in the US East (Ohio) Region:

Instances (running)	0	Auto Scaling Groups	0	Dedicated Hosts	API Error
Elastic IPs	API Error	Instances	API Error	Key pairs	API Error
Load balancers	0	Placement groups	API Error	Security groups	API Error
Snapshots	API Error	Volumes	API Error		

Launch instance

To get started, launch an Amazon EC2 Instance, which is a virtual server in the cloud.

Launch instance

Migrate a server

Service health

AWS Health Dashboard

Region: US East (Ohio)

Status

Account attributes

Supported platforms

An error occurred: An error occurred retrieving supported platforms

An error occurred: An error occurred checking for a default VPC

Settings

EBS encryption

Zones

EC2 Serial Console

Default credit specification

Console experiments

Explore AWS

## 5. Use IAM Groups:

Create an IAM group and assign permissions to it.

Add IAM users to the group to manage their access collectively.

Remove users from the group when their access requirements change.

→ **Step 1:-** Go to IAM service click on user group and the click on crate group.

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

IAM > User groups

User groups (0) Info

A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

Filter User groups by property or group name and press enter

Create group

Group name	Users	Permissions	Creation time
No resources to display			

**Step 2:-** Then give the name to the group and select users you want to add in that group. Then add the policy to the group. Then click on the create group

Create user group

**Name the group**

User group name  
Enter a meaningful name to identify this group.

atharvagroup

Maximum 128 characters. Use alphanumeric and "+", "@", "." characters.

**Add users to the group - Optional** (Selected 3/3) [Info](#)

An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS. A user can belong to up to 10 groups.

Q Search

<input checked="" type="checkbox"/>	User name	Groups	Last activity	Creation time
<input checked="" type="checkbox"/>	Atharva	0	None	5 days ago
<input checked="" type="checkbox"/>	atharva2002	0	None	Yesterday
<input checked="" type="checkbox"/>	new	0	4 days ago	4 days ago

**Attach permissions policies - Optional** (Selected 1/860) [Info](#)

You can attach up to 10 policies to this user group. All the users in this group will have permissions that are defined in the selected policies.

Q Filter policies by property or policy name and press enter.

<input checked="" type="checkbox"/>	Policy name	Type	Description
<input checked="" type="checkbox"/>	atharvapolicy	Customer managed	
<input type="checkbox"/>	AdministratorAccess	AWS managed - job function	Provides full access to A
<input type="checkbox"/>	PowerUserAccess	AWS managed - job function	Provides full access to A
<input type="checkbox"/>	ReadOnlyAccess	AWS managed - job function	Provides read-only acces
<input type="checkbox"/>	AWSCloudFormationReadOnlyAccess	AWS managed	Provides access to AWS
<input type="checkbox"/>	CloudFrontFullAccess	AWS managed	Provides full access to th
<input type="checkbox"/>	AWSDirectConnectReadOnlyAccess	AWS managed	Provides read only acces
<input type="checkbox"/>	AmazonAppStreamFullAccess	AWS managed	Provides full access to A

Cancel Create group

**Step 3:-** Group is created.

✓ atharvagroup user group created. [View group](#) ✕

[IAM](#) > User groups

**User groups (1)** [Info](#)

A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

Q Filter User groups by property or group name and press enter

<input type="checkbox"/>	Group name	Users	Permissions	Creation time
<input type="checkbox"/>	atharvagroup	3	✓ Defined	Now

**Step 4:-** to remove user from group then click on group select user you want to remove and click on remove.

atharvagroup Delete

**Summary** Edit

User group name atharvagroup	Creation time July 02, 2023, 19:54 (UTC+05:30)	ARN <a href="#">arn:aws:iam::274567741532:group/atharvagroup</a>
---------------------------------	---------------------------------------------------	---------------------------------------------------------------------

**Users** | Permissions | Access Advisor

**Users in this group** (Selected 1/3)  
An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.

Search < 1 > ⓘ

<input type="checkbox"/>	User name ↗	Groups	Last activity	Creation time
<input checked="" type="checkbox"/>	Atharva	1	None	5 days ago
<input type="checkbox"/>	new	1	4 days ago	4 days ago
<input type="checkbox"/>	atharva2002	1	None	Yesterday

## 6. Implement IAM Access Analyzer:

**Enable IAM Access Analyzer to identify unintended access to your AWS resources.**

**Review and resolve findings generated by IAM Access Analyzer.**

**Continuously monitor and remediate any potential security risks.**

**Enable AWS CloudTrail for IAM:**

**Enable CloudTrail to capture API activity related to IAM.**

**Configure CloudTrail to store logs in an S3 bucket.**

**Review and analyze CloudTrail logs for any suspicious or unauthorized activities.**

**Implement IAM Password Policies:**

Define and enforce password policies for IAM users.

Set requirements such as minimum password length, complexity, and expiration.

Regularly remind users to update their passwords according to the policy.

→ **Step 1:-** click on Account settings and click on edit option in password policy.

**Identity and Access Management (IAM)**

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles
- Policies
- Identity providers
- Account settings**

Access reports

- Access analyzer
- Archive rules
- Analysers

**Account settings** Info

**Password policy** Info Edit

Configure the password requirements for the IAM users.

This AWS account uses the following default password policy:

**Password minimum length**  
8 characters

**Password strength**  
Include a minimum of three of the following mix of character types:

- Uppercase
- Lowercase
- Numbers
- Non-alphanumeric characters

**Other requirements**

- Never expire password
- Must not be identical to your AWS account name or email address

**Step 2:-** and then click on custom and do the changes whatever you want

and click on save change.

### Password policy

☐ IAM default  
Apply default password requirements.

☒ Custom  
Apply customized password requirements.

**Password minimum length.**  
Enforce a minimum length of characters.

characters

Needs to be between 6 and 128.

**Password strength**

☐ Require at least one uppercase letter from the Latin alphabet (A-Z)

☐ Require at least one lowercase letter from the Latin alphabet (a-z)

☐ Require at least one number

☐ Require at least one non-alphanumeric character (!@#\$%^&\*()\_+-=[ ]{|'})

**Other requirements**

☐ Turn on password expiration

☐ Password expiration requires administrator reset

☐ Allow users to change their own password

☐ Prevent password reuse

Cancel

Save changes