

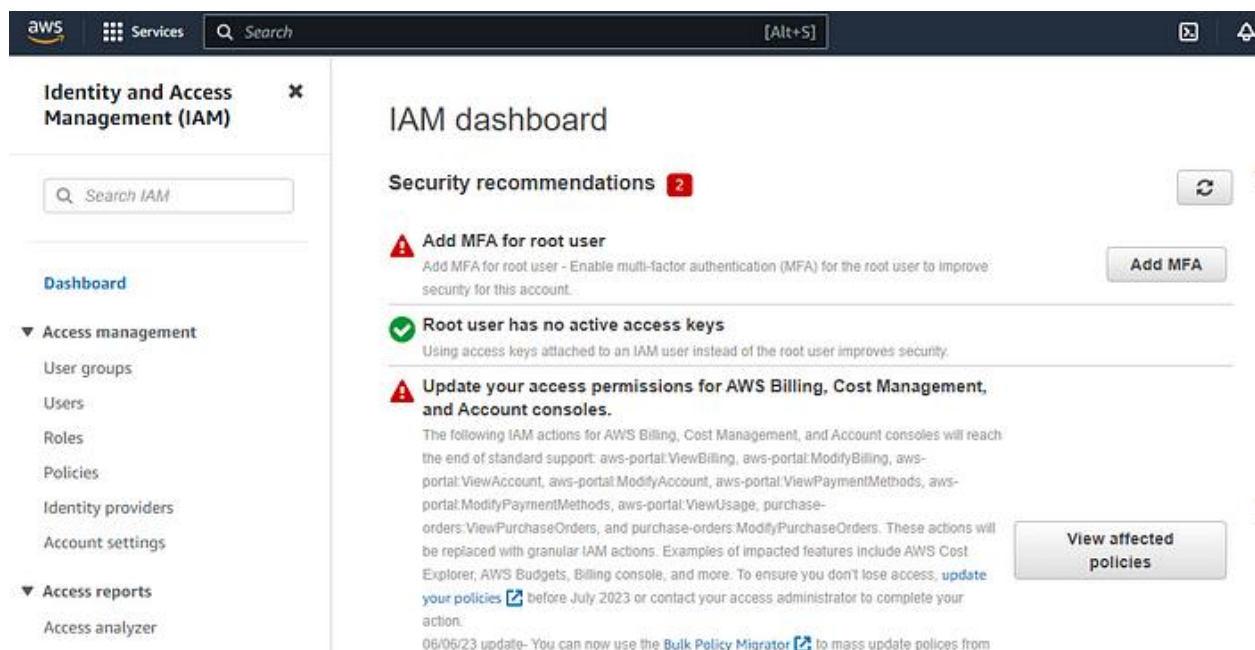
How to create an IAM User and add MFA

Step 1: Sign in to the AWS Management Console

Go to the AWS Management Console

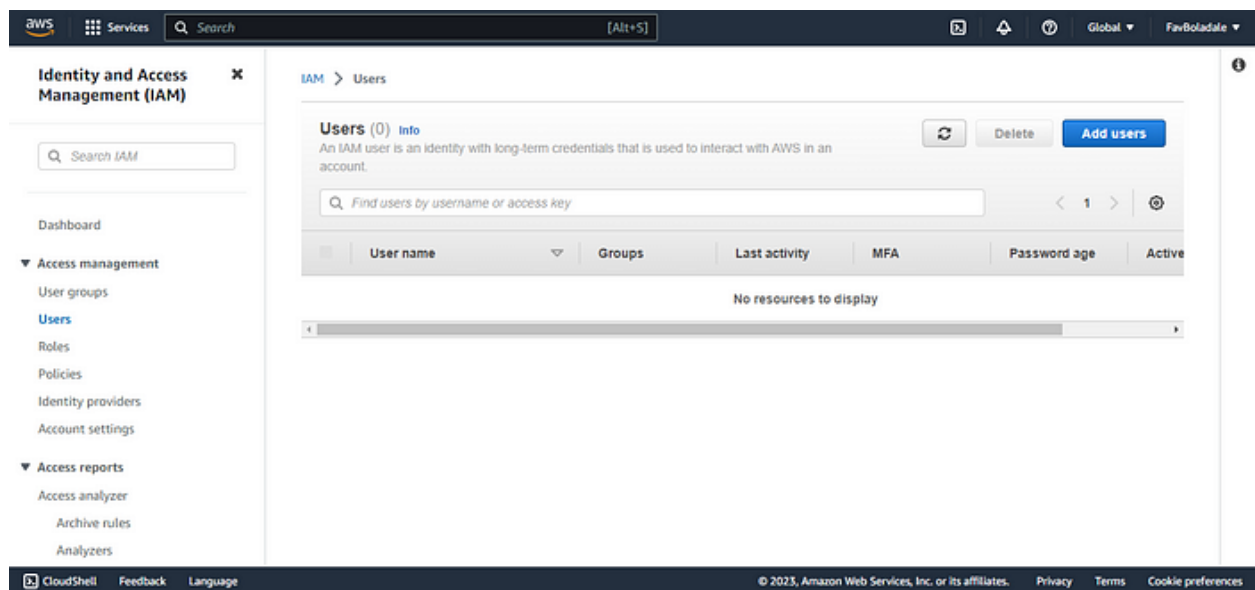
(<https://console.aws.amazon.com>) and sign in using your AWS

account credentials. Access the IAM service: Once logged in, search for “IAM” in the AWS Management Console search bar and select the IAM service from the results.



Step 2: Navigate to “Users” in the IAM console

In the IAM console, on the left-hand side, click on “Users” under the “Access management” section. This will display a list of existing IAM users in your account. In my case I have none, so I have to click the add users on the top right corner.



Step 3: Specify User Details

Here, we have to specify user details by supplying the necessary information as required such as user name, type of user, password (autogenerated or custom), etc. You can also choose to enable programmatic access (which generates access keys for API

access) and AWS Management Console access (which allows the user to log in to the AWS Management Console).

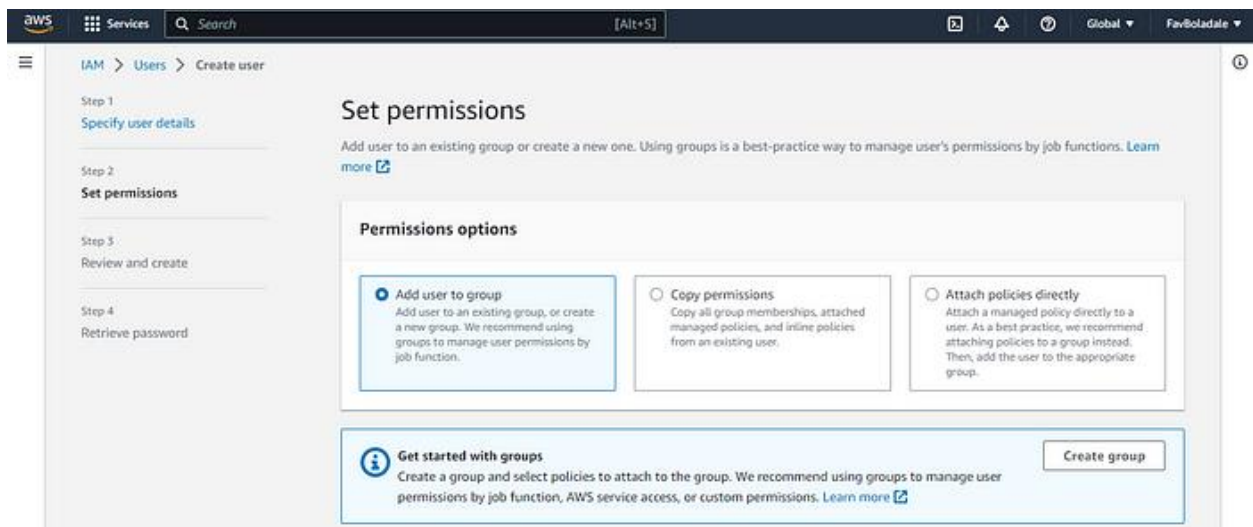
The screenshot shows the 'Specify user details' step in the AWS IAM console. The left sidebar lists four steps: Step 1 (Specify user details), Step 2 (Set permissions), Step 3 (Review and create), and Step 4 (Retrieve password). The main content area is titled 'Specify user details' and contains a 'User details' section. In this section, the 'User name' field is filled with 'favboladale'. Below the field, a note states: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and +, -, @, _ (hyphen)'. A checkbox labeled 'Provide user access to the AWS Management Console - optional' is checked. Below this, a note says: 'If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.' A blue information box titled 'Are you providing console access to a person?' contains two options: 'Specify a user in Identity Center - Recommended' (with a note about using Identity Center for console access) and 'I want to create an IAM user' (with a note about creating IAM users for programmatic access).

You can also enable the user to change the password upon sign in which is highly recommended.

The screenshot shows the 'Console password' step in the AWS IAM console. The left sidebar is the same as in the previous screenshot. The main content area is titled 'Console password' and contains two options: 'Autogenerated password' (with a note: 'You can view the password after you create the user.') and 'Custom password' (selected). Below the 'Custom password' option, there is a password input field with a masked password '*****'. Below the field, two requirements are listed: 'Must be at least 8 characters long' and 'Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & * () _ + - (hyphen) = [] { } | ' '. A checkbox labeled 'Show password' is unchecked. A checkbox labeled 'Users must create a new password at next sign-in - Recommended' is checked, with a note: 'Users automatically get the IAMUserChangePassword policy to allow them to change their own password.' A blue information box at the bottom states: 'If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more'. At the bottom right, there are 'Cancel' and 'Next' buttons. The footer of the console shows 'CloudShell', 'Feedback', 'Language', and copyright information for Amazon Web Services, Inc.

Step 4: Set user permissions

Now we have to set permissions: you can choose to add the user to an existing group or attach policies directly to the user. Groups are a convenient way to manage permissions for multiple users, as you can assign policies to a group, and any user added to that group inherits the group's permissions.



Step 5: Configure Tags, Review, and Create

Here, you can assign key-value pairs as tags to the user. Tags provide additional metadata and can be useful for organizing and managing your IAM users. I used Department- Engineering as below.

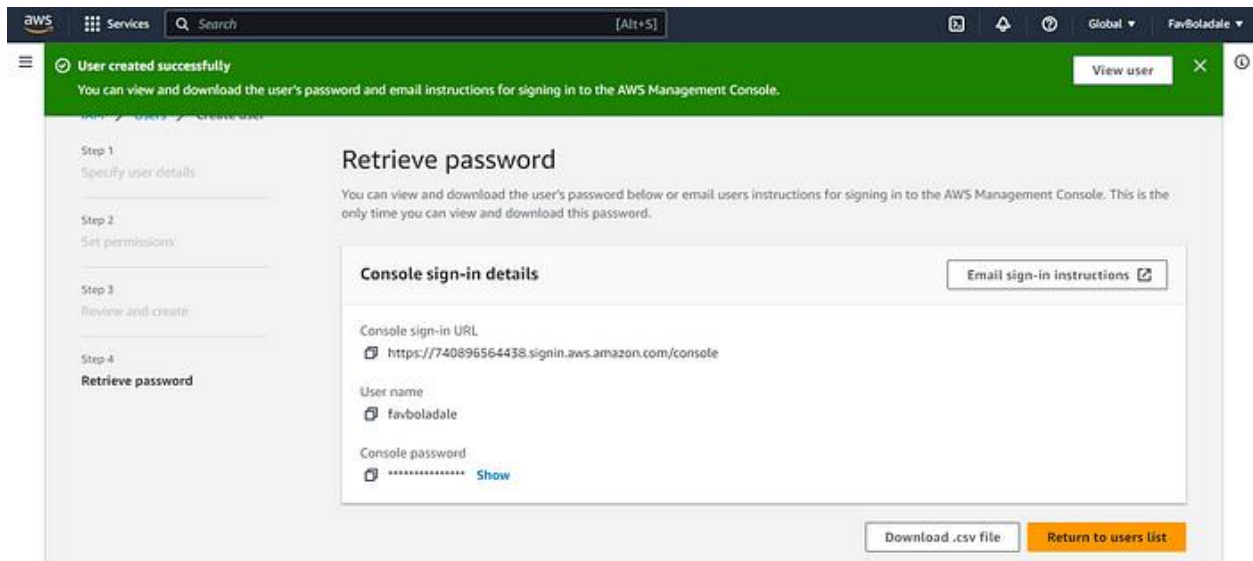
The screenshot shows the AWS IAM console interface during the 'Set permissions' step of creating a new user. The left sidebar indicates the current step is 'Review and create'. The main content area is divided into three sections: 'User details', 'Permissions summary', and 'Tags - optional'. The 'User details' section shows the user name 'favboladale', console password type 'Custom password', and 'Require password reset' set to 'Yes'. The 'Permissions summary' section shows a table with one entry: 'IAMUserChangePassword' (AWS managed) used as a 'Permissions policy'. The 'Tags - optional' section shows two tags: 'Department' and 'Engineering'. The bottom footer contains navigation links and copyright information.

Name	Type	Used as
IAMUserChangePassword	AWS managed	Permissions policy

Key	Value - optional
Department	Engineering

Step 6: Download the Access key and Secret Access Key

After setting up the user details, permissions, and tags, review the configuration. Make sure everything is accurate, and then click on the “Create user” button. Before you leave the page, kindly download the CSV file because it will no longer be available.



How to add a user to MFA

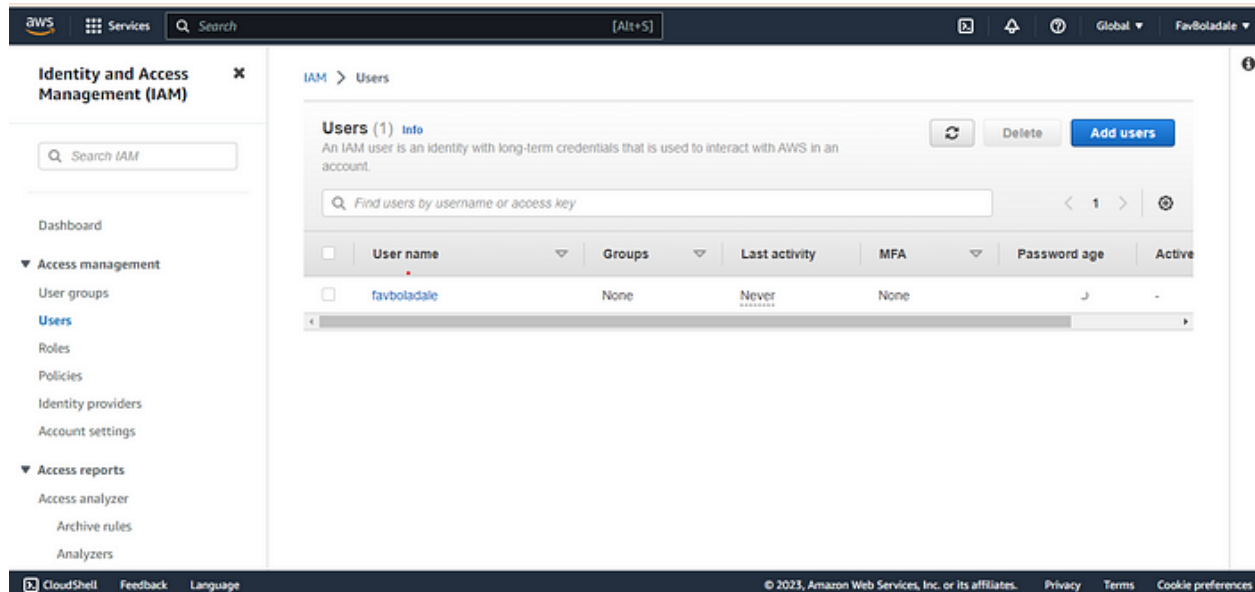
Prerequisites:

Download the Google Authenticator App on your mobile phone available on google play and IOS AppStore.

Step 1: Sign in to the AWS Management Console

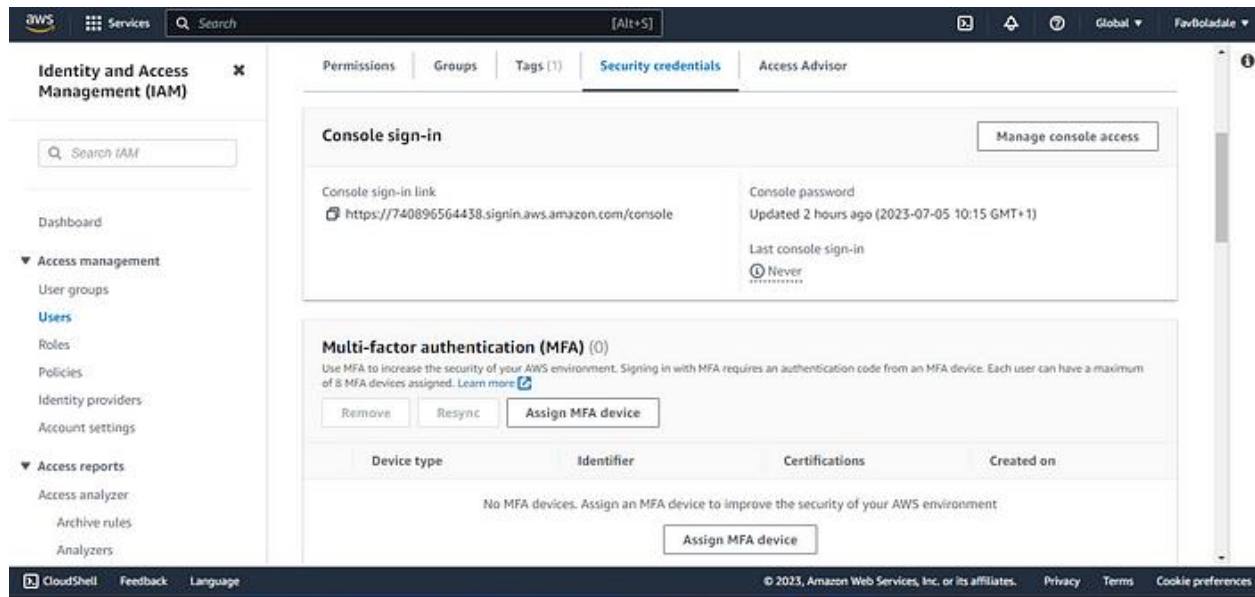
Go to the AWS Management Console and access the IAM Service. In the IAM console, on the left-hand side, click on “Users” under the “Access management” section. This will display a list of existing IAM

users in your account. Then click on the user you just created. For me, I gave the user the name “favboladale” as seen below.



Step 2: Enable MFA for the User

With the user selected, click on the “Security credentials” tab. Under the “Multi-factor authentication (MFA)” section, click on the “Assign MFA Device” button.



Step 3: Choose the “Virtual MFA device” option

Select the “Virtual MFA device” option to use the Google Authenticator app as the MFA device.

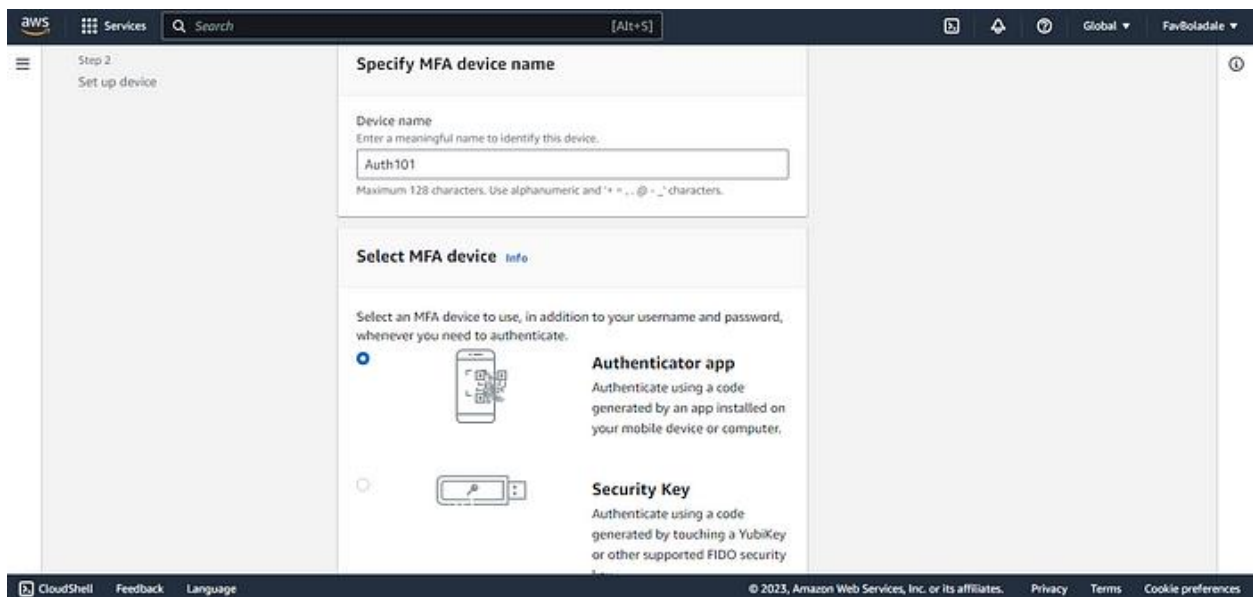
Once you click a new page will pop up, provide the name for your MFA, select Authenticator App, and click next.

Step 4: Set Up the Virtual MFA Device

In this step, you have two options to set up the virtual MFA device:

a. **Use QR Code:** Open the Google Authenticator app on your smartphone or tablet, tap the “+” symbol to add a new account, and select the “Scan a barcode” option. Scan the QR code displayed on the AWS console.

b. **Use Secret Key:** If you are unable to scan the QR code, you can manually enter the secret key displayed on the AWS console into the Google Authenticator app.

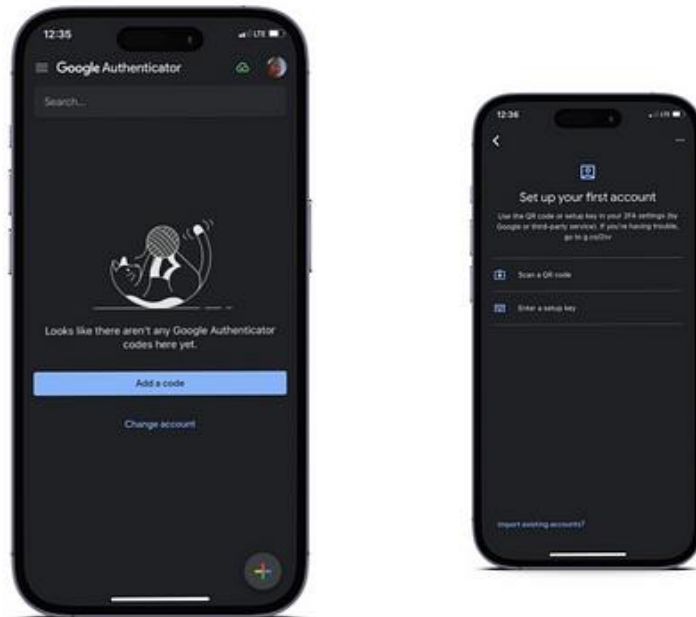


Then this page will come up



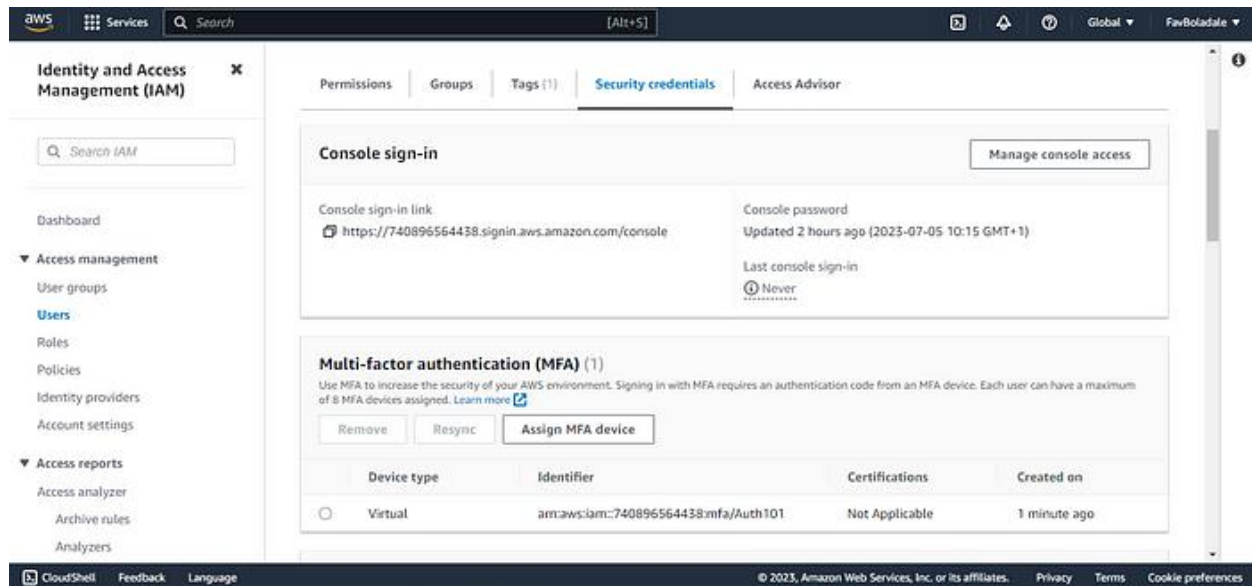
Step 5: Enter the Generated MFA Codes

The Google Authenticator app will generate a 6-digit verification code. Enter the first verification code into the “Authentication Code 1” field on the AWS console, and then enter the second verification code into the “Authentication Code 2” field.



Step 6: Verify and Complete MFA Setup

After entering the verification codes, click on the “Assign MFA” button to complete the MFA setup for the IAM user.



Step 7: Test MFA Setup

To ensure that MFA is working correctly, sign out of the AWS Management Console and sign back in. When prompted, enter your username and password, and then provide the current 6-digit verification code from the Google Authenticator app.

Conclusion

Enabling MFA for IAM users adds an extra layer of security to your AWS environment by requiring users to provide an additional authentication factor. By following the step-by-step instructions

outlined in this guide, you can easily add MFA to an IAM user in AWS using the Google Authenticator app. Protecting your AWS resources with MFA helps safeguard against unauthorized access and enhances the overall security of your cloud infrastructure.

How to Create IAM Roles in AWS

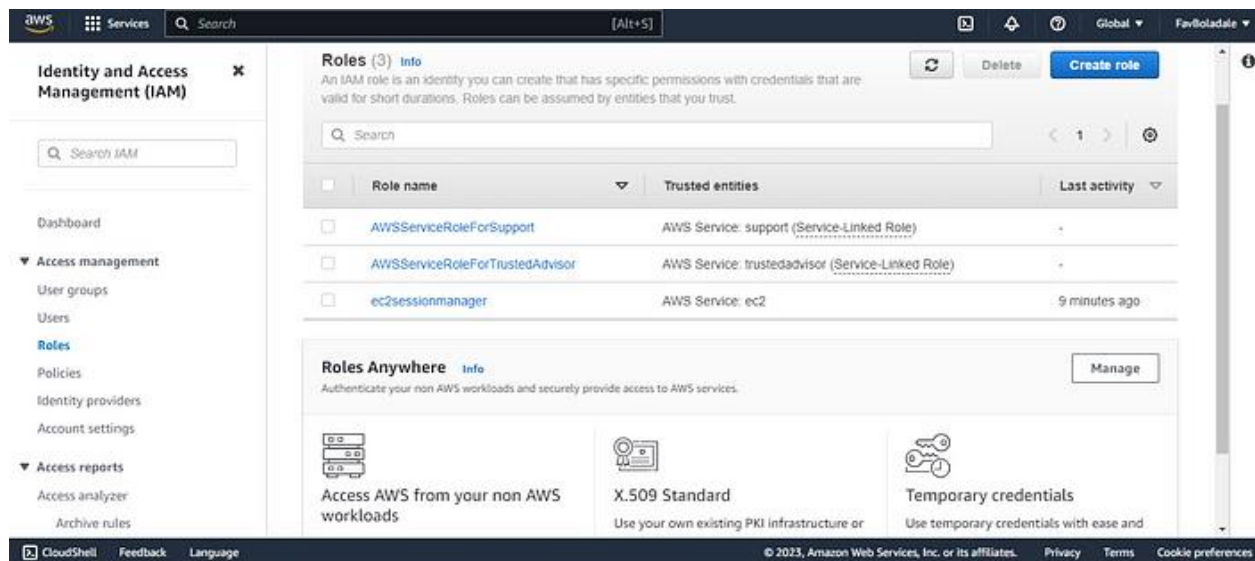
Step 1: Navigate to the IAM Service on AWS

Management Console

Once logged in, search for “IAM” in the services search bar at the top of the AWS Management Console. Click on “IAM” from the suggestions or select it from the list of available services.

Step 2: Access the IAM Roles Section

In the IAM console, you will find the left-hand navigation pane. Click on “Roles” to proceed, then click on the “Create role” button.



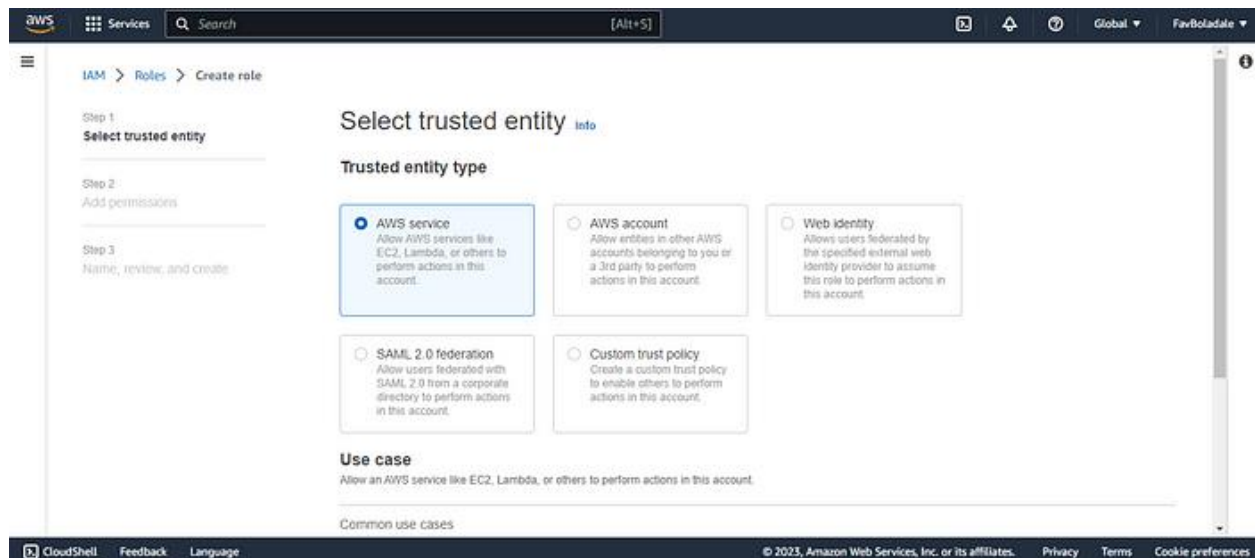
Step 3: Select the Trusted Entity and the Use Case

On the “Select type of trusted entity” page, you have various options:

- a. AWS service: If you want to grant permissions to an AWS service.
- b. Another AWS account: If you want to grant access to another AWS account.
- c. Web identity: If you want to grant access to web identity providers like Amazon Cognito, Google, or Facebook.

d. SAML 2.0 federation: If you want to grant access to users from your corporate directory that supports SAML 2.0.

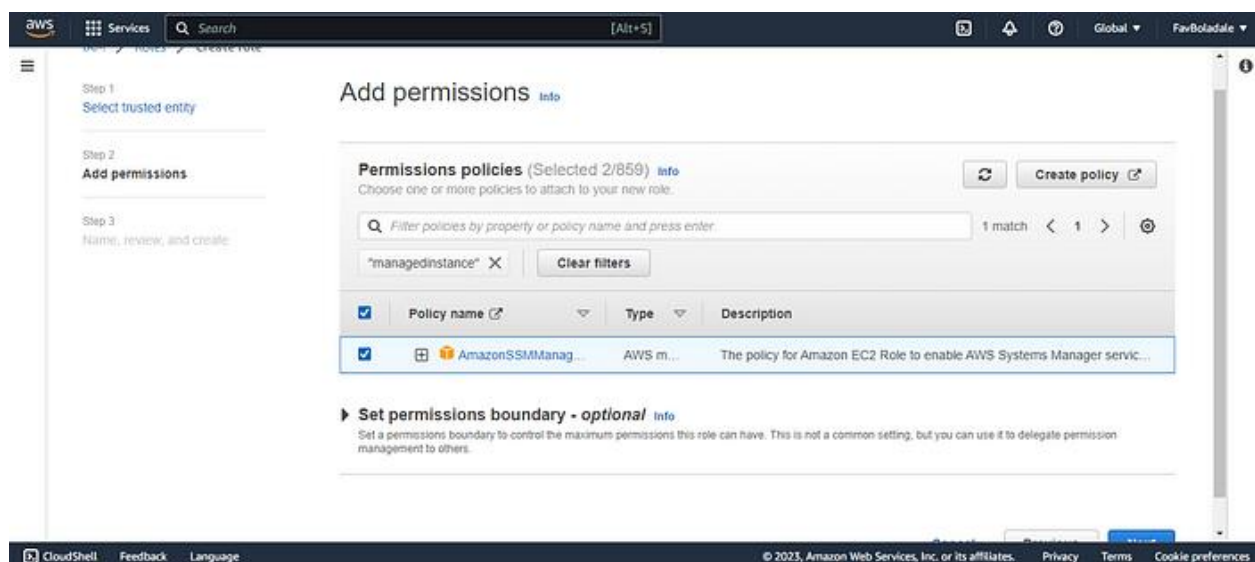
In the Use Case section, choose the appropriate option based on your requirements and click on the “Next: Permissions” button.



Step 4: Attach Permissions Policies

On the “Attach permissions policies” page, you can attach one or more IAM policies to the role. You can choose from existing policies or create a custom policy based on your specific needs. To attach existing policies, search for the desired policy in the search bar and select it.

To create a custom policy, click on the “Create policy” button, and a new tab will open where you can define the policy. Once you have selected the desired policies, click on the “Next: Tags” button. Here I will be using “AmazonS3FullAccess” and AmazonSSManagedInstanceCore policies.



Step 5: Add Role Details and Tags (Optional)

Provide a descriptive role name and description of the role. On the “Add tags” page, you can add tags to the role for better organization and management of your resources. Tags are key-value pairs that help in identifying and categorizing resources. Click on the “Add tags”

button and provide the tag key and value. Once you have added the tags, click on the “Next: Review” button.



The screenshot shows the AWS IAM console interface. On the left, a sidebar indicates the current step is 'Step 3: Name, review, and create'. The main content area is titled 'Role details' and contains the following fields:

- Role name:** A text input field containing 'mys3andssmaccess'. Below the field, it states: 'Maximum 64 characters. Use alphanumeric and '+', '@', '-' characters.'
- Description:** A text input field containing 'Allows EC2 instances to call AWS services on your behalf.'. Below the field, it states: 'Maximum 1000 characters. Use alphanumeric and '+', '@', '-' characters.'
- Step 1: Select trusted entities:** A section with an 'Edit' button and a code editor showing a JSON policy snippet:

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": {
```

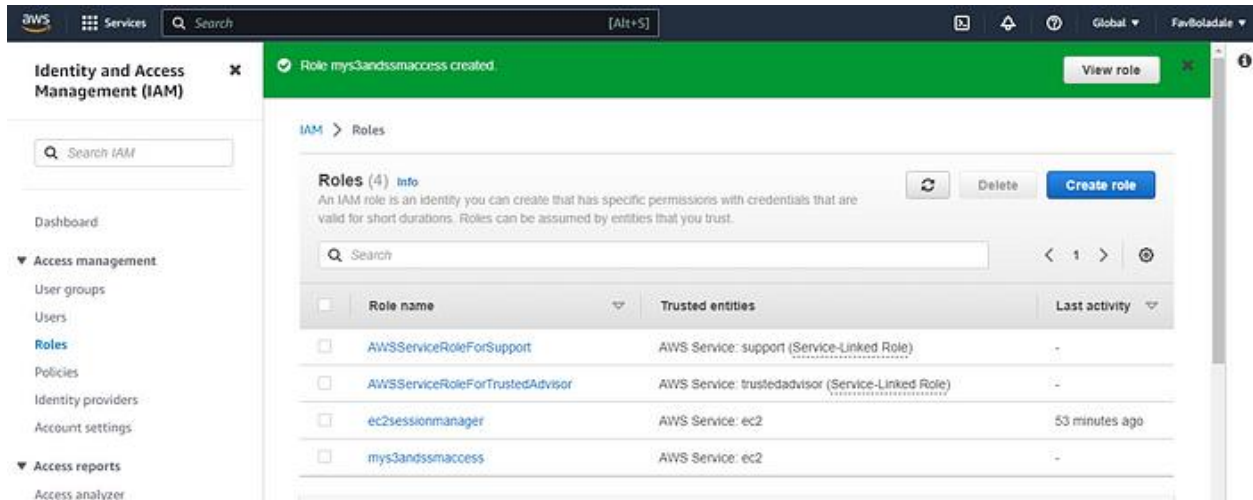
Step 6: Review and Create the Role

On the “Review” page, review the role configuration and ensure that everything is correct. If you need to make any changes, you can go back to the previous steps and modify the settings accordingly.

Once you are satisfied with the configuration, click on the “Create role” button.

Step 7: Role Creation Confirmation

AWS will create the IAM role, and you will be redirected to the “Roles” page. Here, you can see the newly created role and other existing roles.



The screenshot shows the AWS IAM console interface. At the top, a green banner displays a confirmation message: "Role mys3andssmaccess created." with a "View role" button. Below this, the "Roles" page is active, showing a list of four roles. The left sidebar contains navigation links for Identity and Access Management (IAM), including Dashboard, Access management, and Access reports. The main content area includes a search bar, a refresh button, a delete button, and a "Create role" button. The roles are listed in a table with columns for Role name, Trusted entities, and Last activity.

<input type="checkbox"/>	Role name	Trusted entities	Last activity
<input type="checkbox"/>	AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
<input type="checkbox"/>	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
<input type="checkbox"/>	ec2sessionmanager	AWS Service: ec2	53 minutes ago
<input type="checkbox"/>	mys3andssmaccess	AWS Service: ec2	-

CREATING IAM POLICY

Crafting IAM Policies: Practical Examples

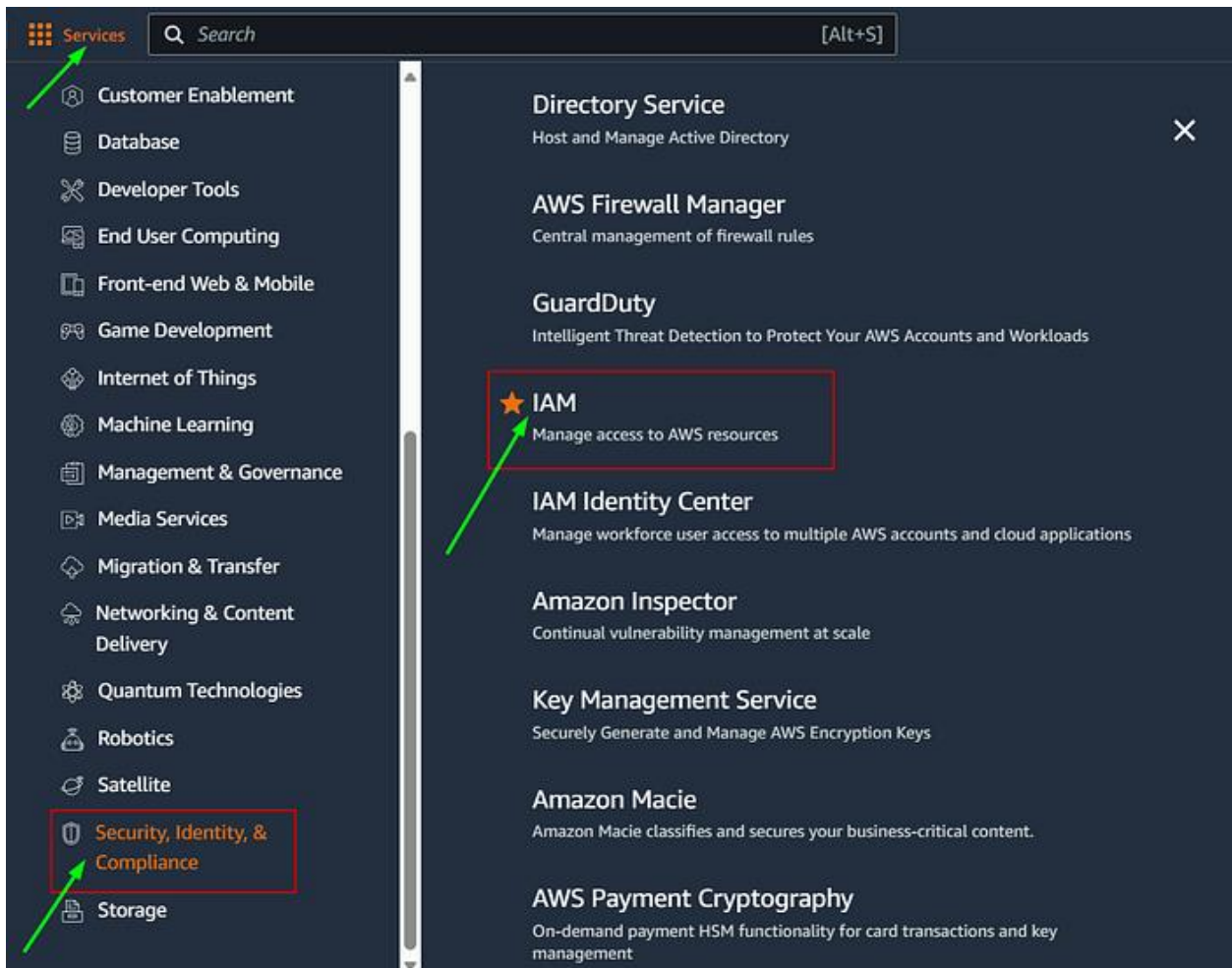
Access the AWS Management Console:

Log in to the AWS Management Console:

<https://aws.amazon.com/console/>.

2. Navigate to IAM:

In the AWS Management Console, go to the “Services” menu and select “IAM” under the “Security, Identity, & Compliance” section.



3. Access the Policy Generator:

In the IAM console, select “Policies” from the left-hand navigation pane.

Identity and Access Management (IAM)



Search IAM

Dashboard

▼ Access management

User groups

Users

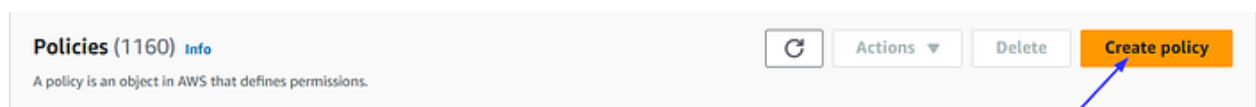
Roles

Policies

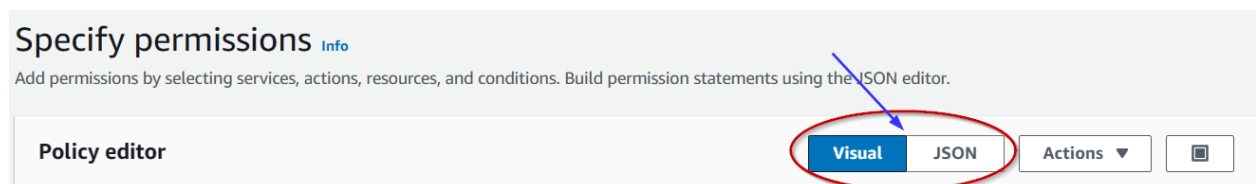
Identity providers

Account settings

Click on the “Create policy” button.



Choose between “Visual” or “JSON” tab from the policy editor.



Use Case 1: EC2 Snapshot Management

⇒ Define Permissions:

Select the service (e.g., EC2) and actions needed for EC2 snapshot management.

The screenshot shows the 'Add permissions' page in the AWS IAM console. The 'Service' dropdown menu is highlighted with a red circle and a blue arrow pointing to it. Below the 'Service' section, the 'EC2' section is expanded, showing 'Set permissions for EC2'. The 'Actions allowed' section is also expanded, showing a search bar for 'Filter Actions'. To the right of the 'Filter Actions' search bar, there is a red box containing the 'Effect' section, which has two radio buttons: 'Allow' (selected) and 'Deny'. A blue arrow points to the 'Allow' radio button.

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

Service
Choose a service ▼

▼ **EC2**
Set permissions for EC2

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

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

Q Filter Actions

Effect
☒ Allow ☐ Deny

⇒ Add actions:

Specify actions like Modify Snapshot Attribute, Reset Snapshot Attribute, etc.

Here's an example of an allowing All-permissions management actions in EC2:

Specify actions from the service to be allowed.

Effect
☒ Allow ☐ Deny

Manual actions | [Add actions](#)

☐ All EC2 actions (ec2:*)

Access level

▶ List (172)

▶ Read (35)

▶ Write (417)

▼ **Permissions management (Selected 5/5)**

☒ All permissions management actions

☒ CreateNetworkInterfacePermission Info ☒ DeleteNetworkInterfacePermission Info ☒ ModifySnapshotAttribute Info

☒ ModifyVpcEndpointServicePermissions Info ☒ ResetSnapshotAttribute Info

▶ Tagging (2)

[Expand all](#) | [Collapse all](#)

⇒ Select Resources:

Choose whether the policy applies to all resources or specific resources.

▼ Resources

Specify resource ARNs for these actions.

☐ All

☒ Specific

If selecting specific resources, specify the ARN (Amazon Resource Name) of the snapshots.

▼ Resources

Specify resource ARNs for these actions.

☐ All

☒ Specific

network-interface	Info	⚠ Specified network-interface resource ARN for the AssignIpv6 Addresses and 31 more actions. Add ARNs to restrict access.	<input type="checkbox"/> Any in this account
snapshot	Info	⚠ Specified snapshot resource ARN for the CopySnapshot and 25 more actions. Add ARNs to restrict access.	<input type="checkbox"/> Any
vpc-endpoint-service	Info	⚠ Specified vpc-endpoint-service resource ARN for the AcceptVpcEndpointConnections and 14 more actions. Add ARNs to restrict access.	<input type="checkbox"/> Any in this account

You can select any resources from this, now I am choosing all actions in snapshot.

▼ Resources

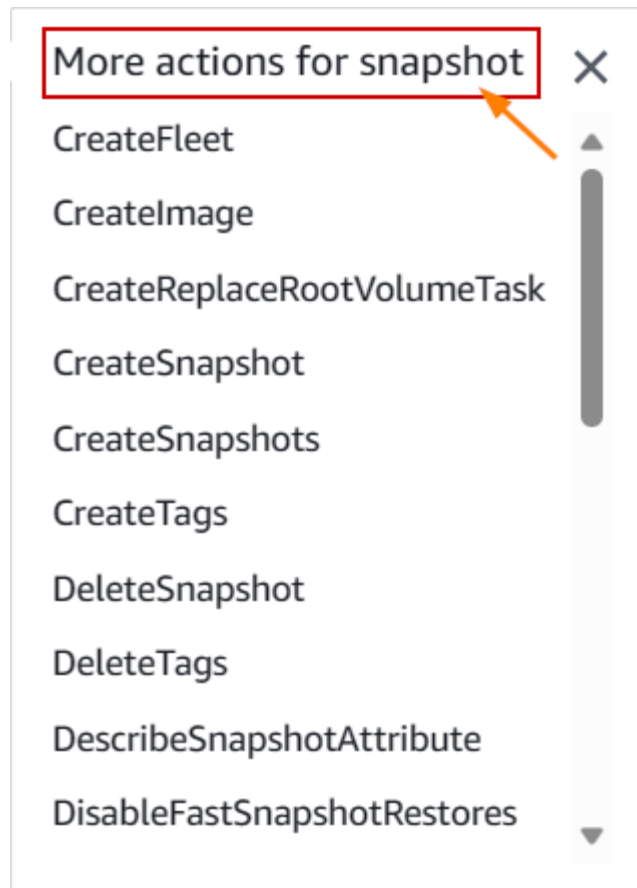
Specify resource ARNs for these actions.

☐ All

☒ Specific

network-interface	Info	⚠ Specified network-interface resource ARN for the AssignIpv6 Addresses and 31 more actions. Add ARNs to restrict access.	<input type="checkbox"/> Any in this account
snapshot	Info	<input type="text" value="arn:aws:ec2::*:snapshot/*"/>	<input checked="" type="checkbox"/> Any
vpc-endpoint-service	Info	⚠ Specified vpc-endpoint-service resource ARN for the AcceptVpcEndpointConnections and 14 more actions. Add ARNs to restrict access.	<input type="checkbox"/> Any in this account

To specify snapshot resource ARNs for the `CopySnapshot` and other actions, you can create an AWS Identity and Access Management (IAM) policy in Visual format.



Then Click “NEXT”



⇒ Review and Create:

After configuring the policy, review the summary.

Click on the “Add” button to generate the policy.

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:

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.

If you want to change the permissions, go to right side of tab and click the “Edit.”

then you can edit the permissions defined in this policy.

Permissions defined in this policy [Info](#)

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 402 services) Show remaining 401 services

Service	Access level	Resource	Request condition
EC2	Limited: Permissions management	SnapshotID string like [All, region] string like [All	None

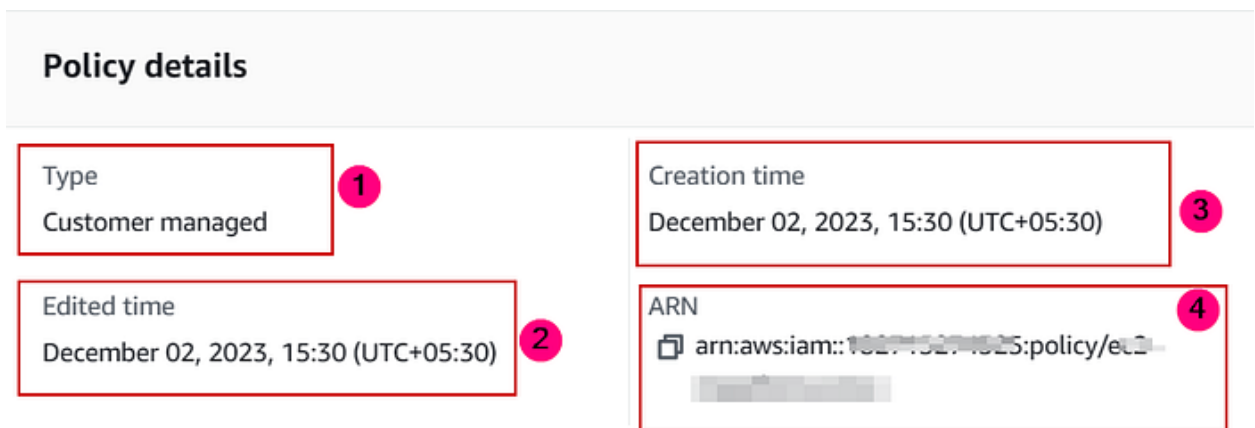
Then click on the “Create policy” icon.



Now we successfully created the policy. If you want to view the policy, click on the “View policy.”



⇒ Policy details:



1. *Type: policy type*

2. *Edited time: When the policy edited*

3. *Creation time: When the policy created*

4. ARN: Policy ARN

Permissions defined in this policy in JSON format:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "ec2:ModifyVpcEndpointServicePermissions",
        "ec2:DeleteNetworkInterfacePermission",
        "ec2:ModifySnapshotAttribute",
        "ec2:CreateNetworkInterfacePermission",
        "ec2:ResetSnapshotAttribute"
      ],
      "Resource": "arn:aws:ec2:*::snapshot/*"
    }
  ]
}
```

⇒ Apply the Policy:

Once generated, you can attach the policy to IAM users, groups, or roles based on your use case.

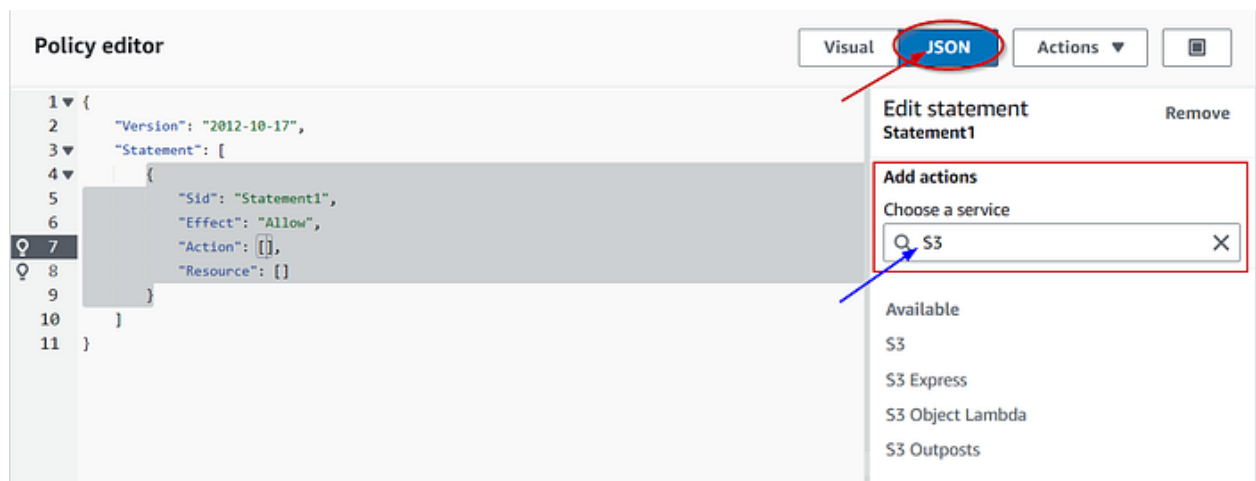
Use Case 2: S3 Bucket Access Control

Using JSON:

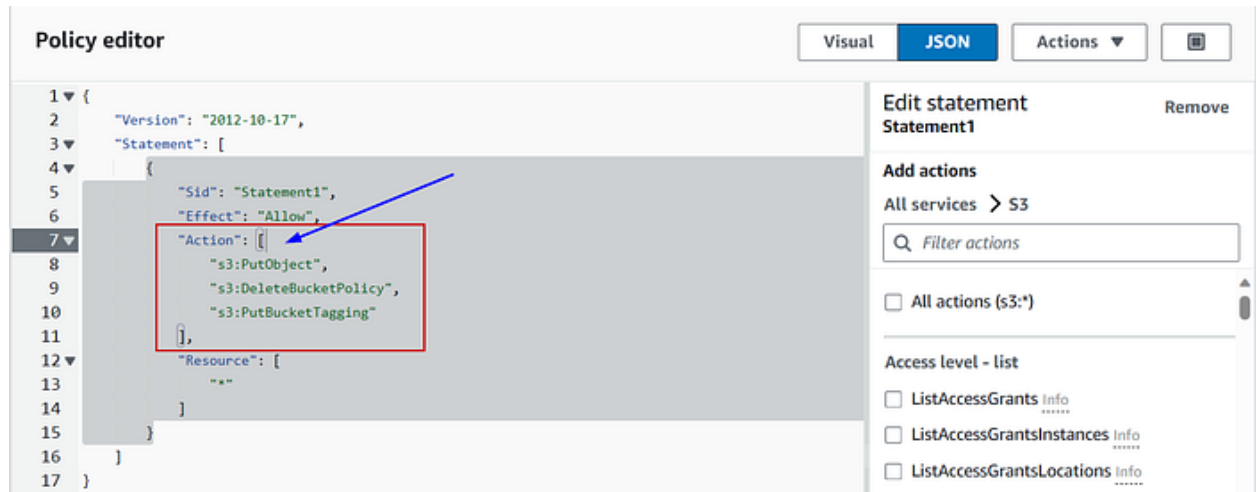
If you are using JSON, you need to write the policy.

⇒ Define Permissions:

Select the service (e.g., S3) and actions needed for bucket access control.



Specify actions like Put Object, Delete Bucket Policy, etc.



⇒ Select Resources:

Choose whether the policy applies to all S3 buckets or specific buckets.

Add resource

×

Specify the resource type and ARN to add for the selected service.

Service

S3

Resource type

All Resources

Resource ARN

*

Cancel

Add resource

If selecting specific buckets, specify the bucket ARN.

Add resource×

Specify the resource type and ARN to add for the selected service.

Service

S3

Resource type

object 1

Resource ARN

arn:aws:s3:::{BucketName}/{ObjectName} 2

Cancel Add resource

1. Resource type: object/bucket

2. Resource ARN:

Bucket Name: your Bucket Name

Object Name: your Object Name

⇒ Review and Create:

After configuring the policy, review the summary.

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.

If you want to change the permissions, go to right side of tab and click the “Edit.”

Click on the “Add” button to generate the policy.


Permissions defined in this policy [Info](#)
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 402 services) ☐ Show remaining 401 services

Service	Access level	Resource	Request condition
S3	Limited: Permissions management, Tagging, Write	All resources	None

⇒ Policy details:

Policy details

Type Customer managed	Creation time December 03, 2023, 11:31 (UTC+05:30)	Edited time December 03, 2023, 11:31 (UTC+05:30)	ARN  arn:aws:iam::[redacted]:policy/[redacted]
--------------------------	---	---	---

In ARN

arn:aws:iam::"Account ID":policy/"Policy Name"

1. The Account ID (who create the policy)

2. The Policy Name (Given by you)

The JSON Policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:DeleteBucketPolicy",
        "s3:PutBucketTagging"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

By following these step-by-step instructions for each use case, you can craft IAM policies tailored to your specific needs for EC2 snapshot management and S3 bucket access control.