



CLOUD TRAIN
ACCELERATE YOUR GROWTH

MODULE 7

IAM & MONITORING

AWS Workshop

Contact us

TO ACCELERATE YOUR CAREER GROWTH

For questions and more details:

please call @ +91 98712 72900, or

visit <https://www.thecloudtrain.com/>, or

email at support@thecloudtrain.com, or

WhatsApp us @ +91 98712 72900

Setup IAM User, Group and Roles management

To create one or more IAM users (console)

Step 1. Sign in to the AWS Management Console and open the IAM console at

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

Step 2. In the navigation pane, choose **Users** and then choose **Add users**.

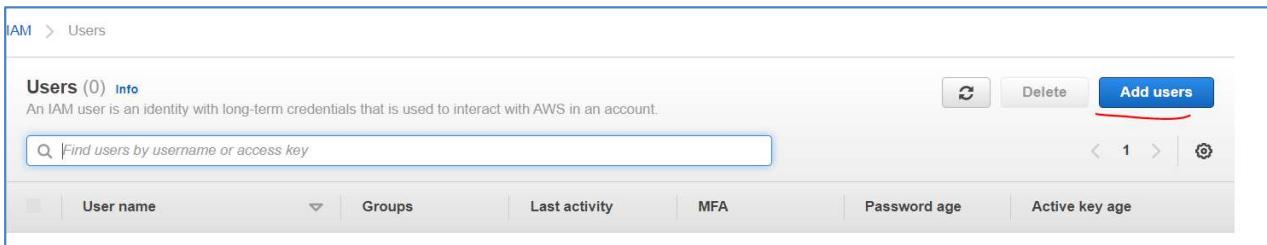


The screenshot shows the AWS IAM Dashboard. On the left, the navigation pane is visible with the following structure:

- Identity and Access Management (IAM) (selected)
- Dashboard (highlighted in orange)
- Access management
 - User groups
 - Users** (highlighted with a red underline)
 - Roles

The main dashboard displays the following information:

- IAM dashboard
- Sign-in URL for IAM users in this account: <https://090322976386.signin.aws.amazon.com/console> | Customize
- IAM resources
 - Users: 0
 - User groups: 0
 - Customer managed policies: 0
- Roles: 10
- Identity providers:



The screenshot shows the 'Users' page under the IAM service. The navigation bar indicates 'IAM > Users'. The page displays the following details:

- Users (0) Info
- An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.
- A search bar with placeholder text 'Find users by username or access key'.
- Action buttons: 'Edit' (with a red underline), 'Delete', and 'Add users' (highlighted with a red underline).
- Pagination controls: '< 1 >' and a refresh icon.
- A table header with columns: User name, Groups, Last activity, MFA, Password age, and Active key age.

Step 3. Type the user name for the new user, say awsuser1. This is the sign-in name for AWS. If you want to add multiple users, choose **Add another user** for each additional user and type their user names. You can add up to 10 users at one time.

Note: The number and size of IAM resources in an AWS account are limited. User names can be a combination of up to 64 letters, digits, and these characters: plus (+), equal (=), comma (,), period (.), at sign (@), underscore (_), and hyphen (-). Names must be unique within an account. They are not distinguished by case. For example, you cannot create two users named *TESTUSER* and *testuser*.

Step 4. Select the type of access this set of users will have. You can select programmatic access, access to the AWS Management Console, or both.

- Select **Programmatic access** if the users require access to the API, AWS CLI, or Tools for Windows PowerShell. This creates an access key for each new user. You can view or download the access keys when you get to the **Final** page.
- Select **AWS Management Console access** if the users require access to the AWS Management Console. This creates a password for each new user. **This one we use here:**

For **Console password**, choose one of the following:

- **Autogenerated password**. Each user gets a randomly generated password that meets the [account password policy](#). You can view or download the passwords when you get to the **Final** page.
- **Custom password**. Each user is assigned the password that you type in the box.
(Optional) We recommend that you select **Require password reset** to ensure that users are forced to change their password the first time they sign in.

Note: If an administrator has enabled the **Allow users to change their own password** account password policy setting, then this check box does nothing. Otherwise, it automatically attaches an AWS managed policy named [IAMUserChangePassword](#) to the new users. The policy grants them permission to change their own passwords.

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*	<input type="text" value="awsuser1"/>
+ Add another user	

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type*	<input type="checkbox"/> Programmatic access Enables an access key ID and secret access key for the AWS API, CLI, SDK, and other development tools.
	<input checked="" type="checkbox"/> AWS Management Console access Enables a password that allows users to sign-in to the AWS Management Console.

Console password*

<input type="radio"/> Autogenerated password
<input checked="" type="radio"/> Custom password

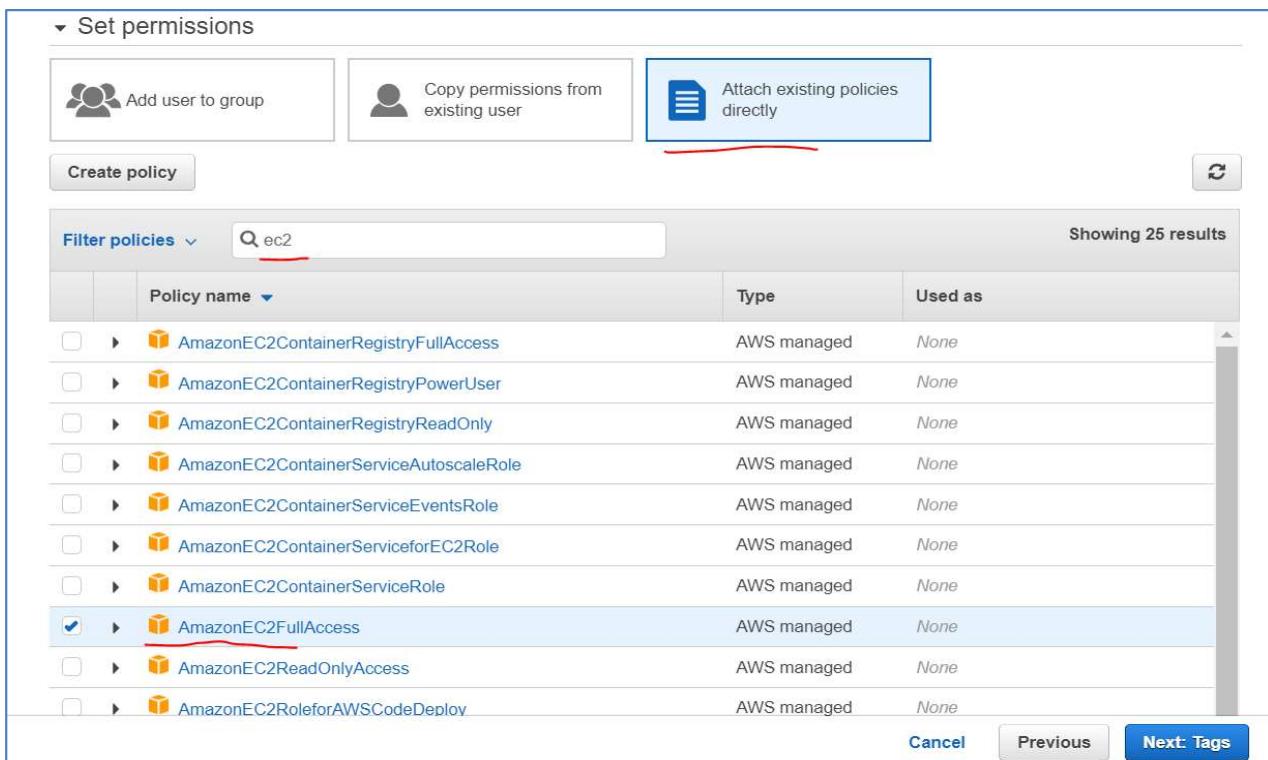
Show password

Require password reset User must create a new password at next sign-in
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

* Required [Cancel](#) Next: Permissions

Step 5. Choose **Next: Permissions**.**Step 6.** On the **Set permissions** page, specify how you want to assign permissions to this set of new users.**Step 7.** Choose one of the following three options:

- **Add user to group.** Choose this option if you want to assign the users to one or more groups that already have permissions policies. IAM displays a list of the groups in your account, along with their attached policies. You can select one or more existing groups, or choose **Create group** to create a new group. For more information, see [Changing permissions for an IAM user](#).
- **Copy permissions from existing user.** Choose this option to copy all of the group memberships, attached managed policies, embedded inline policies, and any existing [permissions boundaries](#) from an existing user to the new users. IAM displays a list of the users in your account. Select the one whose permissions most closely match the needs of your new users.
- **Attach existing policies directly.** Choose this option to see a list of the AWS managed and customer managed policies in your account. Select the policies that you want to attach to the new users .



The screenshot shows the 'Set permissions' step of the IAM wizard. At the top, there are three options: 'Add user to group', 'Copy permissions from existing user', and 'Attach existing policies directly'. The 'Attach existing policies directly' button is highlighted with a red underline. Below it is a 'Create policy' button. A search bar with the text 'ec2' is shown above a list of policies. The list includes various AWS managed policies, with 'AmazonEC2FullAccess' being checked and underlined in red. The table columns are 'Policy name', 'Type', and 'Used as'. At the bottom right are 'Cancel', 'Previous', and 'Next: Tags' buttons.

Policy name	Type	Used as
AmazonEC2ContainerRegistryFullAccess	AWS managed	None
AmazonEC2ContainerRegistryPowerUser	AWS managed	None
AmazonEC2ContainerRegistryReadOnly	AWS managed	None
AmazonEC2ContainerServiceAutoscaleRole	AWS managed	None
AmazonEC2ContainerServiceEventsRole	AWS managed	None
AmazonEC2ContainerServiceforEC2Role	AWS managed	None
AmazonEC2ContainerServiceRole	AWS managed	None
AmazonEC2FullAccess	AWS managed	None
AmazonEC2ReadOnlyAccess	AWS managed	None
AmazonEC2RoleforAWSCodeDeploy	AWS managed	None

Step 8. Choose **Next: Tags**.

Add tags (optional)

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. [Learn more](#)

Key	Value (optional)	Remove
name	ec2user	x
Add new key		

Step 9. Choose **Next: Review** to see all of the choices you made up to this point. When you are ready to proceed, choose **Create user**.

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	awsuser1
AWS access type	AWS Management Console access - with a password
Console password type	Custom
Require password reset	No
Permissions boundary	Permissions boundary is not set

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	AmazonEC2FullAccess

Tags

The new user will receive the following tag

Key	Value

[Cancel](#)[Previous](#)[Create user](#)**Step 10.** To view the users' access keys (access key IDs and secret access keys), choose **Show** next to each password and access key that you want to see. To save the access keys, choose **Download .csv** and then save the file to a safe location.

 **Success**

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://090322976386.signin.aws.amazon.com/console>

[!\[\]\(f8e7be3c2bd30232a05cdc54a8b2d22a_img.jpg\) Download .csv](#)

	User	Email login instructions
	awsuser1	Send email 

Important: This is your only opportunity to view or download the secret access keys, and you must provide this information to your users before they can use the AWS API. Save the user's new access key ID and secret access key in a safe and secure place. **You will not have access to the secret keys again after this step.**

Step 11. (Optional) Provide each user with his or her credentials. On the final page you can choose **Send email** next to each user. Your local mail client opens with a draft that you can customize and send. The email template includes the following details to each user:

- User name
- URL to the account sign-in page. Use the following example, substituting the correct account ID number or account alias:

`https://AWS-account-ID or alias.signin.aws.amazon.com/console`

- User can get the login URL from credentials tab that you emailed him from user summary and login to it:

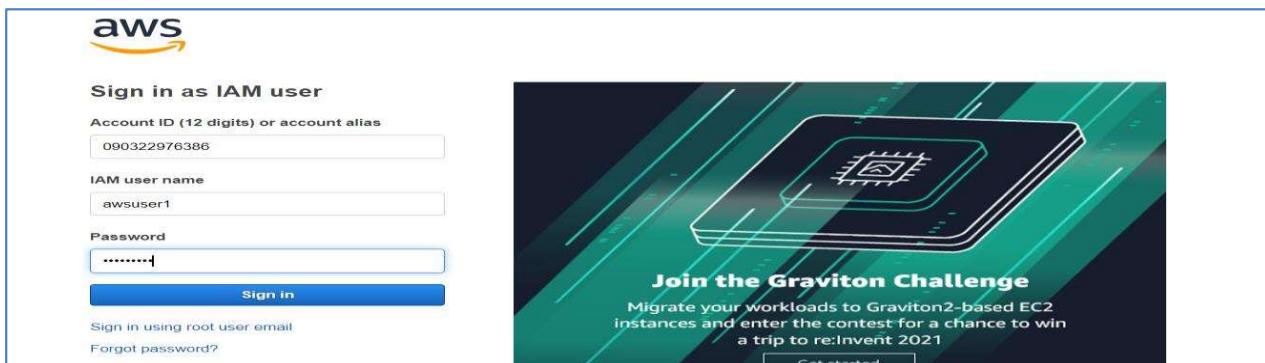
Summary

User ARN	arn:aws:iam::090322976386:user/awsuser1 Edit
Path	/
Creation time	2021-08-05 15:24 UTC+0530

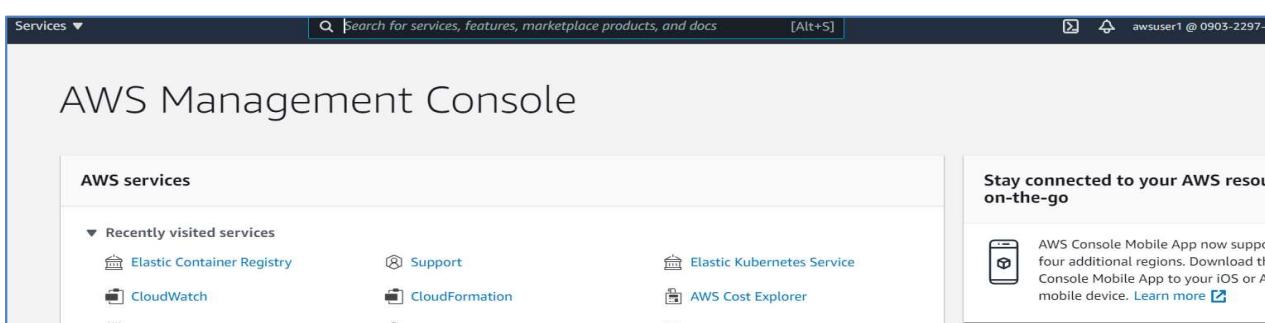
Permissions Groups Tags (1) Security credentials Access Advisor

Sign-in credentials

Summary	• Console sign-in link: https://090322976386.signin.aws.amazon.com/console Edit
Console password	Enabled (never signed in) Manage
Assigned MFA device	Not assigned Manage
Signing certificates	None Edit

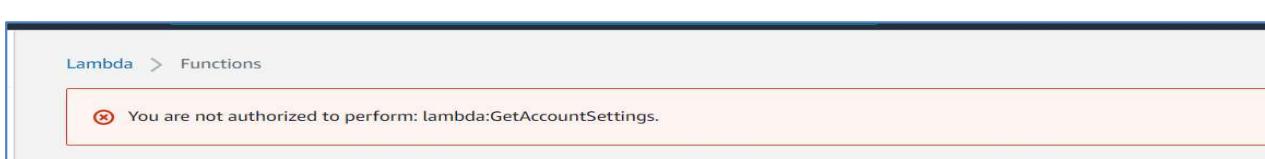


The screenshot shows the AWS Sign in as IAM user page. The user has entered the account ID (090322976386), IAM user name (awsuser1), and a password. Below the form is a "Sign in" button. To the right of the form is a promotional banner for the "Join the Graviton Challenge". The banner features a stylized computer chip and text encouraging users to migrate workloads to Graviton2-based EC2 instances for a chance to win a trip to re:Invent 2021.



The screenshot shows the AWS Management Console homepage. The top navigation bar includes "Services ▾", a search bar ("Search for services, features, marketplace products, and docs [Alt+S]"), and a user profile icon ("awsuser1 @ 0903-2297-6"). The main content area is titled "AWS Management Console". On the left, there's a sidebar titled "AWS services" with a "Recently visited services" section listing Elastic Container Registry, CloudWatch, and EC2. To the right, a sidebar titled "Stay connected to your AWS resources on-the-go" promotes the AWS Console Mobile App, mentioning support for four additional regions and providing a download link.

This user can access only EC2 services. If he tries to access any other, for e.g. Lambda, then below error will occur

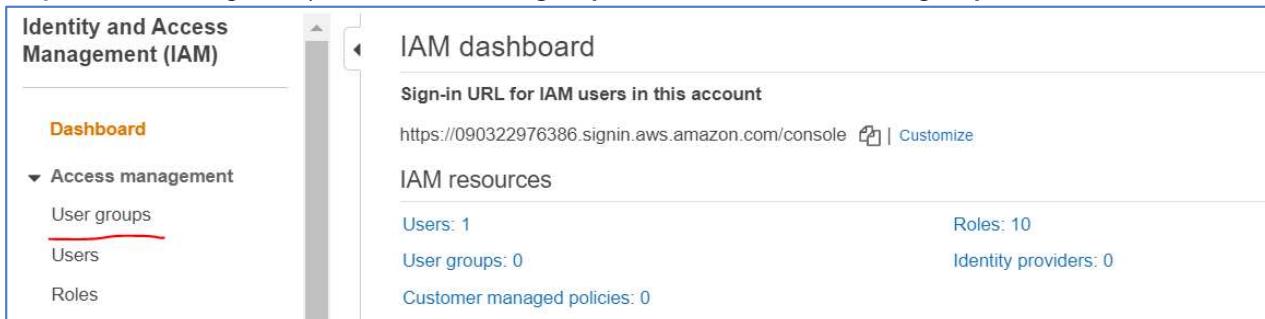


The screenshot shows the AWS Lambda > Functions page. The user is trying to access the "lambda:GetAccountSettings" function, but receives an error message: "You are not authorized to perform: lambda:GetAccountSettings." This indicates that the user does not have the necessary permissions to access certain AWS services.

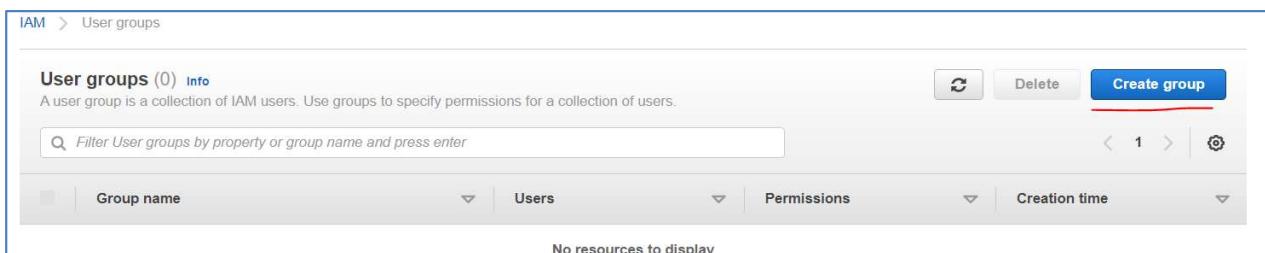
To create an IAM user group and attach policies (console)

Step 1. Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.

Step 2. In the navigation pane, choose **User groups** and then choose **Create group**.



The screenshot shows the IAM dashboard. On the left, the navigation pane has 'Identity and Access Management (IAM)' selected. Under 'Access management', 'User groups' is highlighted with a red underline. The main area shows the 'IAM dashboard' with statistics: 'Sign-in URL for IAM users in this account' (https://090322976386.signin.aws.amazon.com/console), 'Users: 1', 'User groups: 0', 'Roles: 10', and 'Identity providers: 0'. Below these stats, it says 'Customer managed policies: 0'.



The screenshot shows the 'User groups' page. The top navigation bar includes 'IAM > User groups'. The main content area shows 'User groups (0) Info' with a note: 'A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.' Below this is a search bar with the placeholder 'Filter User groups by property or group name and press enter'. To the right are buttons for 'Create group' (highlighted with a red underline), 'Delete', and 'Actions'. A pagination indicator shows '1' of '1'. At the bottom, there are filters for 'Group name', 'Users', 'Permissions', and 'Creation time', and a message 'No resources to display'.

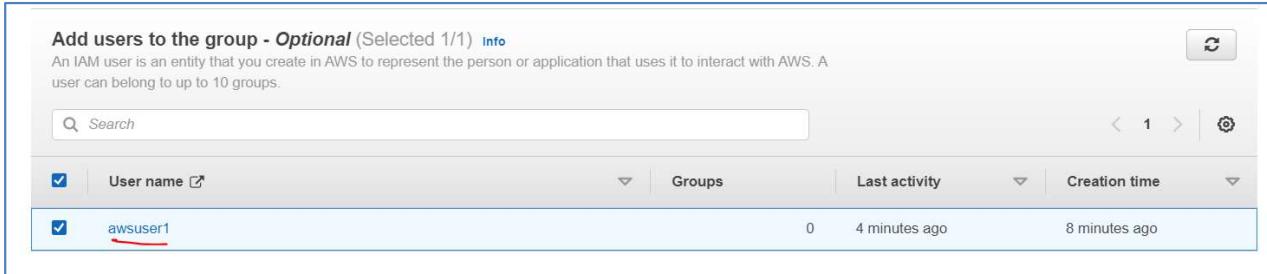
Step 3. For **User group name**, type the name of the group, say **awsgrp1**.

Note: The number and size of IAM resources in an AWS account are limited. For more information, see [IAM and AWS STS quotas](#). Group names can be a combination of up to 128 letters, digits, and these characters: plus (+), equal (=), comma (,), period (.), at sign (@), underscore (_), and hyphen (-). Names must be unique within an account. They are not distinguished by case. For example, you cannot create groups named both **ADMINS** and **admins**.



The screenshot shows the 'Create user group' form. The title is 'Create user group'. The first section is 'Name the group' with a 'User group name' input field containing 'awsgrp1'. Below the input field is a note: 'Enter a meaningful name to identify this group.' and a character limit note: 'Maximum 128 characters. Use alphanumeric and '+,=,@,_' characters.'

Step 4. In the list of users, select the check box for each user that you want to add to the group.

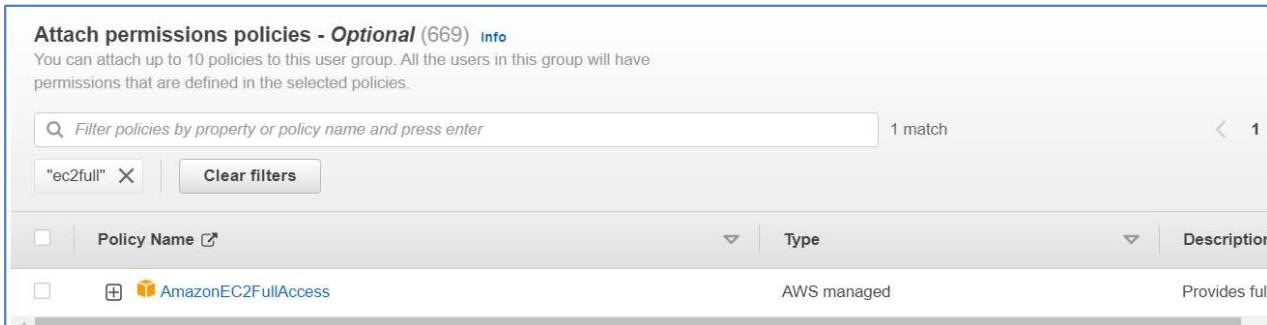


Add users to the group - *Optional* (Selected 1/1) [Info](#)

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.

User name	Groups	Last activity	Creation time
<input checked="" type="checkbox"/> awsuser1	0	4 minutes ago	8 minutes ago

Step 5. In the list of policies, select the check box for each policy that you want to apply to all members of the group.

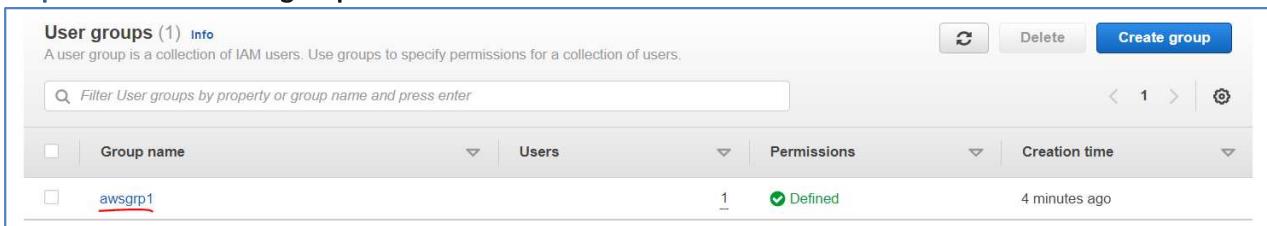


Attach permissions policies - *Optional* (669) [Info](#)

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.

Policy Name	Type	Description
<input checked="" type="checkbox"/> AmazonEC2FullAccess	AWS managed	Provides full

Step 6. Choose **Create group**.



User groups (1) [Info](#)

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

Group name	Users	Permissions	Creation time
<input checked="" type="checkbox"/> awsgrp1	1	Defined	4 minutes ago

To create a IAM role (console)

Step 1. Sign in to the AWS Management Console and open the IAM console at

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

Step 2. In the navigation pane of the console, choose **Roles** and then choose **Create role**.



Dashboard

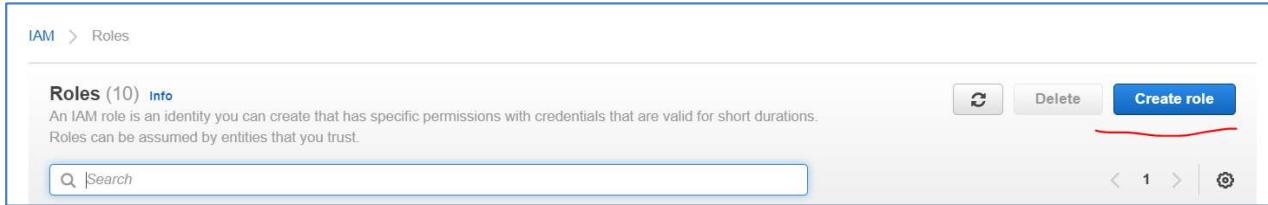
Access management

- User groups
- Users
- Roles**

<https://090322976386.signin.aws.amazon.com/console>  | [Customize](#)

IAM resources

Users: 1	Roles: 10
User groups: 1	Identity providers: 0
Customer managed policies: 0	



Step 3. Choose the **Another AWS account** role type.

Step 4. For **Account ID**, type the AWS account ID to which you want to grant access to your resources.

Use the account ID of AWS user we created earlier, for e.g. **090322976386**

The administrator of the specified account can grant permission to assume this role to any IAM user in that account. To do this, the administrator attaches a policy to the user or a group that grants permission for the `sts:AssumeRole` action. That policy must specify the role's ARN as the `Resource`.

Step 5. If you are granting permissions to users from an account that you do not control, and the users will assume this role programmatically, then select **Require external ID**. The external ID can be any word or number that is agreed upon between you and the administrator of the third-party account. This option automatically adds a condition to the trust policy that allows the user to assume the role only if the request includes the correct `sts:ExternalID`.

Important: Choosing this option restricts access to the role only through the AWS CLI, Tools for Windows PowerShell, or the AWS API. This is because you cannot use the AWS console to switch to a role that has an `externalId` condition in its trust policy. However, you can create this kind of access programmatically by writing a script or an application using the relevant SDK.

Step 6. If you want to restrict the role to users who sign in with multi-factor authentication (MFA), select **Require MFA**. This adds a condition to the role's trust policy that checks for an MFA sign-in. A user who wants to assume the role must sign in with a temporary one-time password from a configured MFA device. Users without MFA authentication cannot assume the role

Select type of trusted entity

AWS service EC2, Lambda and others

Another AWS account Belonging to you or 3rd party

Web identity Cognito or any OpenID provider

SAML 2.0 federation Your corporate directory

Allows entities in other accounts to perform actions in this account. [Learn more](#)

Specify accounts that can use this role

Account ID*

Options Require external ID (Best practice when a third party will assume this role)
 Require MFA 

Step 7. Choose **Next: Permissions**.

Step 8. IAM includes a list of the AWS managed and customer managed policies in your account. Select the policy to use for the permissions policy or choose **Create policy** to open a new browser tab and create a new policy from scratch.

Step 9. After you create the policy, close that tab and return to your original tab. Select the check box next to the permissions policies that you want anyone who assumes the role to have. If you prefer, you can select no policies at this time, and then attach policies to the role later. By default, a role has no permissions.

Create role

1 2 3 4

▼ Attach permissions policies

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

Create policy 

Filter policies ▾ Showing 3 results

Policy name	Used as
<input checked="" type="checkbox"/> AmazonEC2FullAccess	Permissions policy (2)
<input type="checkbox"/> AWSEC2FleetServiceRolePolicy	None
<input type="checkbox"/> EC2FleetTimeShiftableServiceRolePolicy	None

Step 10. Choose Next: Tags.

Create role

1 2 3 **4**

Add tags (optional)

IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. [Learn more](#)

Key	Value (optional)	Remove
name	awsrole1	X
Add new key		

Step 11. Choose Next: Review.

Step 12. For **Role name**, type a name for your role, say **awsrole1**. Role names must be unique within your AWS account. They are not distinguished by case. For example, you cannot create roles named both **PRODROLE** and **prodrole**. Because other AWS resources might reference the role, you cannot edit the name of the role after it has been created.

Step 13. (Optional) For **Role description**, type a description for the new role.

Review

Provide the required information below and review this role before you create it.

Role name*	awsrole1
Use alphanumeric and '+-=.,@-_' characters. Maximum 64 characters.	
Role description	
Maximum 1000 characters. Use alphanumeric and '+-=.,@-_' characters.	
Trusted entities	The account 090322976386
Policies	 AmazonEC2FullAccess 

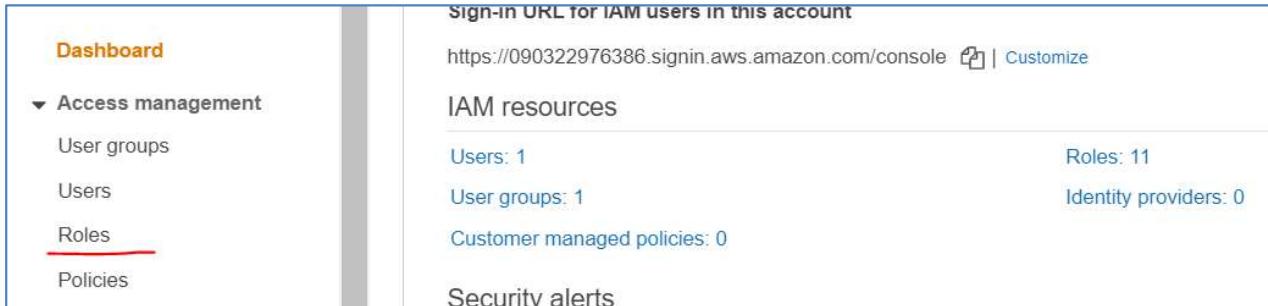
Step 14. Review the role and then choose **Create role**. Once created, it can be seen under roles list:

Roles (11) 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.		Edit Delete Create role	
<input type="checkbox"/>	Role name	Trusted entities	Last activity
<input type="checkbox"/>	awsrole1	Account: 090322976386	-

Enable S3 access from EC2 by IAM role

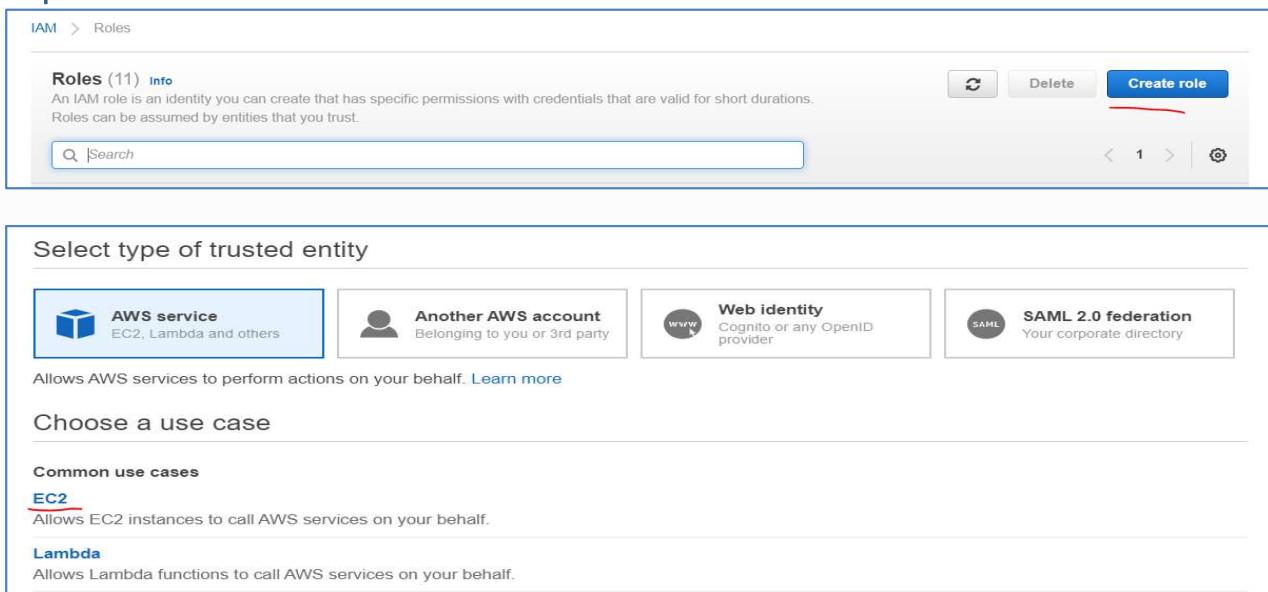
Create a new IAM role

Step 1. Choose “IAM” in the AWS main console:



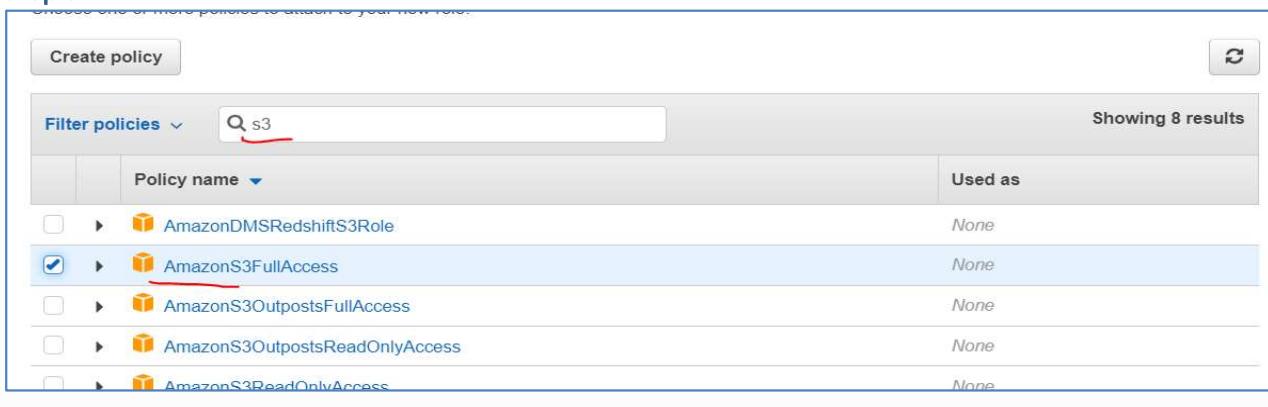
The screenshot shows the AWS IAM Dashboard. On the left, there's a sidebar with "Dashboard" at the top, followed by a "▼ Access management" section containing "User groups", "Users", "Roles" (which is underlined in red), and "Policies". To the right, there's a "Sign-in URL for IAM users in this account" field with the URL <https://090322976386.sigin.aws.amazon.com/console>. Below it is a "Customize" button. The main area displays "IAM resources" with counts: "Users: 1", "User groups: 1", "Roles: 11", and "Identity providers: 0". It also shows "Customer managed policies: 0" and a "Security alerts" section.

Step 2. Then choose “Roles” in the IAM console and click on “Create role”:



The screenshot shows the "Create role" wizard. Step 1 is "Select type of trusted entity". It has four options: "AWS service" (selected, showing "EC2, Lambda and others"), "Another AWS account" (showing "Belonging to you or 3rd party"), "Web identity" (showing "Cognito or any OpenID provider"), and "SAML 2.0 federation" (showing "Your corporate directory"). Below this, a note says "Allows AWS services to perform actions on your behalf." with a "Learn more" link. Step 2 is "Choose a use case". It has two sections: "Common use cases" with "EC2" (selected, showing "Allows EC2 instances to call AWS services on your behalf.") and "Lambda" (showing "Allows Lambda functions to call AWS services on your behalf."). There are also "Next Step" and "Cancel" buttons at the bottom.

Step 3. Search for “S3” and then select “AmazonS3FullAccess”:



The screenshot shows the "Create policy" wizard. Step 2 is "Filter policies". A search bar contains "s3". The results table shows eight policies: "AmazonDMSRedshiftS3Role" (unchecked), "AmazonS3FullAccess" (checked and underlined in red), "AmazonS3OutpostsFullAccess" (unchecked), "AmazonS3OutpostsReadOnlyAccess" (unchecked), and "AmazonS3ReadOnlyAccess" (unchecked). The table has columns for "Policy name" and "Used as". There are "Create policy" and "Cancel" buttons at the top and bottom respectively.

Add tags (optional)

IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. [Learn more](#)

Key	Value (optional)	Remove
name	s3fullaccess	x

Step 4. Finally, give this role a descriptive name. Here I use “full_S3_access_from_EC2”. (For the “Role description”, enter whatever you like or just keep default.)

Create role

1 2 3 4

Review

Provide the required information below and review this role before you create it.

Role name* full_S3_access_from_EC2

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

Role description

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=_,@-_` characters.

Trusted entities AWS service: ec2.amazonaws.com

Policies  AmazonS3FullAccess 

Permissions boundary Permissions boundary is not set

The new role will receive the following tag

Key	Value
name	s3fullaccess

* Required

Cancel

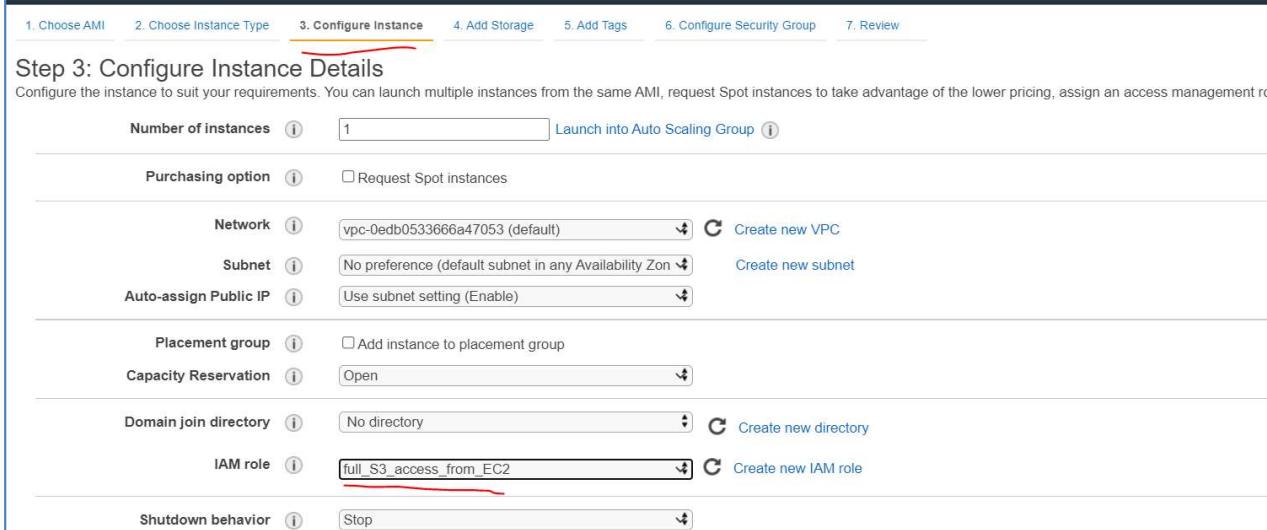
Previous

Create role

Step 5. Now a new IAM role is created. This only needs to be done once.

Assign that role to EC2

Step 1. Whenever you launch a new EC2 instance, in “Step 3: Configure Instance Details”, select the IAM role you created previously for the “IAM role” option.



The screenshot shows the 'Step 3: Configure Instance Details' section of the AWS CloudFormation wizard. The 'IAM role' dropdown menu is open, showing the option 'full_S3_access_from_EC2' which is highlighted with a red box. Other options in the dropdown include 'Create new IAM role'. The rest of the form includes fields for Number of instances (1), Purchasing option (Request Spot instances), Network (vpc-0edb0533666a47053), Subnet (No preference), Auto-assign Public IP (Use subnet setting), Placement group (Add instance to placement group), Capacity Reservation (Open), Domain join directory (No directory), and Shutdown behavior (Stop).

Step 2. No need to touch other options on this page and just launch as usual. On this EC2 instance, you don't need to run `aws configure`, and commands like `aws s3 ls` will just work (as long as AWSCLI is installed). This is actually a better practice since you never type your security credentials on this server (which might be stolen if your server gets hacked).

```
ssh -i "awskey.pem" ec2-user@ec2-3-239-195-198.compute-1.amazonaws.com
```

```
aws s3 ls
```

```
PS C:\Users\Public\Downloads> ssh -i "awskey.pem" ec2-user@ec2-3-239-195-198.compute-1.amazonaws.com
The authenticity of host 'ec2-3-239-195-198.compute-1.amazonaws.com (3.239.195.198)' can't be established.
ECDSA key fingerprint is SHA256:3QCdZ4e3opSAisXp2H+Sxn1sh77FmSwAGV36WmcnzE.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-3-239-195-198.compute-1.amazonaws.com,3.239.195.198' (ECDSA) to the list of known hosts.

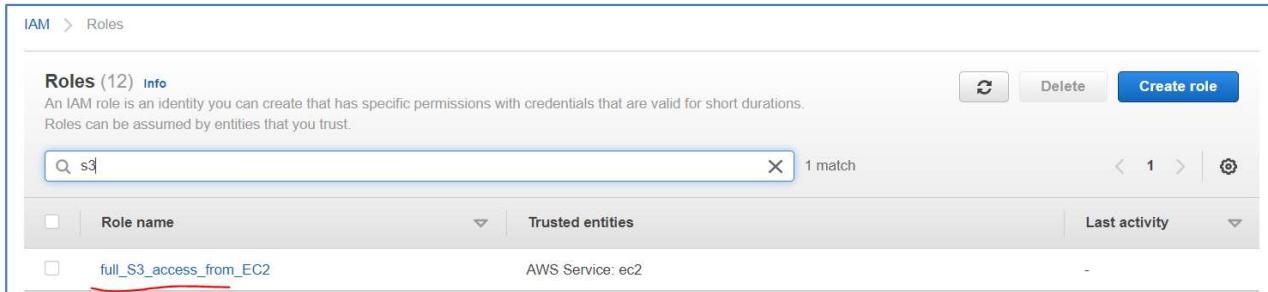
[ec2-user@ip-172-31-7-68 ~]$ aws s3 ls
2021-08-04 06:47:28 bucket0821
[ec2-user@ip-172-31-7-68 ~]$
```

Create a JSON document using access policy

For reference policies, refer

https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_examples.html

Step 1. Use the same roles that we created in previous task i.e. **full_S3_access_from_EC2** and lets create a policy to block its access to S3 completely. Go to **IAM -> Roles**:



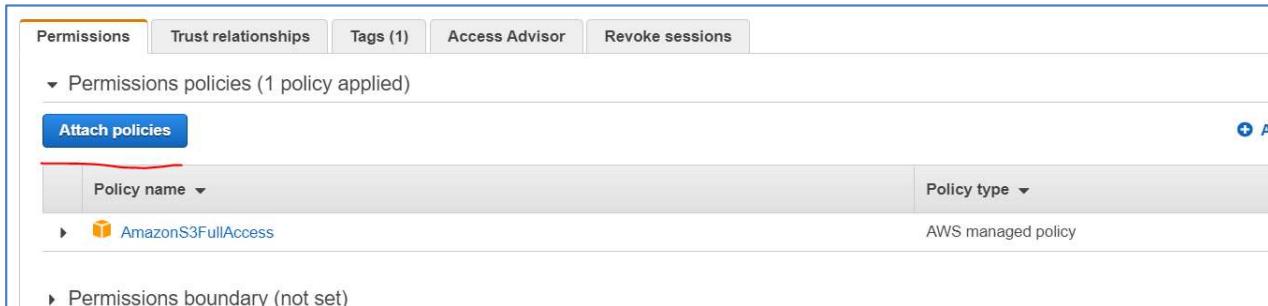
IAM > Roles

Roles (12) 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: s3

Role name	Trusted entities	Last activity
full_S3_access_from_EC2	AWS Service: ec2	-

Step 2. Click on the role **full_S3_access_from_EC2** and select Attach policies:



Permissions Trust relationships Tags (1) Access Advisor Revoke sessions

▼ Permissions policies (1 policy applied)

Attach policies + A

Policy name	Policy type
AmazonS3FullAccess	AWS managed policy

▶ Permissions boundary (not set)

Step 3. Now, we have option to select from existing and create our own. Here, we will create own custom policy to block complete S3 access for this role. So, click on **Create Policy**:



Add permissions to **full_S3_access_from_EC2**

Attach Permissions

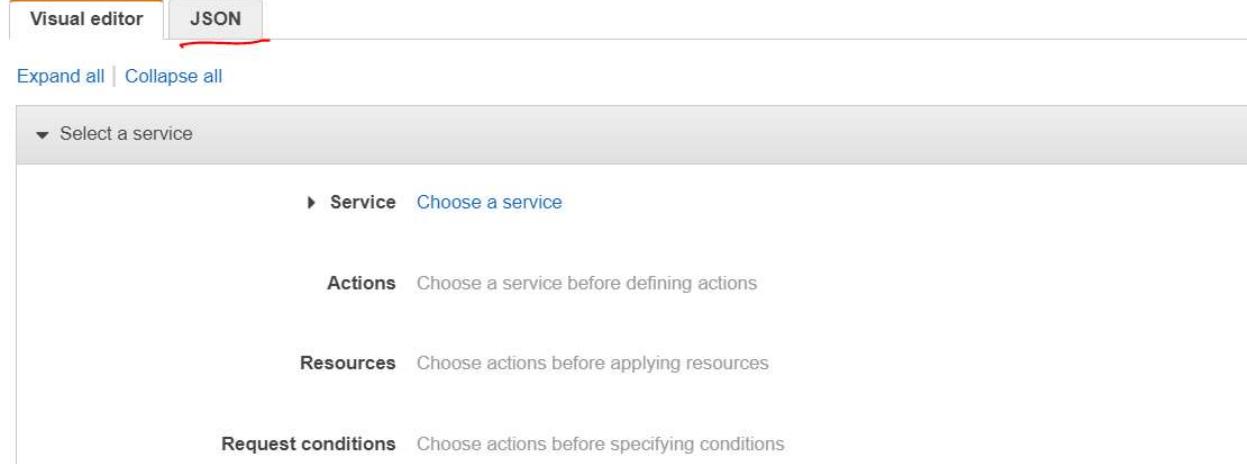
Create policy (Red Underline)

Filter policies ▼ Search

Policy name ▼

Step 4. You see a policy editor with two options. In visual editor, you can pick every component one by one and derive the policy, while in JSON format you can write your policy in once go. That's what we are going to do here:

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON.



The screenshot shows the AWS Policy Editor interface. At the top, there are two tabs: "Visual editor" and "JSON". The "JSON" tab is highlighted with a red underline. Below the tabs, there are buttons for "Expand all" and "Collapse all". A dropdown menu labeled "Select a service" is open, showing a list of services with "Service" and "Choose a service" options. Below this, there are sections for "Actions", "Resources", and "Request conditions", each with a "Choose a service before defining actions" link. The entire interface is contained within a blue-bordered box.

Step 5. Copy the below content and paste to JSON tab and validate its do not have any error. Go to add tags then:

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

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and

Visual editor

JSON

```
1 {  
2     "Version": "2012-10-17",  
3     "Statement": [  
4         {  
5             "Effect": "Deny",  
6             "Action": "s3:*",  
7             "Resource": "*"  
8         }  
9     ]  
10 }
```

Step 6. Add tags if you want or skip to review:

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

Next: Review

Step 7. Give your policy a name, say '**s3blockallaccess**' and click on **Create**.

Review policy

Name*

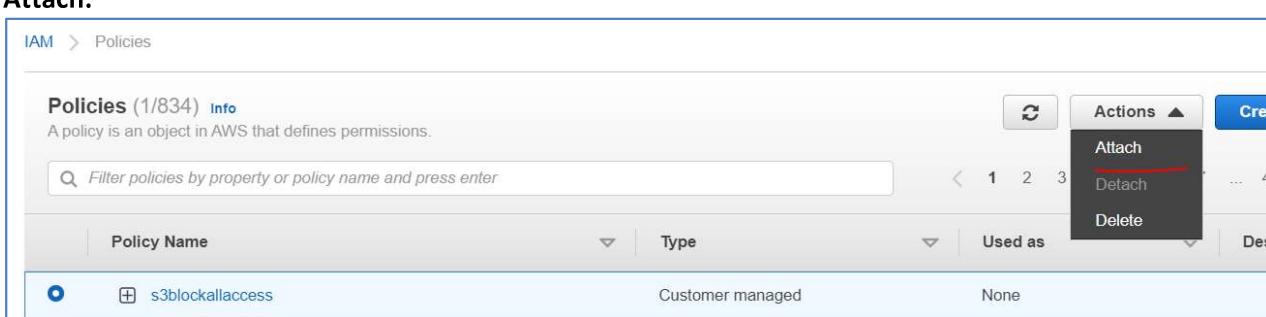
s3blockallaccess

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

Description

Maximum 1000 characters. Use alphanumeric and '+=_@-' characters.

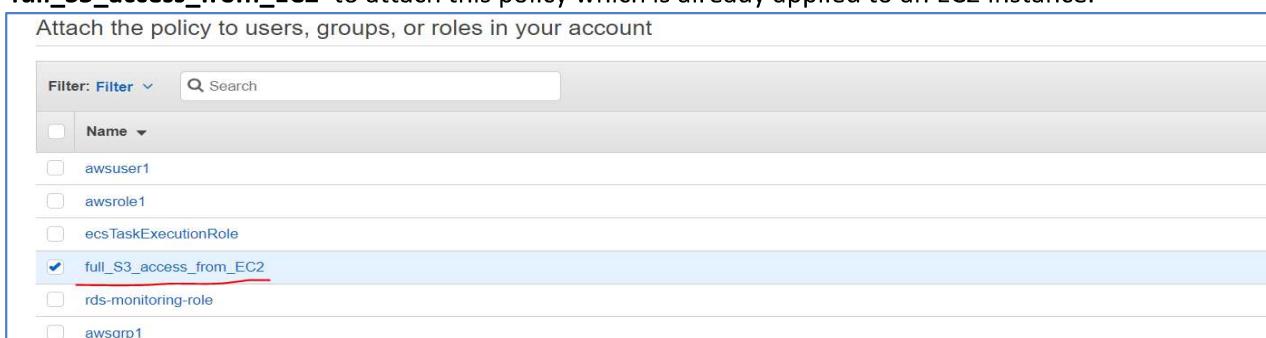
Step 8. Once your policy is created, it will be listed under Policies tab. Select it and select **Actions -> Attach:**



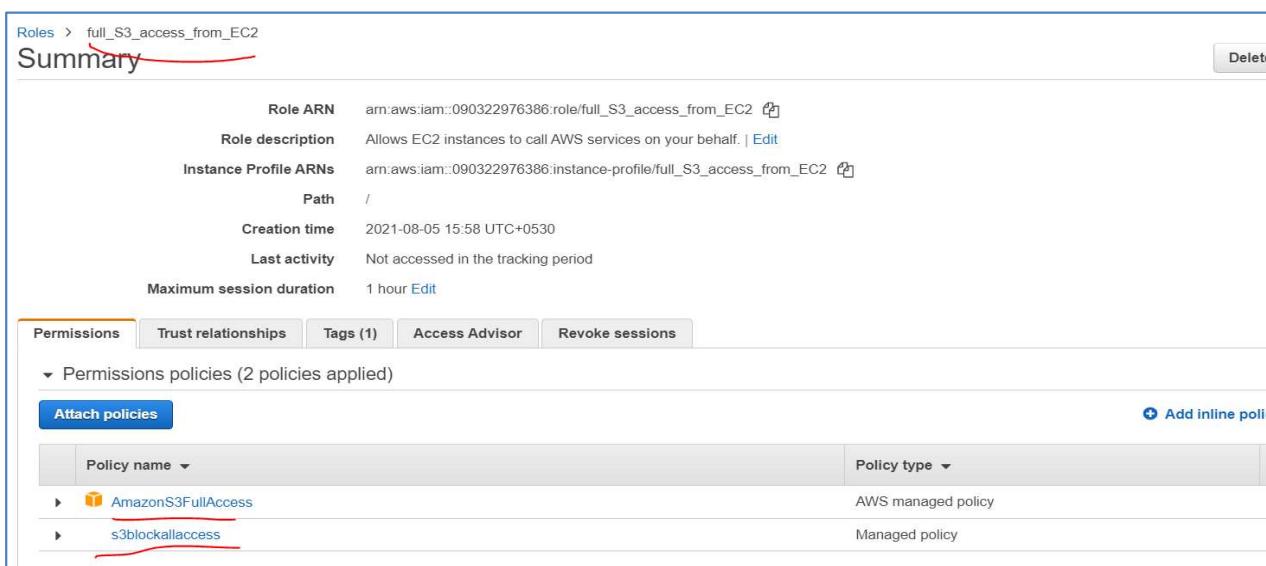
The screenshot shows the AWS IAM Policies list. A policy named "s3blockallaccess" is selected. In the top right corner, a context menu is open over the selected row, with "Attach" highlighted. Other options in the menu include "Detach" and "Delete".

Step 9. You will see all users, groups and roles to attach this policy with. Pick roles:

'full_S3_access_from_EC2' to attach this policy which is already applied to an EC2 instance.



The screenshot shows the AWS IAM Roles list. A role named "full_S3_access_from_EC2" is selected. In the top left corner, there is a search bar and a filter dropdown set to "Filter: Name". Below the list, there is a table with columns for "Name", "Type", and "Used as".



The screenshot shows the AWS IAM Role summary page for "full_S3_access_from_EC2". The "Permissions" tab is selected, showing two attached policies: "AmazonS3FullAccess" and "s3blockallaccess". The "Policy type" column indicates that "AmazonS3FullAccess" is an "AWS managed policy" and "s3blockallaccess" is a "Managed policy".

Step 10. Login to the same EC2 instance where this role was attached and check s3 bucket list again:

```
aws s3 ls
```

```
[ec2-user@ip-172-31-7-68 ~]$ aws s3 ls
```

```
An error occurred (AccessDenied) when calling the ListBuckets operation: Access Denied
[ec2-user@ip-172-31-7-68 ~]$
```

The new custom policy is blocking the access now.