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

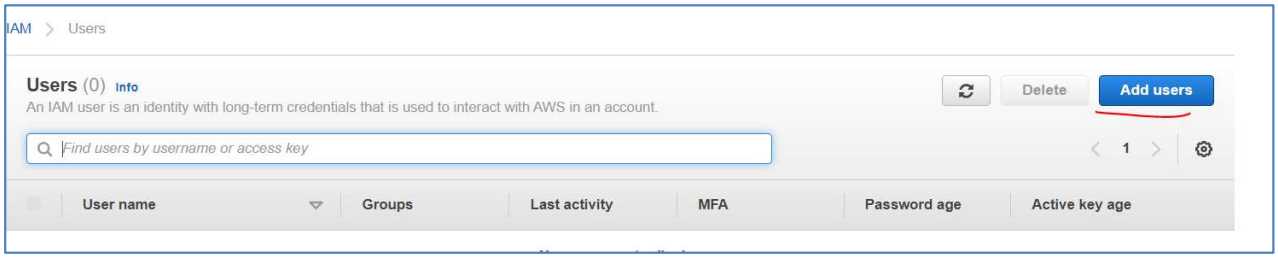
WhatsApp us @ +91 98712 72900

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**.



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.

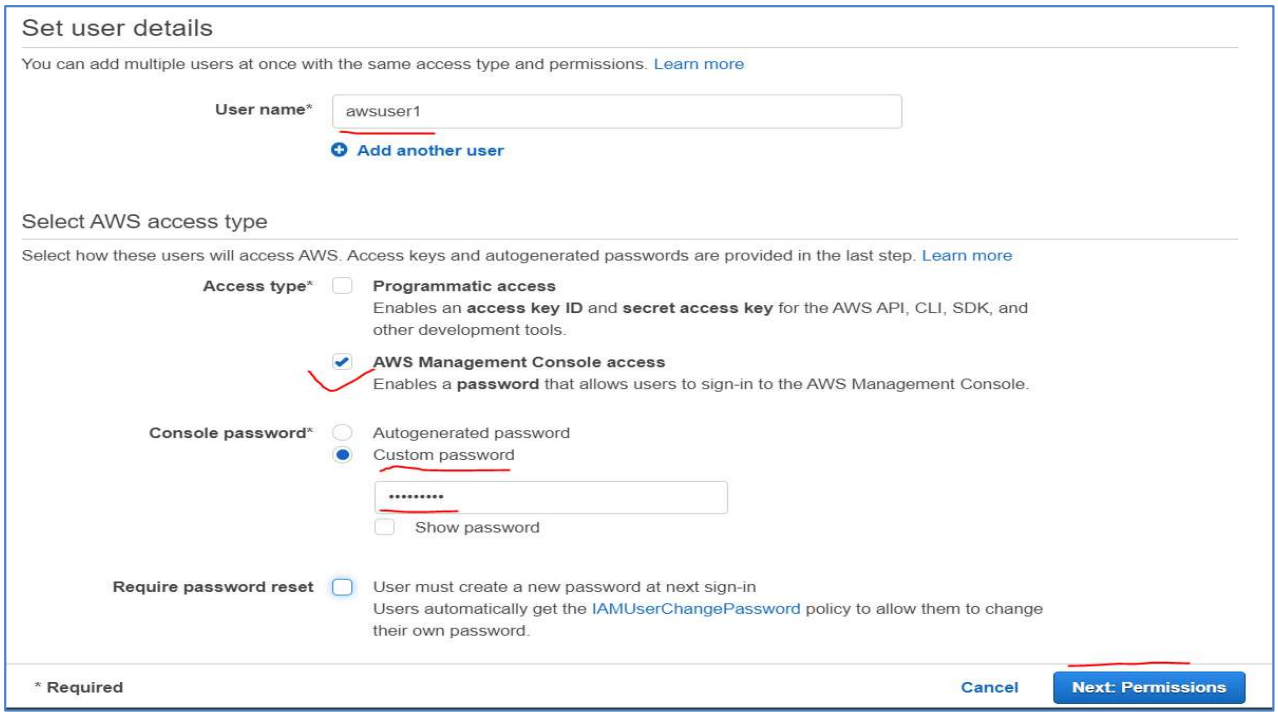
- Select **Programmatic access** if the users require access to the API, AWS CLI, or Tools for Windows PowerShell. This creates an access key for each new user. You can view or download the access keys when you get to the **Final** page.
- Select **AWS Management Console access** if the users require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

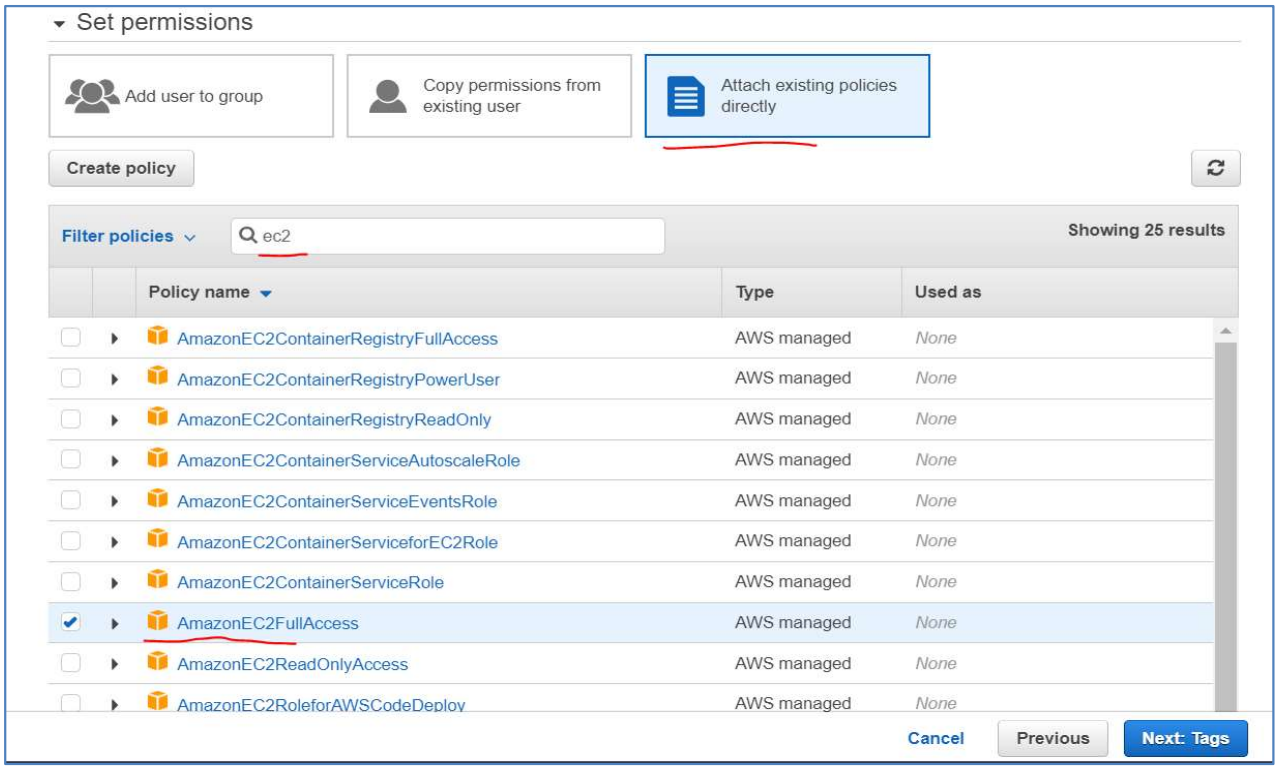


Step 5. Choose **Next: Permissions**.

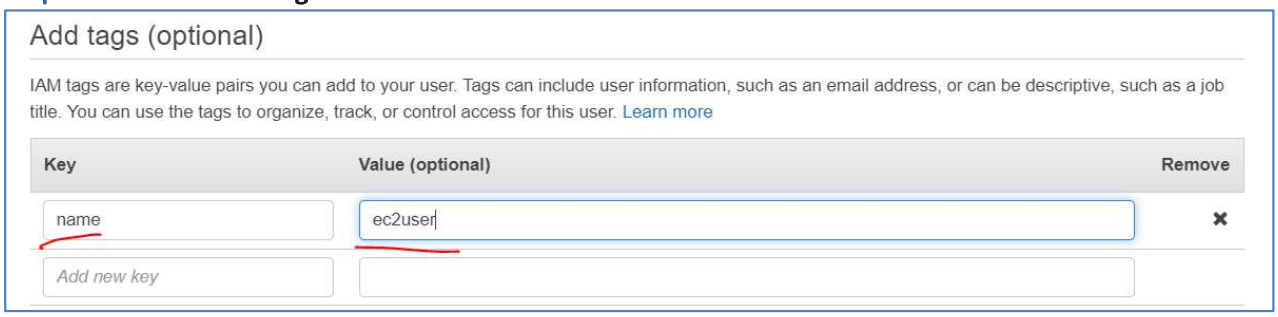
Step 6. On the **Set permissions** page, specify how you want to assign permissions to this set of new users.

Step 7. Choose one of the following three options:

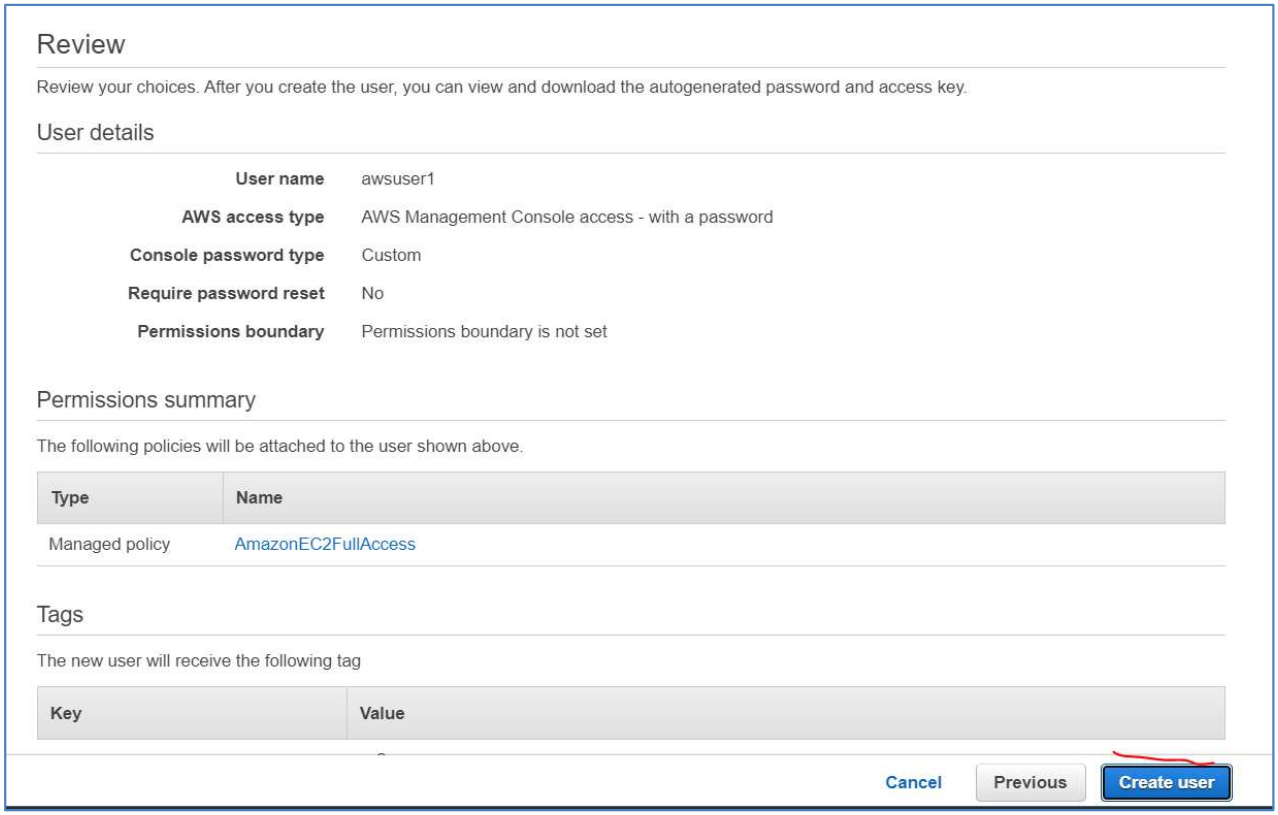
- **Add user to group**. Choose this option if you want to assign the users to one or more groups that already have permissions policies. IAM displays a list of the groups in your account, along with their attached policies. You can select one or more existing groups, or choose **Create group** to create 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 .



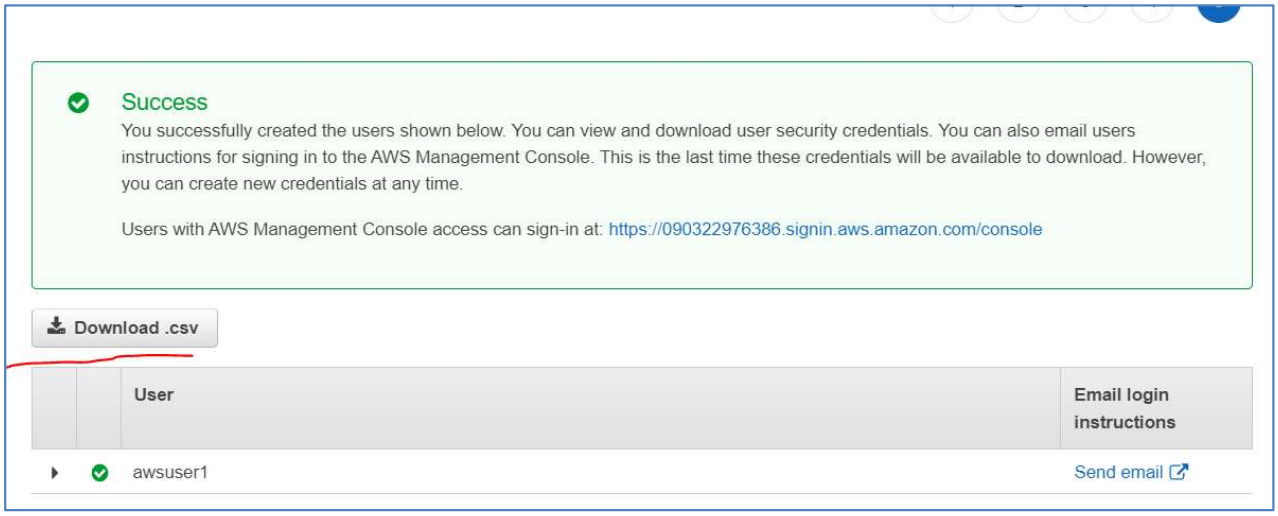
Step 8. Choose **Next: Tags**.



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**.



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.



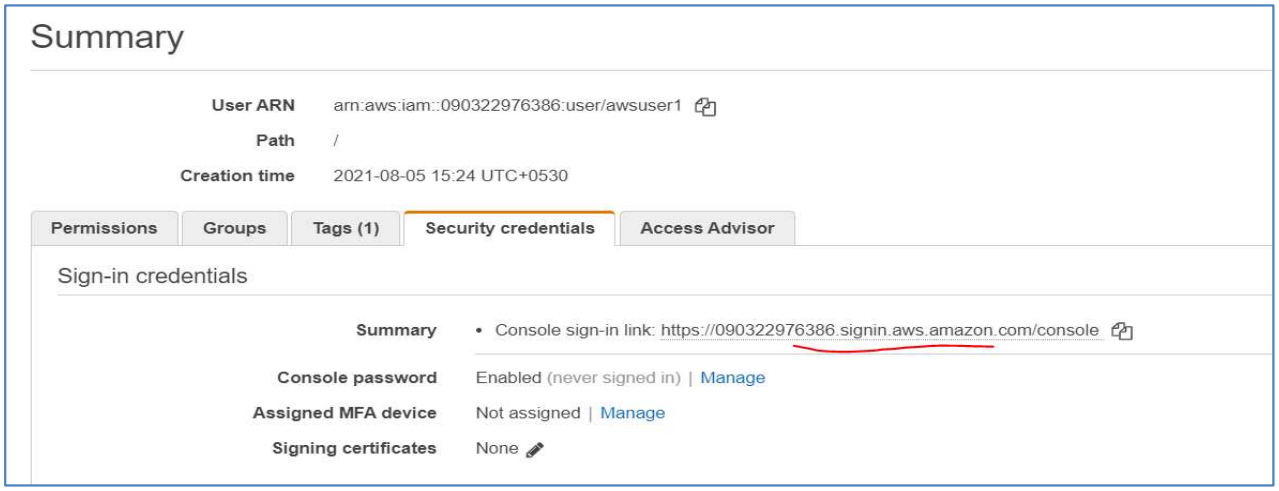
Important: This is your only opportunity to view or download the secret access keys, and you must provide this information to your users before they can use the AWS API. Save the user's new access key ID and secret access key in a safe and secure place. **You will not have access to the secret keys again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

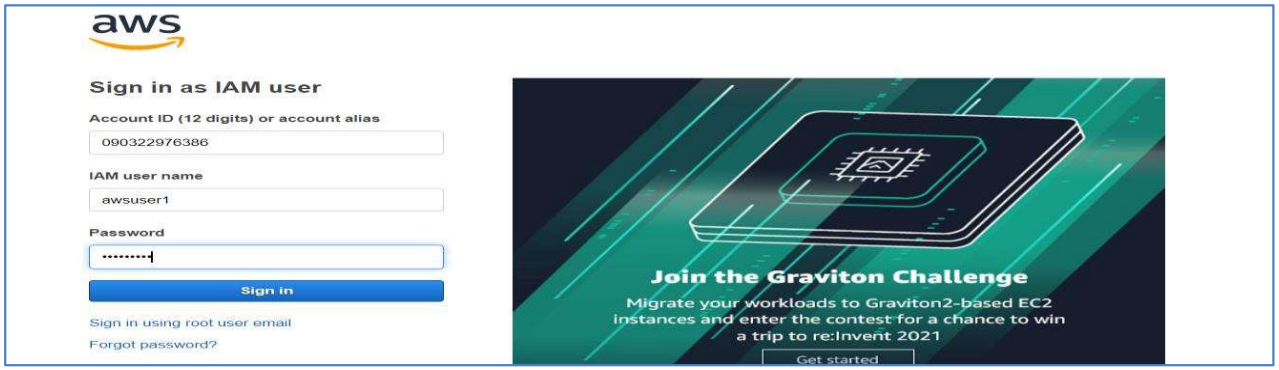
Step 11. (Optional) Provide each user with his or her credentials. On the final page you can choose **Send email** next to each user. Your local mail client opens with a draft that you can customize and send. The email template includes the following details to each user:

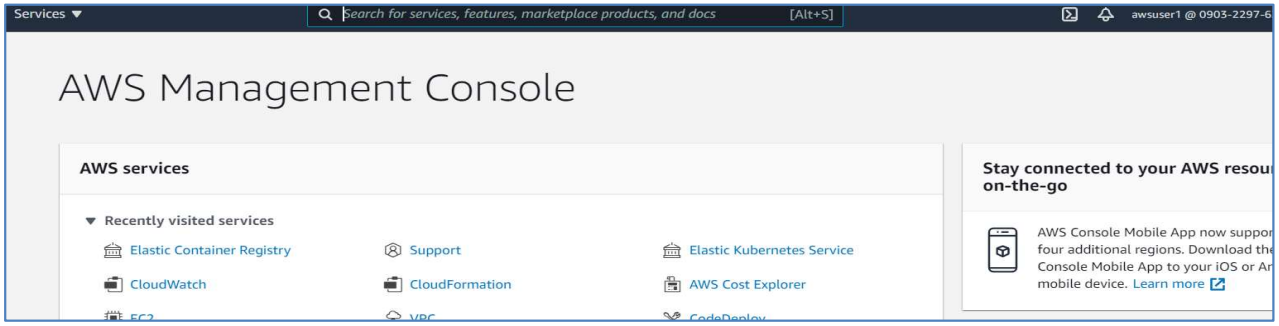
- User name
- URL to the account sign-in page. Use the following example, substituting the correct account ID number or account alias:

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

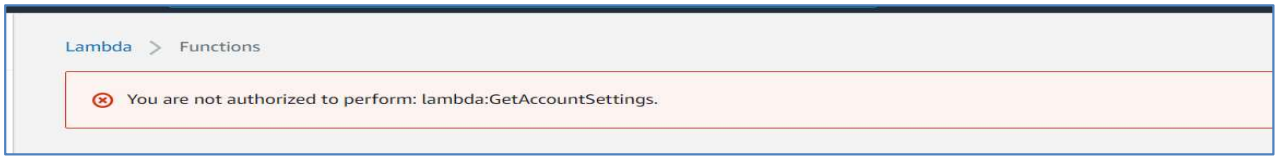
- User can get the login URL from credentials tab that you emailed him from user summary and login to it:







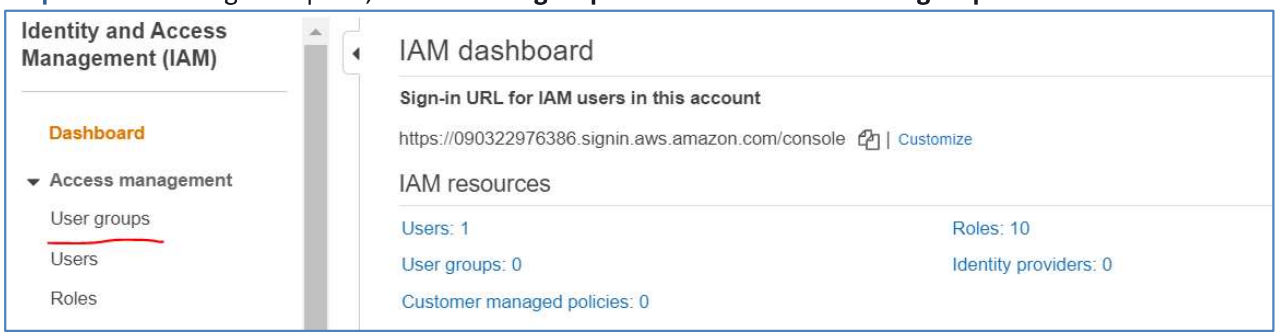
This user can access only EC2 services. If he tries to access any other, for e.g. Lambda, then below error will occur

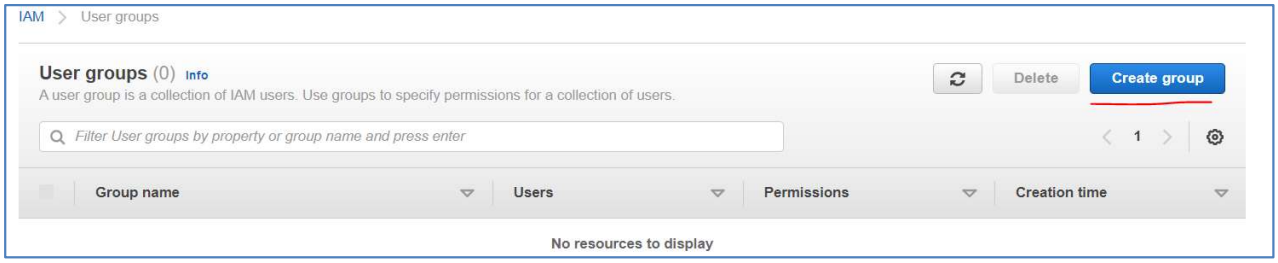


To create an IAM user group and attach policies (console)

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**.

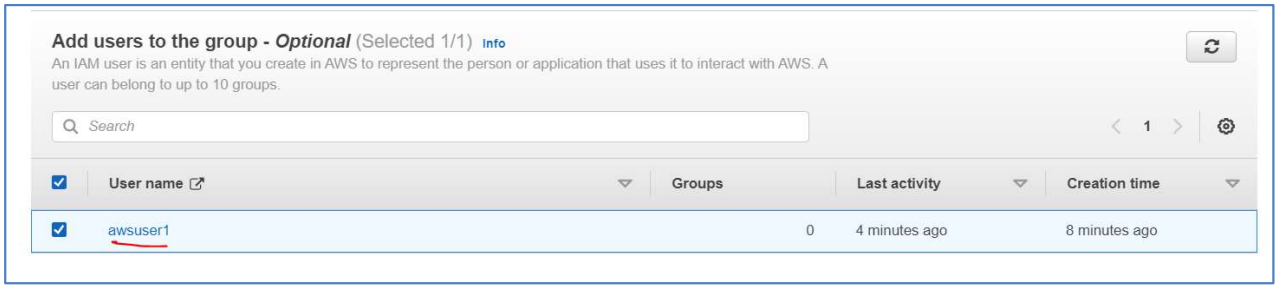




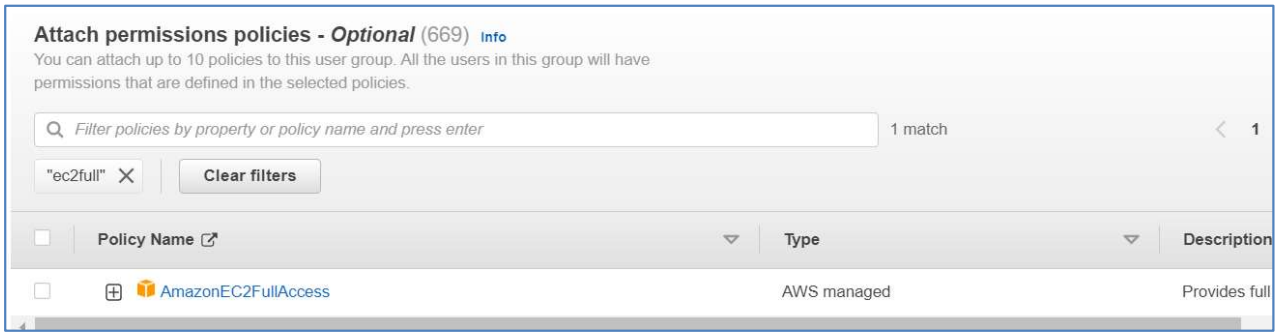
Step 3. For **User group name**, type the name of the group, say **awsgrp1**.

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**.

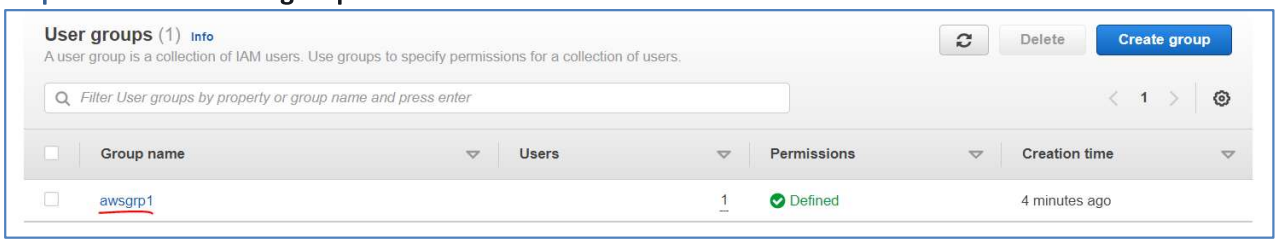
Step 4. In the list of users, select the check box for each user that you want to add to the group.



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.



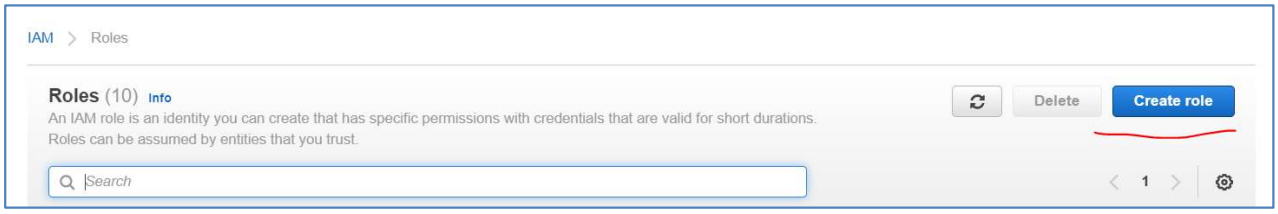
Step 6. Choose **Create group**.



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**.



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

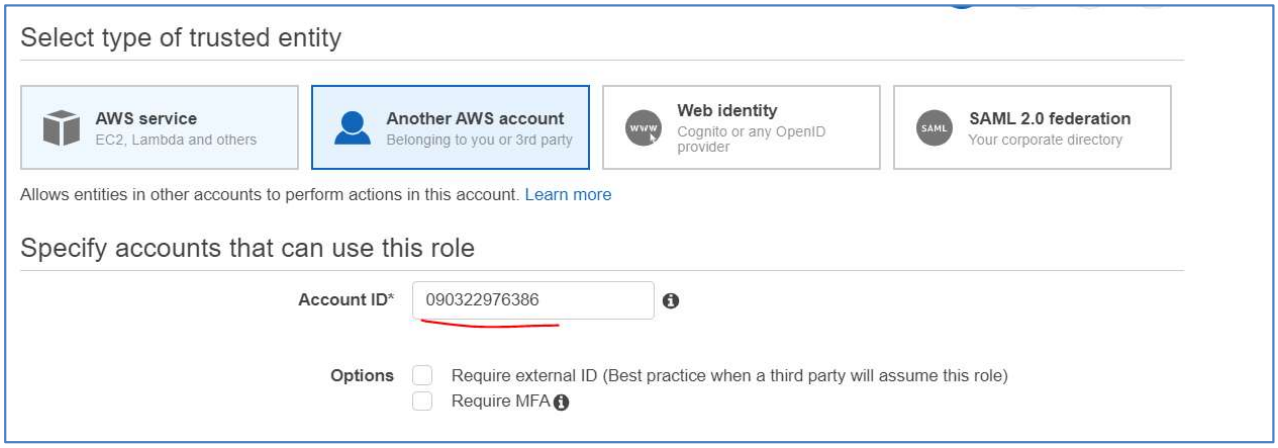
Step 4. For **Account ID**, type the AWS account ID to which you want to grant access to your resources. Use the account ID of AWS user we created earlier, for e.g. **090322976386**

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.

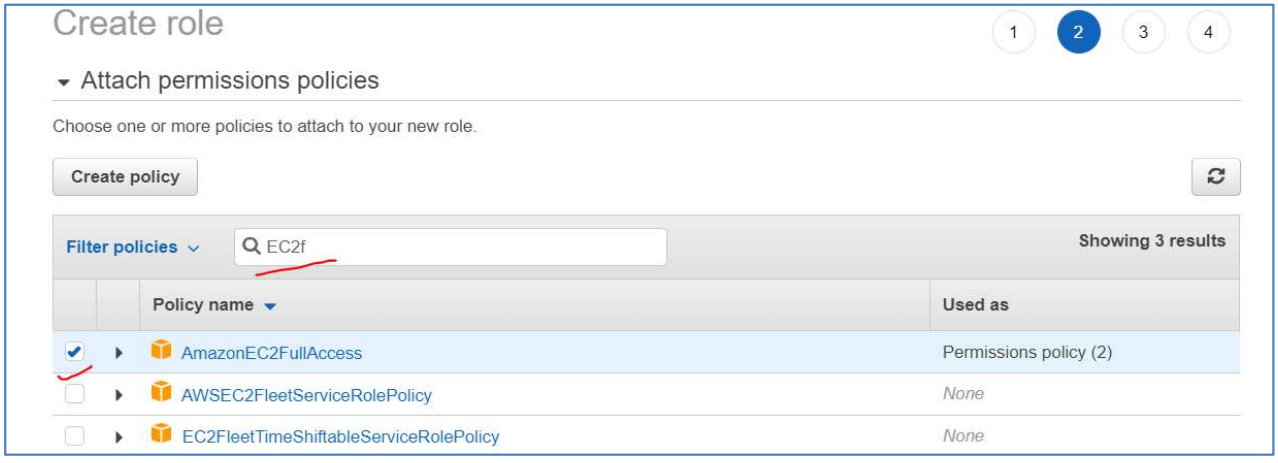
Step 6. If you want to restrict the role to users who sign in with multi-factor authentication (MFA), select **Require MFA**. This adds a condition to the role's trust policy that checks for an MFA sign-in. A user who wants to assume the role must sign in with a temporary one-time password from a configured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



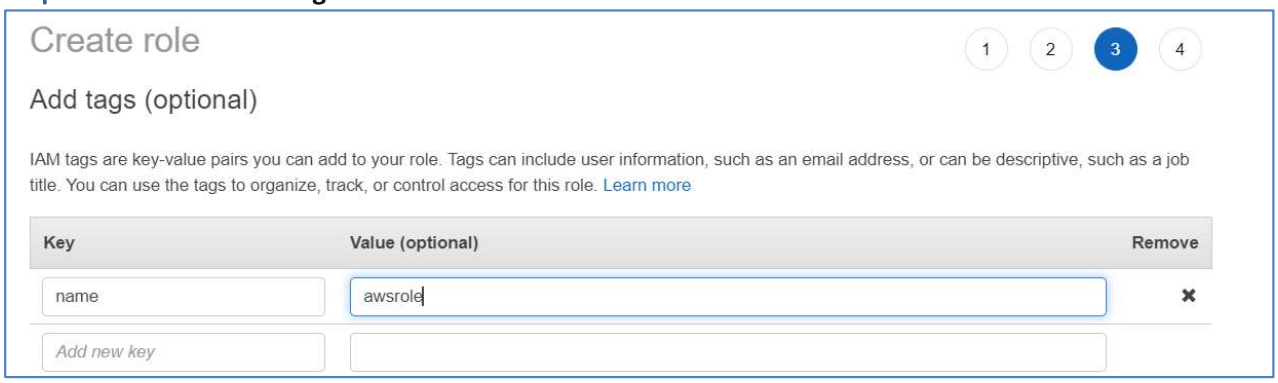
Step 7. Choose **Next: Permissions**.

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.



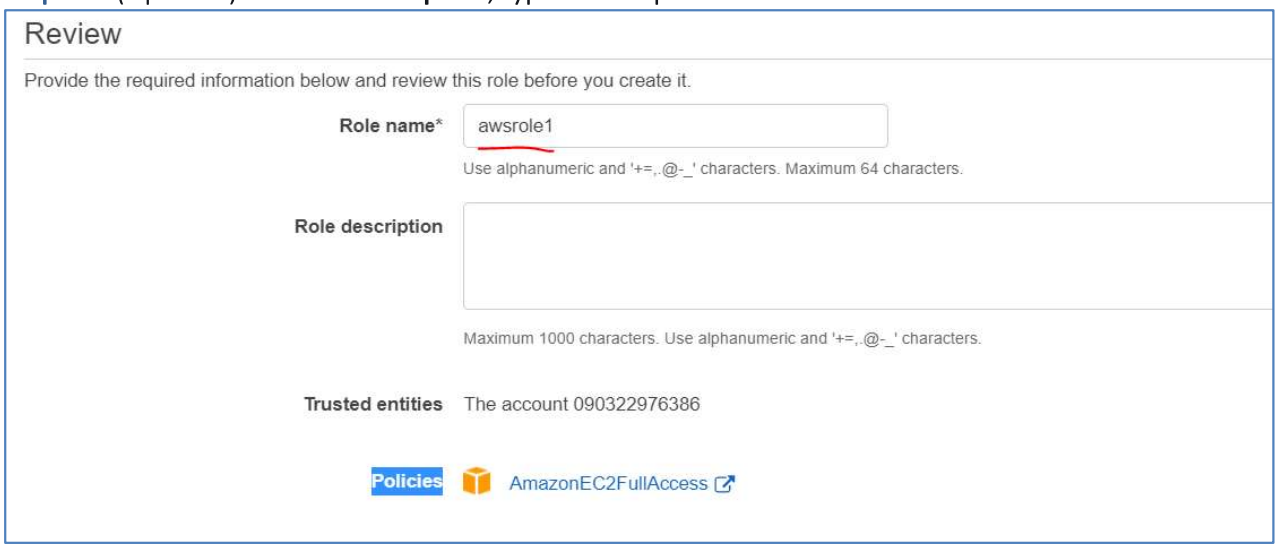
Step 10. Choose **Next: Tags**.



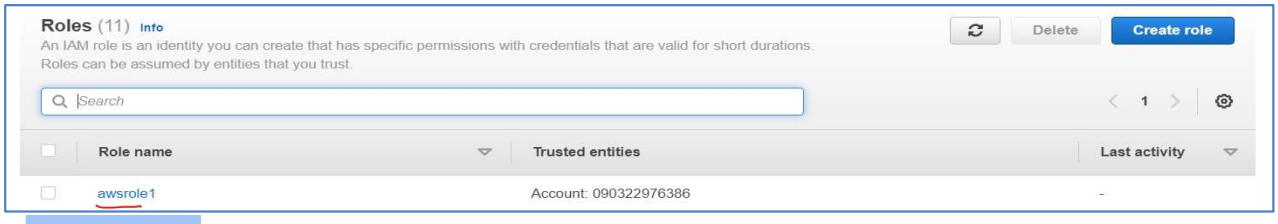
Step 11. Choose **Next: Review**.

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.



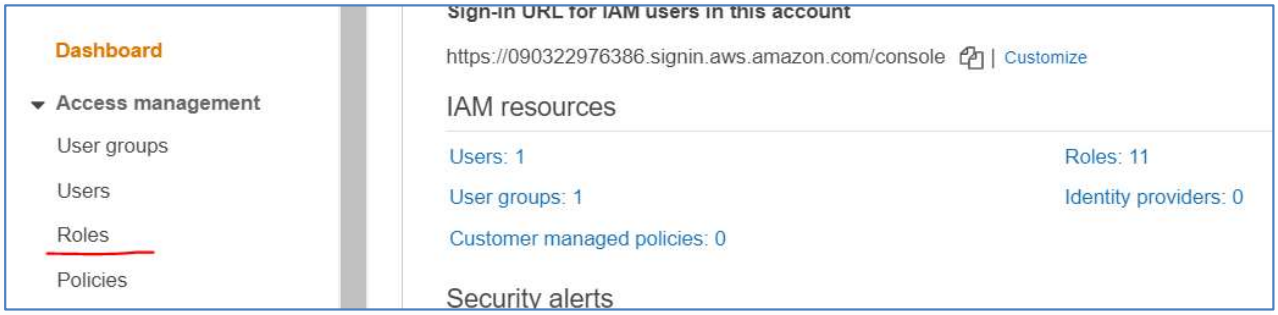
Step 14. Review the role and then choose **Create role**. Once created, it can be seen under roles list:



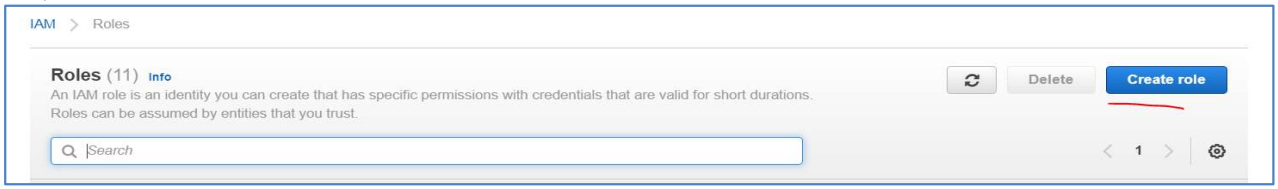
Enable S3 access from EC2 by IAM role

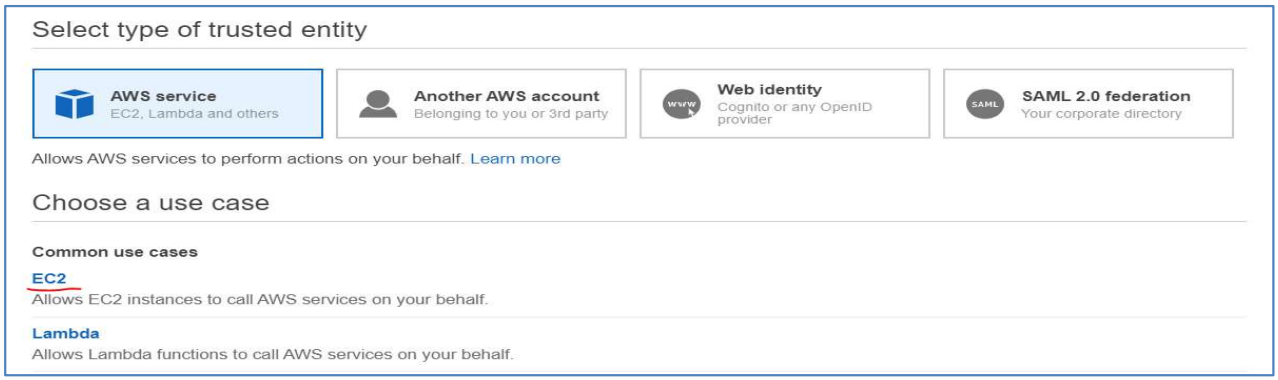
Create a new IAM role

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

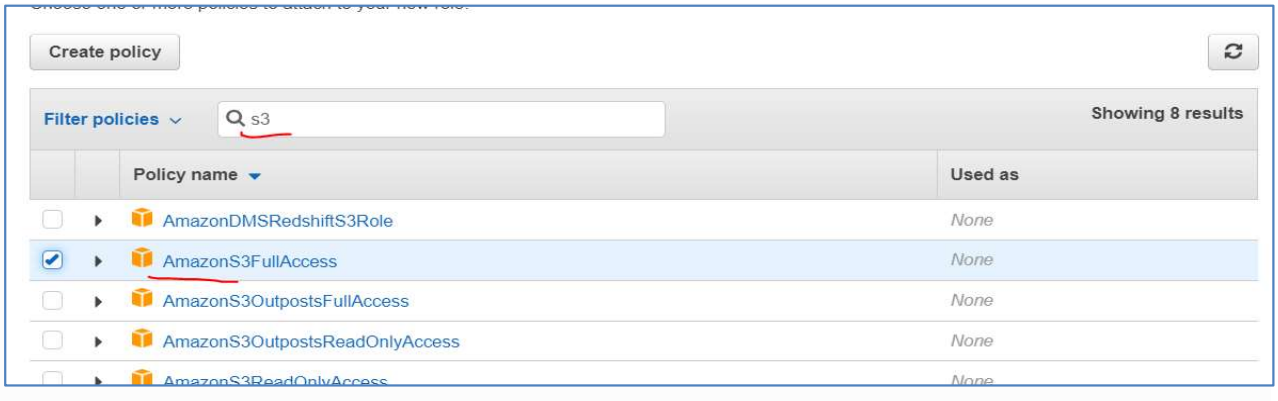


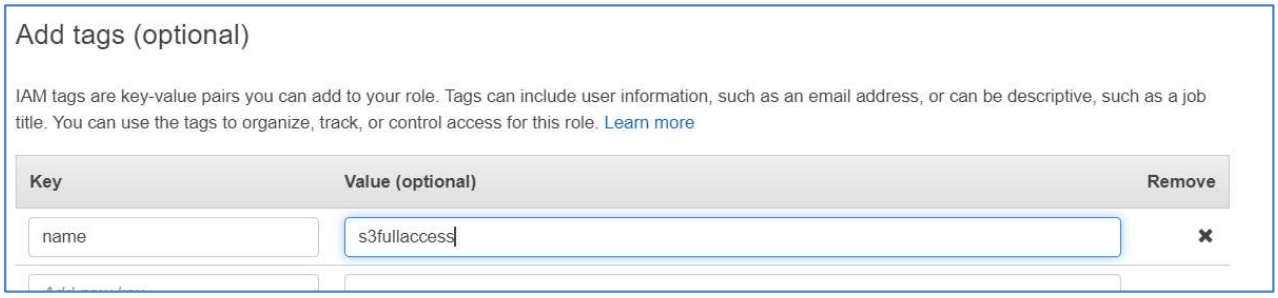
Step 2. Then choose “Roles” in the IAM console and click on “Create role”:



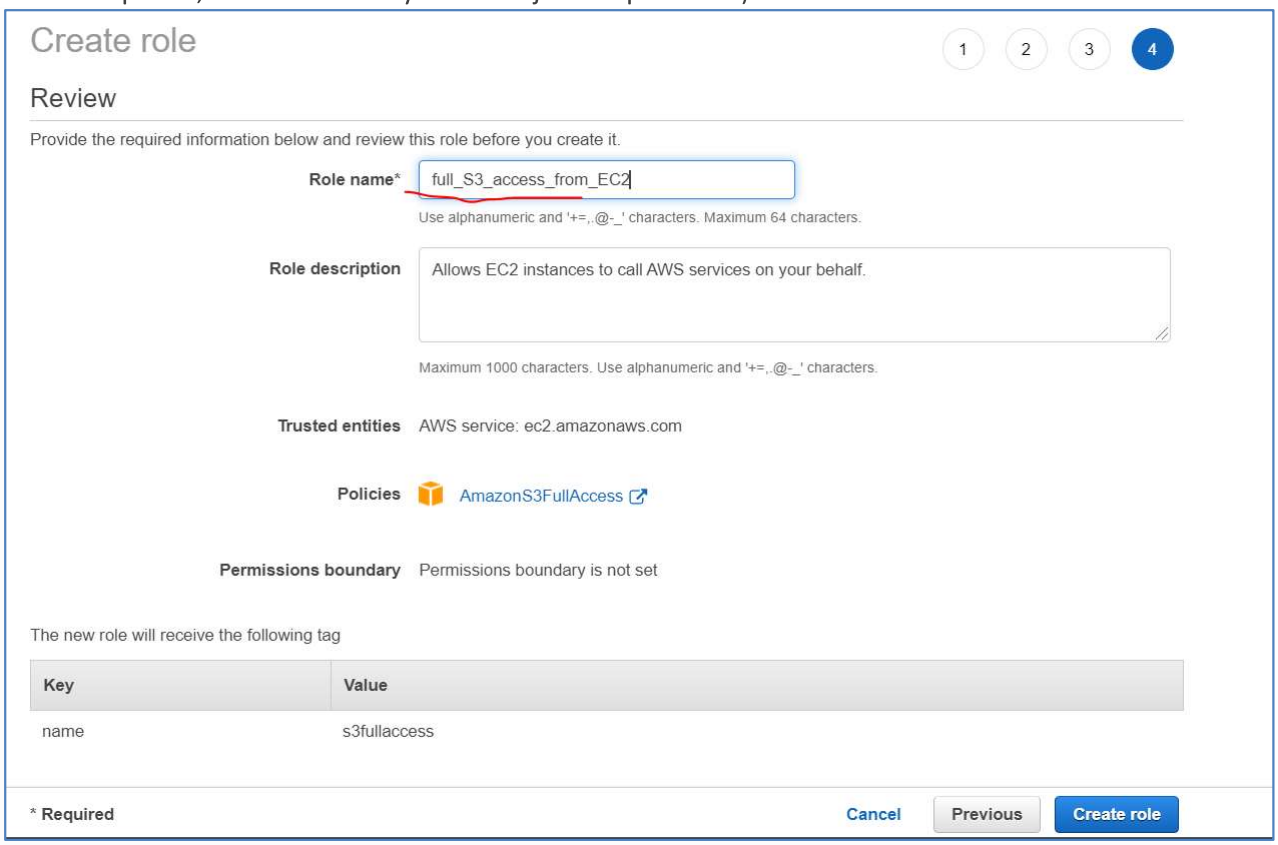


Step 3. Search for “S3” and then select “AmazonS3FullAccess”:





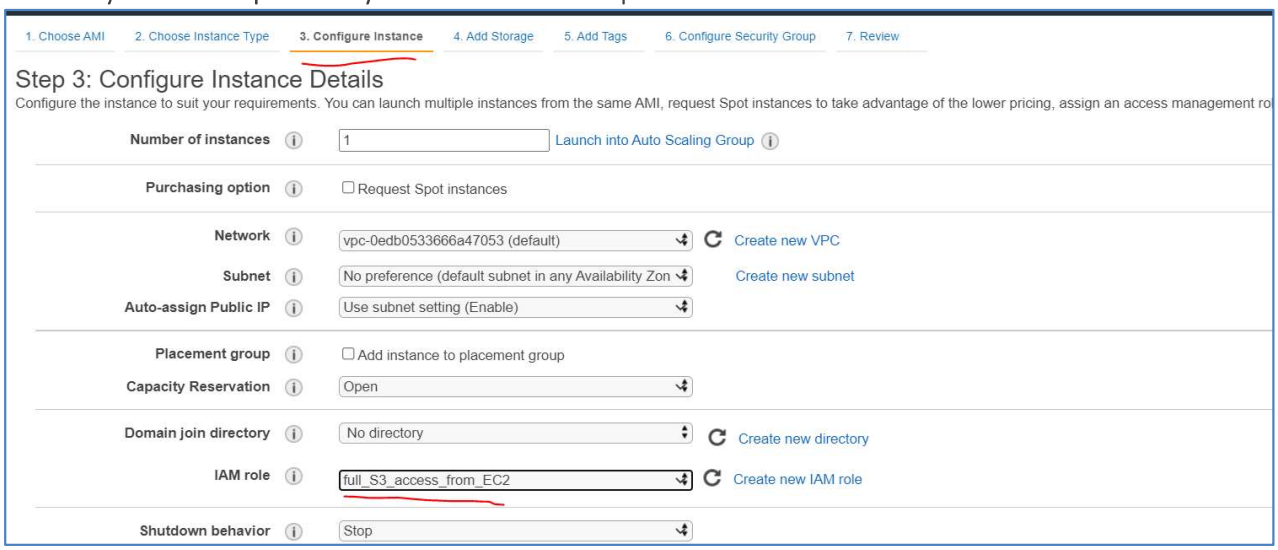
Step 4. Finally, give this role a descriptive name. Here I use "full_S3_access_from_EC2". (For the "Role description", enter whatever you like or just keep default.)



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.



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
```

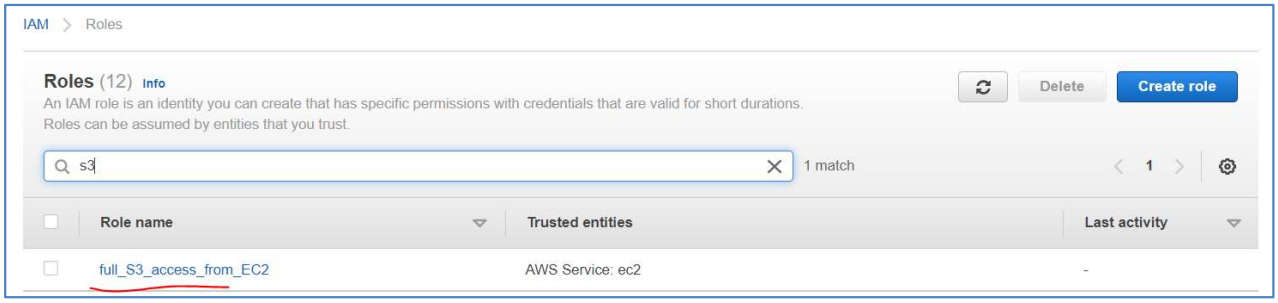
```
aws s3 ls
```

```
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+Sxn1sht77FmSwAGV36WmcnzE.
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.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-7-68 ~]$ aws s3 ls
2021-08-04 06:47:28 bucket0821
[ec2-user@ip-172-31-7-68 ~]$
```

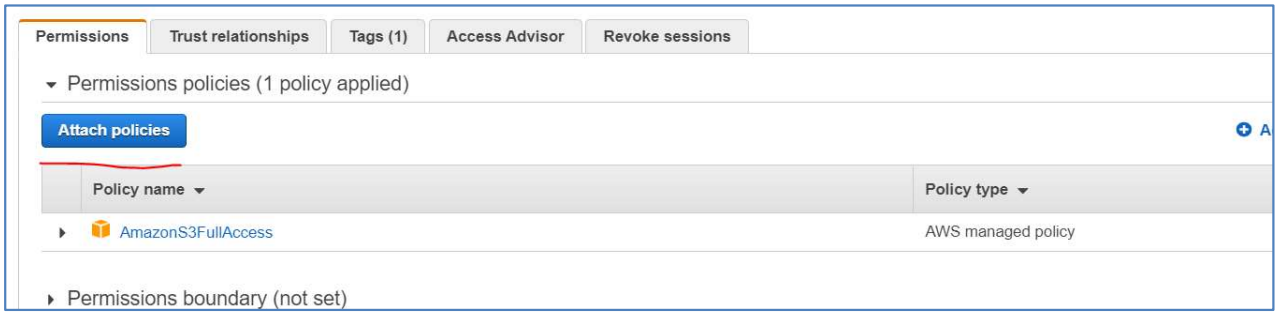
Create a JSON document using access policy

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:**

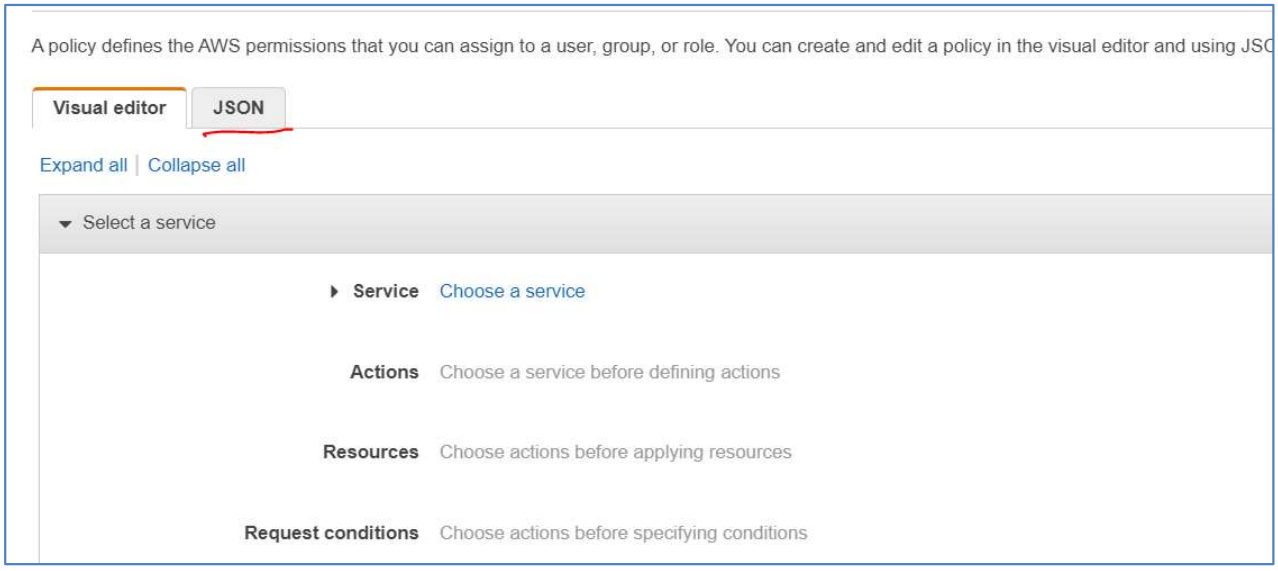


Step 2. Click on the role **full_S3_access_from_EC2** and select Attach policies:



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:**

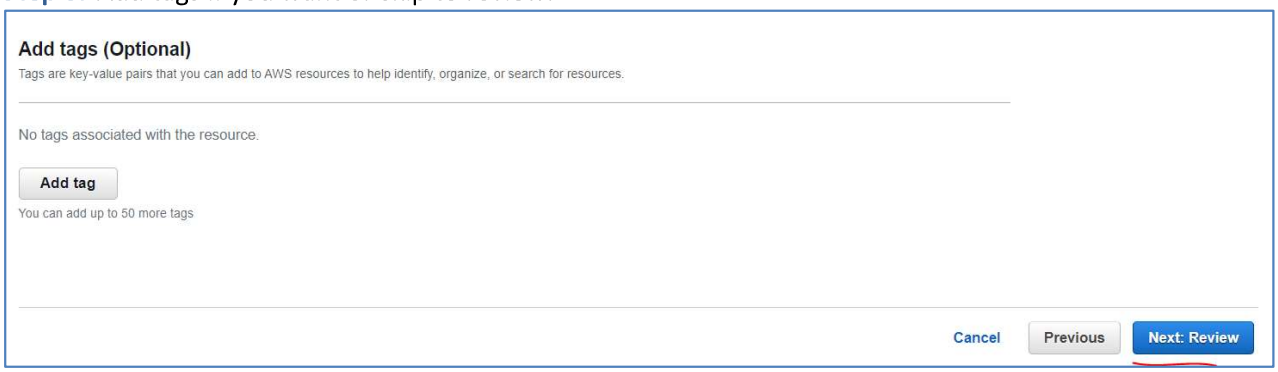
Step 4. You see a policy editor with two options. In visual editor, you can pick every component one by one and derive the policy, while in JSON format you can write your policy in once go. That's what we are going to do here:



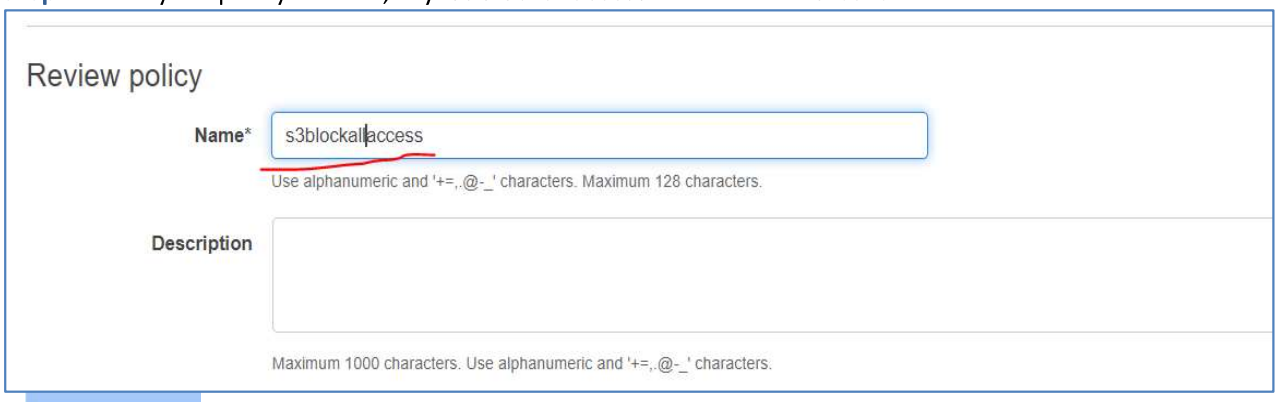
Step 5. Copy the below content and paste to JSON tab and validate its do not have any error. Go to add tags then:

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

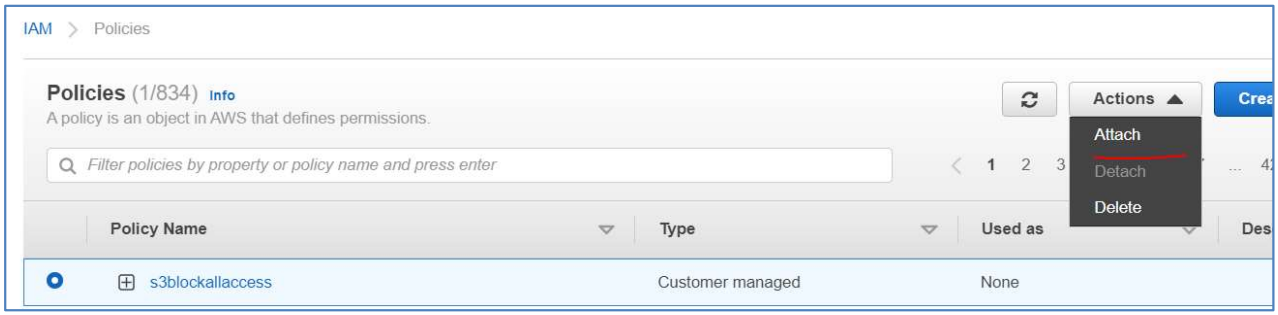
Step 6. Add tags if you want or skip to review:



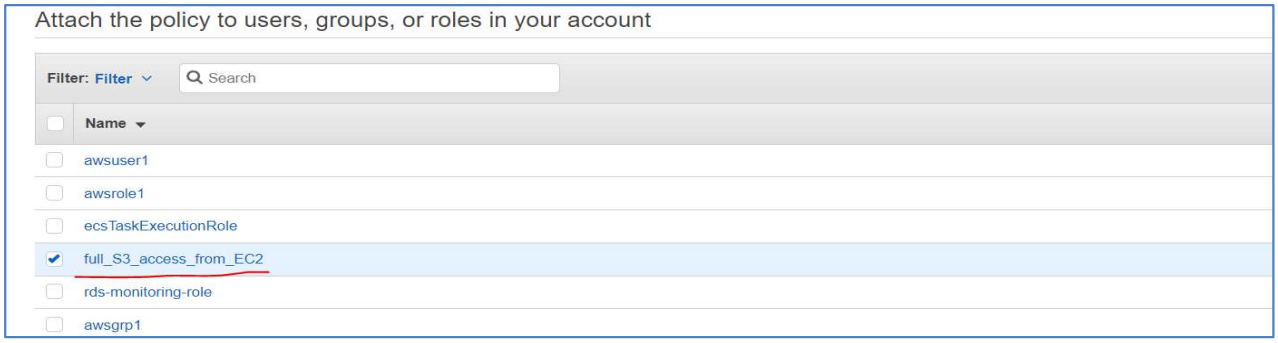
Step 7. Give your policy a name, say **'s3blockallaccess'** and click on **Create**.

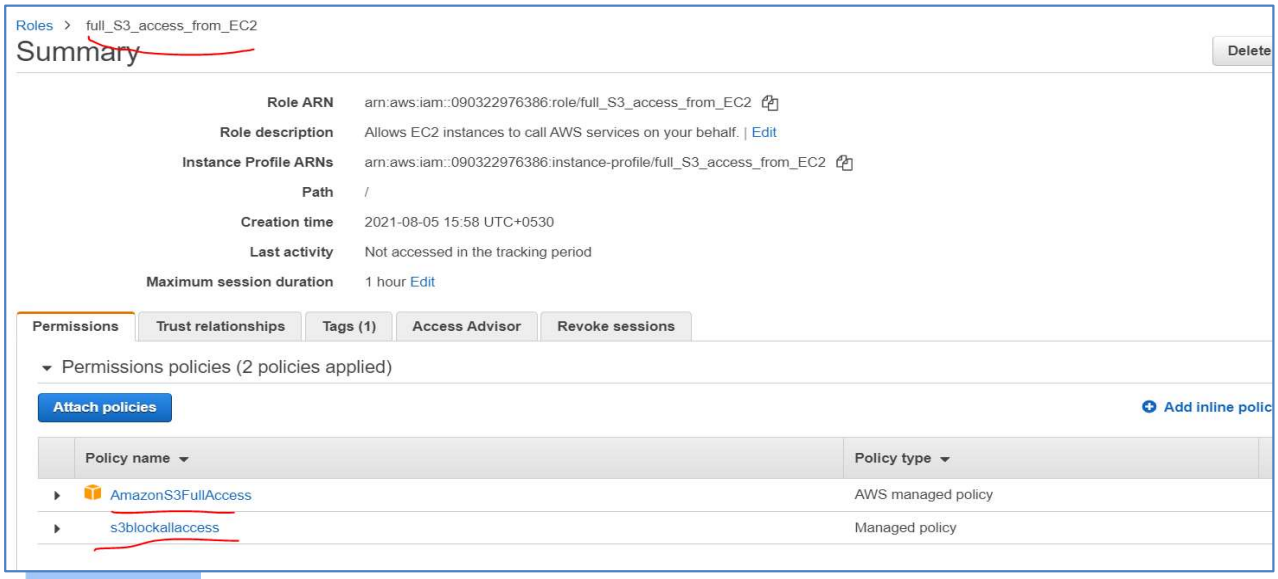


Step 8. Once your policy is created, it will be listed under Policies tab. Select it and select **Actions -> Attach:**



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.





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.