AWS IAM:

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. With IAM, you can centrally manage permissions that control which AWS resources users can access. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

Access management:

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

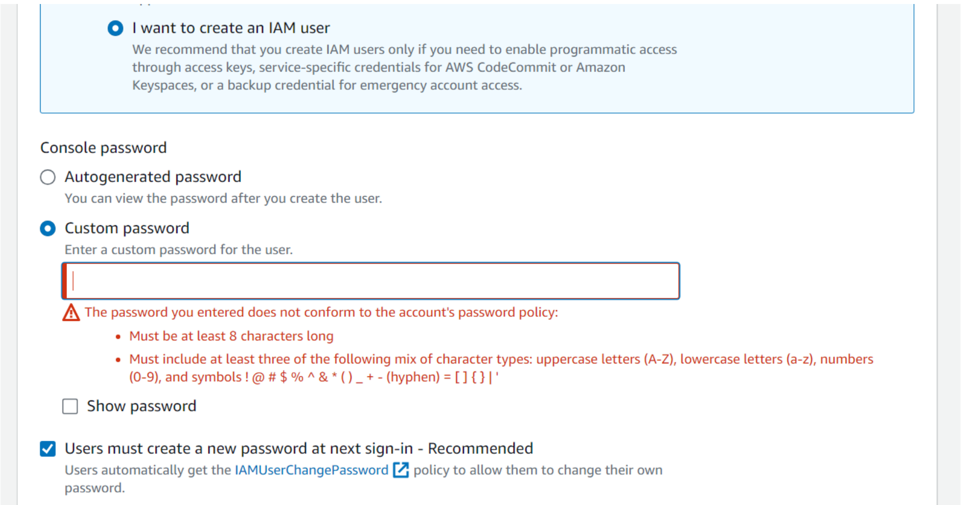
Users:

An IAM *users* is an identity within your AWS account that has specific permissions for a single person or application. Where possible, best practices recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys.

1. On the **Console Home** page, select the IAM service.
2. select **Users** and then select **Add users**.
3. **User details**, in Username, enter the name for the new user and next.
4. On the **Set permissions** select which option we want and next.

5. if we want to provide console access to a person.

6. select Provide user access to the AWS Management Console.



● I want to create an IAM user

We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

Console password

○ Autogenerated password

You can view the password after you create the user.

● Custom password

Enter a custom password for the user.

⚠ The password you entered does not conform to the account's password policy:

- Must be at least 8 characters long
- 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) = [ ] { } | '

□ Show password

☑ Users must create a new password at next sign-in - Recommended

Users automatically get the IAMUserChangePassword policy to allow them to change their own password.

7. select I want to create an IAM user and select **Custom password.**

8. select user must create a new password at next sign-in-Recommended. (User will create new password after first login)

9. next

User groups:

An IAM user group is a collection of IAM users. User groups let you specify permissions for multiple users, which can make it easier to manage the permissions for those users.

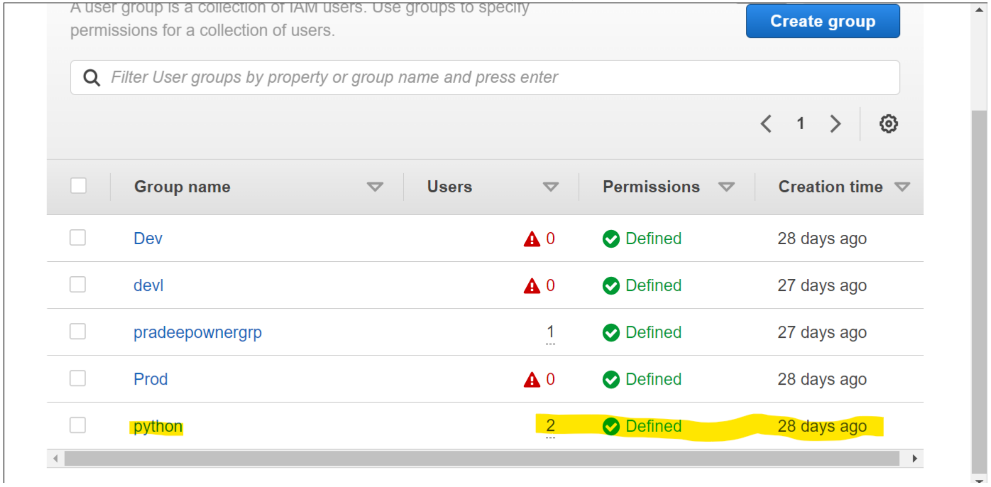
1. Sign into the AWS Management Console and open the IAM console
2. In the navigation pane, choose **User groups** and then choose **Create group**.
3. For **User group name**, type the name of the group.

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

6. Choose **Create group**.

Example: 2 users need to access only ec2 instances of python-servers

Group name: Python
Users: user1, user2
Policy: EC2_restrict_pythonservers (created custom policy)



Create group
Filter User groups by property or group name and press enter
1

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
□	Dev	⚠ 0	✓ Defined	28 days ago
□	devl	⚠ 0	✓ Defined	27 days ago
□	pradeepownergrp	1	✓ Defined	27 days ago
□	Prod	⚠ 0	✓ Defined	28 days ago
□	python	2	✓ Defined	28 days ago

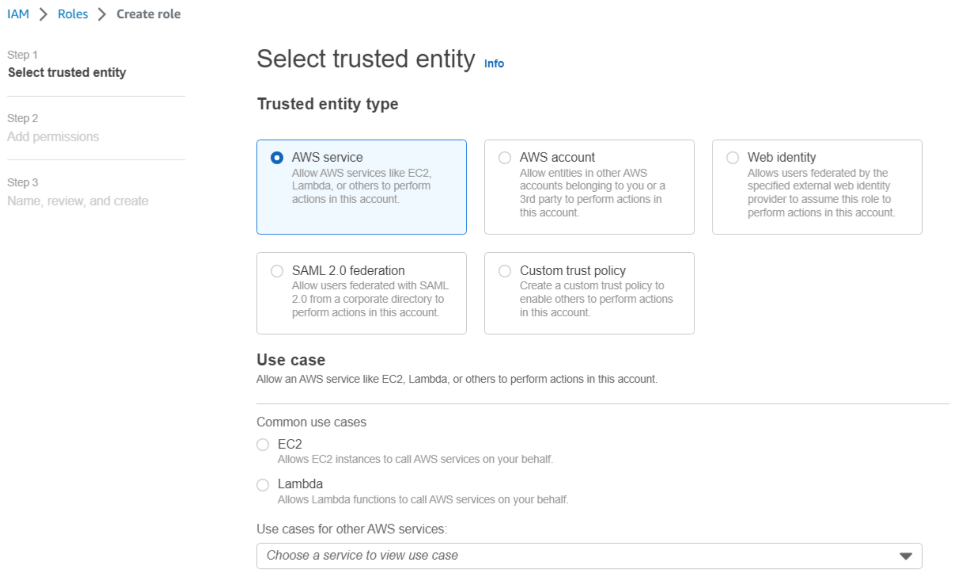
Roles: For one service giving access of another service we will create roles.

If we go with roles no need to create access keys and secrete key, you can use roles to delegate access to users, applications, or services that don't normally have access to your AWS resources

Example:
For 1 ec2 instance giving access of S3 directly.

1. Sign into the AWS Management Console and open the IAM console
2. In the navigation pane of the console, choose **Roles** and then choose **Create role**.
3. Choose **AWS service** role type.

4. Choose in **use case** and **common use case** select **EC2** and next.



IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity Info

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

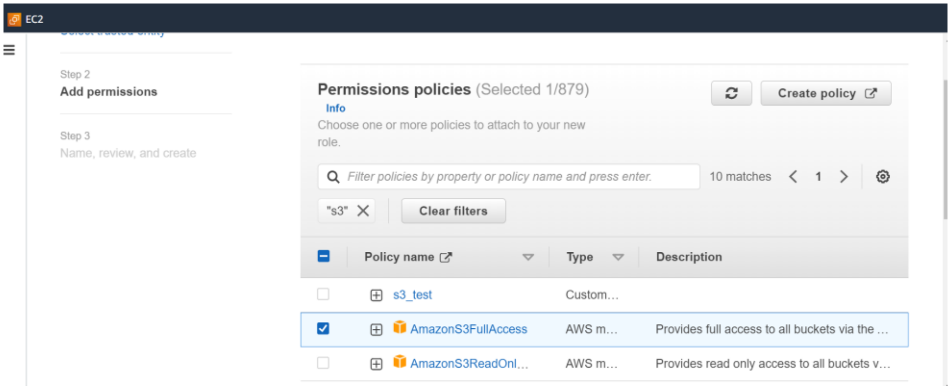
○ EC2
Allows EC2 instances to call AWS services on your behalf.

○ Lambda
Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

Choose a service to view use case

5. Select which police permission you want to give.



EC2

Step 2
Add permissions

Step 3
Name, review, and create

Permissions policies (Selected 1/879) Info
Choose one or more policies to attach to your new role.

Create policy

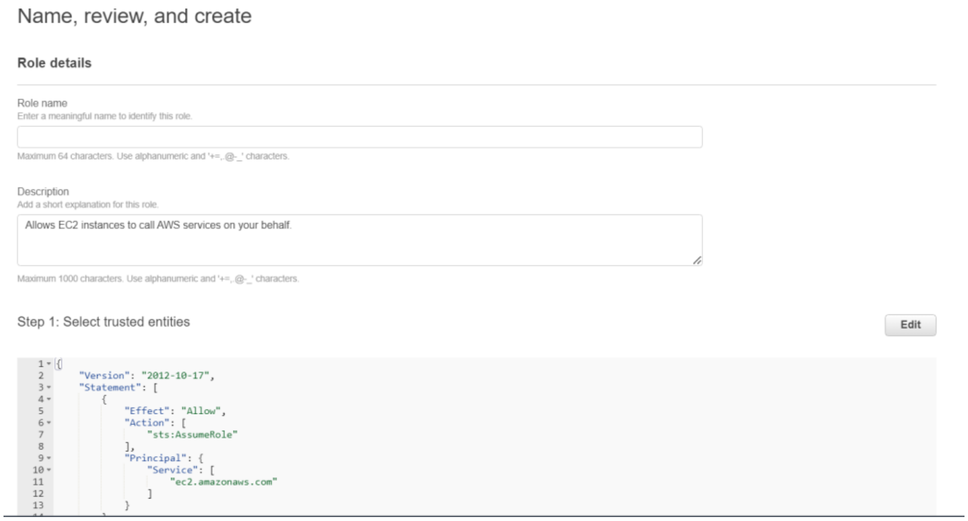
Filter policies by property or policy name and press enter.

10 matches < 1 >

"s3" ✕ Clear filters

	Policy name	Type	Description
□	⊞ s3_test	Custom…	
☑	⊞ AmazonS3FullAccess	AWS m…	Provides full access to all buckets via the …
□	⊞ AmazonS3ReadOnl…	AWS m…	Provides read only access to all buckets v…

6. In role details fill **role name** and **description** (optional).



Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

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

Description
Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

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

Step 1: Select trusted entities

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Principal": {
                "Service": [
                    "ec2.amazonaws.com"
                ]
            }
```

7. Create Role

8. Go to Ec2 Select instance and right click to select **Security > Modify IAM role** and there add the S3 access role.

Policies:

To Creating polices we can give permission for users which permission we want to give

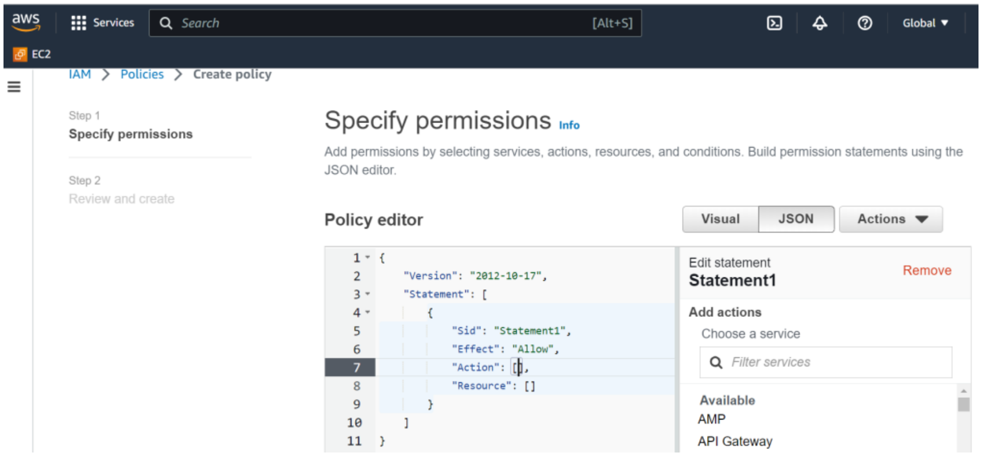
1. Sign into the AWS Management Console and open the IAM console.
2. In the navigation pane of the console, choose policies and then choose **Create policy.**
3. Select JSON and edit the policies or we can select Visual and create policy by selecting required permissions.
4. Resolve any security warnings, errors, or general warnings generated during policy validation, and then choose **Next**.
5. On the **Review and create** page, type a **Policy Name** and a **Description** (optional) for the policy that you are creating. Review **Permissions defined in this policy** to see the permissions that are granted by your policy.
6. Choose **Create policy** to save your new policy.

Inline policy:

An inline policy is a policy created for a single IAM identity (a user, group, or role). Inline policies maintain a strict one-to-one relationship between a policy and an identity. They are deleted when you delete the identity.

1. On the **Console Home** page, select the IAM.
2. select **Users** and then under **Permissions policies.**
3. Select **add Permissions** and then select create inline policy.
4. Select JSON and edit the policies
5. Resolve any security warnings, errors, or general warnings generated during policy validation, and then choose **Next**.
6. On the **Review and create** page, type a **Policy Name** and a **Description** (optional) for the policy that you are creating. Review **Permissions defined in this policy** to see the permissions that are granted by your policy.
7. Choose **Create policy** to save your new policy.

Example: Create custom policy (we can give it to all IAM users/inline policy (user policy)



Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

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

Write custom policy code in Json format based on our requirement and give permissions accordingly. And save the Policy by giving name and Descriptions.

Now we can add policy to Users or Groups, and they get access to resources-based on our Custom policy permissions.

Identity providers:

Use an identity provider (IdP) to manage your user identities outside of AWS but grant the user identities permissions to use AWS resources in your account.

Account settings:

Password policy

Info

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

Security Token Service (STS)
Info

STS is used to create and provide trusted users with temporary security credentials that can control access to your AWS resources.

Session Tokens from the STS endpoints
AWS recommends using regional STS endpoints to reduce latency. Session tokens from regional STS endpoints are valid in all AWS Regions. If you use regional STS endpoints, no action is required. Session tokens from the global STS endpoint (https://sts.amazonaws.com) are valid only in AWS Regions that are enabled by default. If you intend to enable a new Region for your account, you can use session tokens from regional STS endpoints or activate the global STS endpoint to issue session tokens that are valid in all AWS Regions.
Global endpoint

Valid only in AWS Regions enabled by default

Regional endpoints

Valid in all AWS regions

Region name	**Endpoint**	**STS status**
Global Endpoint	https://sts.amazonaws.com	Always active
US East (N. Virginia)	https://sts.us-east-1.amazonaws.com	Always active