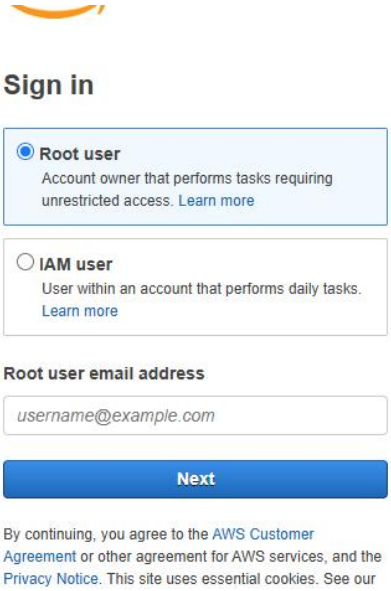
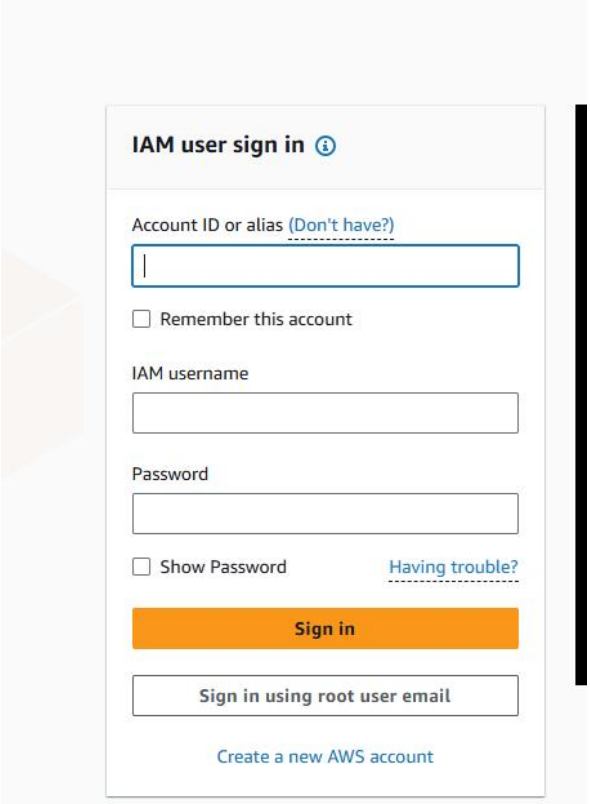
AWS IAM (Identity & Access Management)

We are able to log into AWS cloud and use services -> two types of accounts -->
Root account and IAM account

Root is like the super user and no limitations. It is the most powerful account with no restrictions
If you want to place restrictions on user accounts, then we need IAM user. I can decide which user should have what access



Sign into AWS console, that's what we usually see

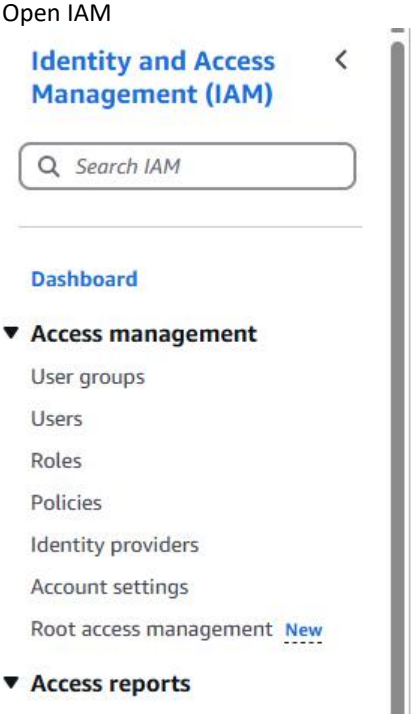


It is used to manage users, groups, policies and roles. IAM is a free service

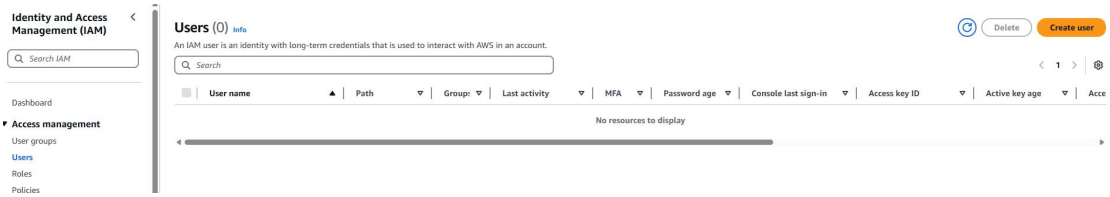
Root account is very powerful with no restrictions and we can access everything in AWS cloud

Key points to be considered for the root account:
We should not use root account for daily task also we should not share root account credentials with anyone (it is also highly-recommended to enable high security by enabling Multi-Factor Authentication MFA for root users)
As a part of a project or team, we will not be getting root account credentials rather we will get IAM account credentials with specific access

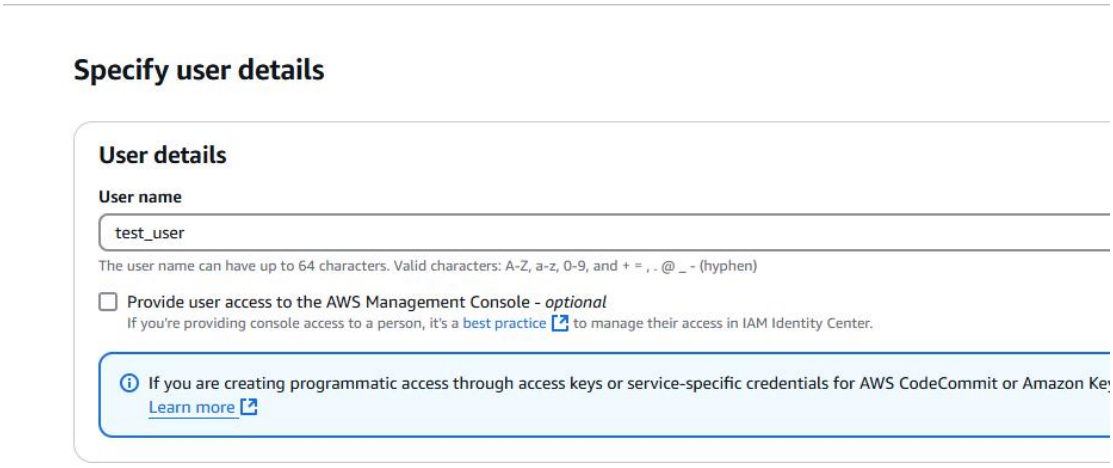
Open IAM



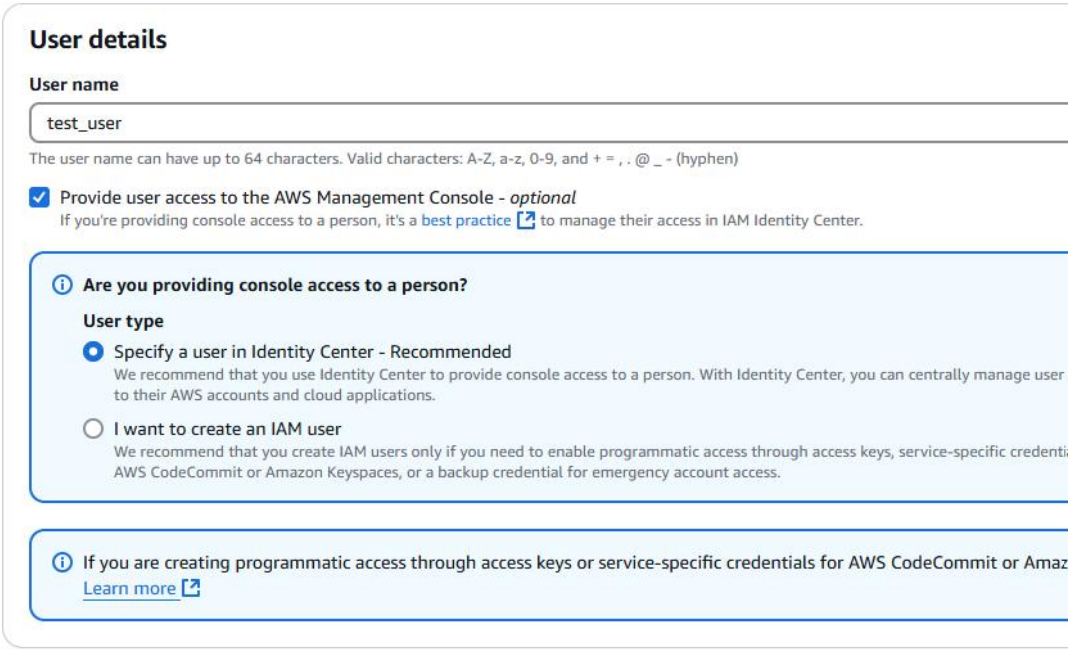
Click Users



Create user

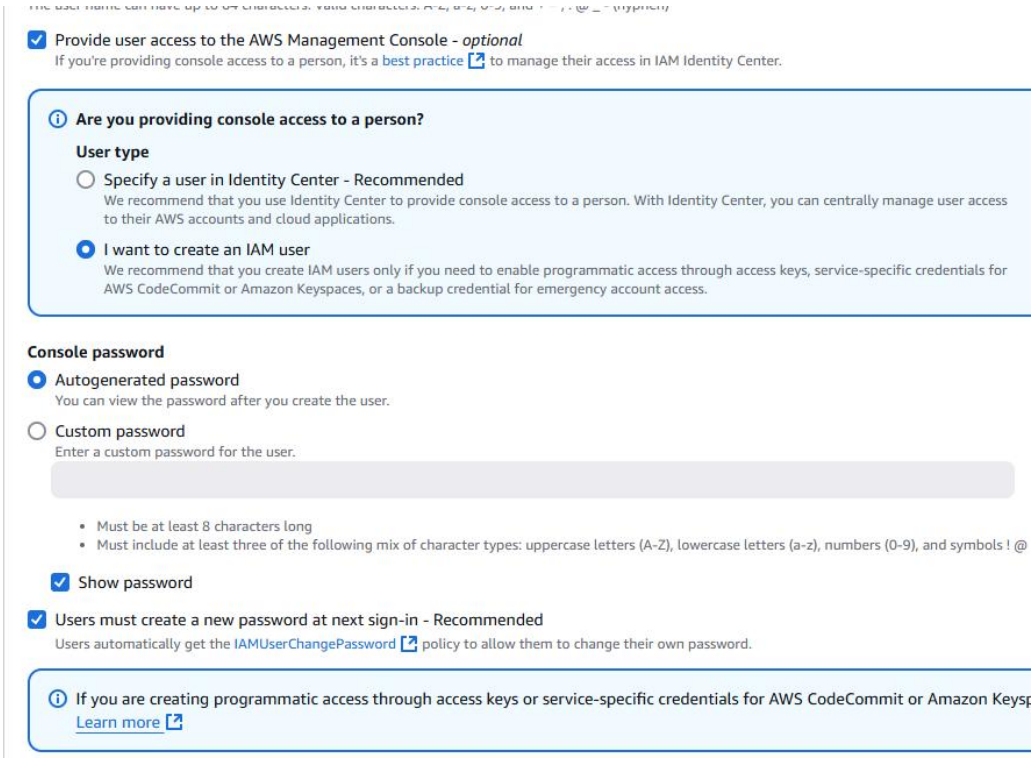


If you don't click 'Provide user access to the AWS management console - optional' then user cannot access AWS console through browser

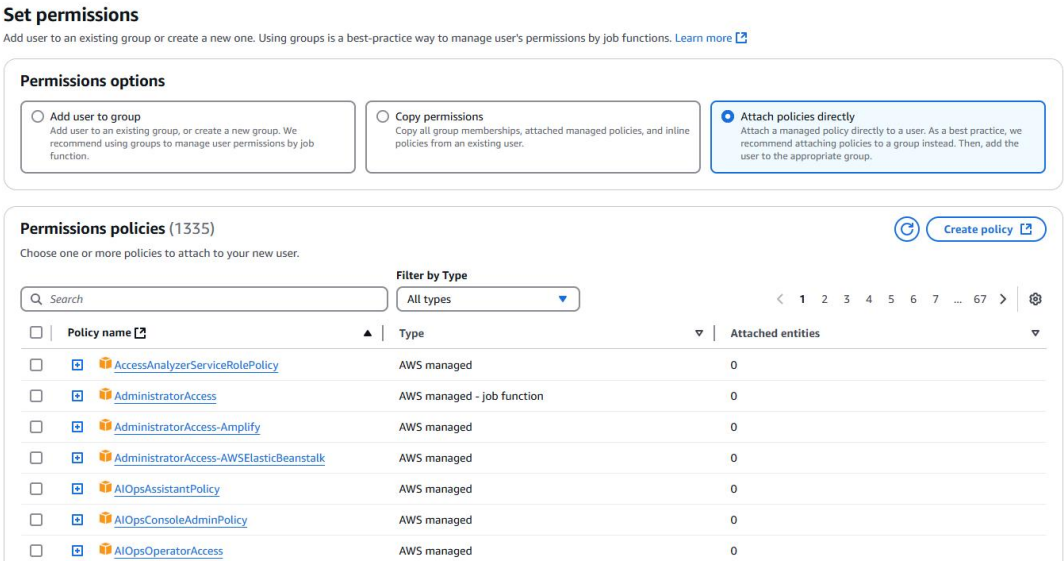


Apart from browser access, AWS console can be accessed via AWS CLI, Terraform, Software Development Kit (SDK). Using Security key or Access key only we can access AWS console

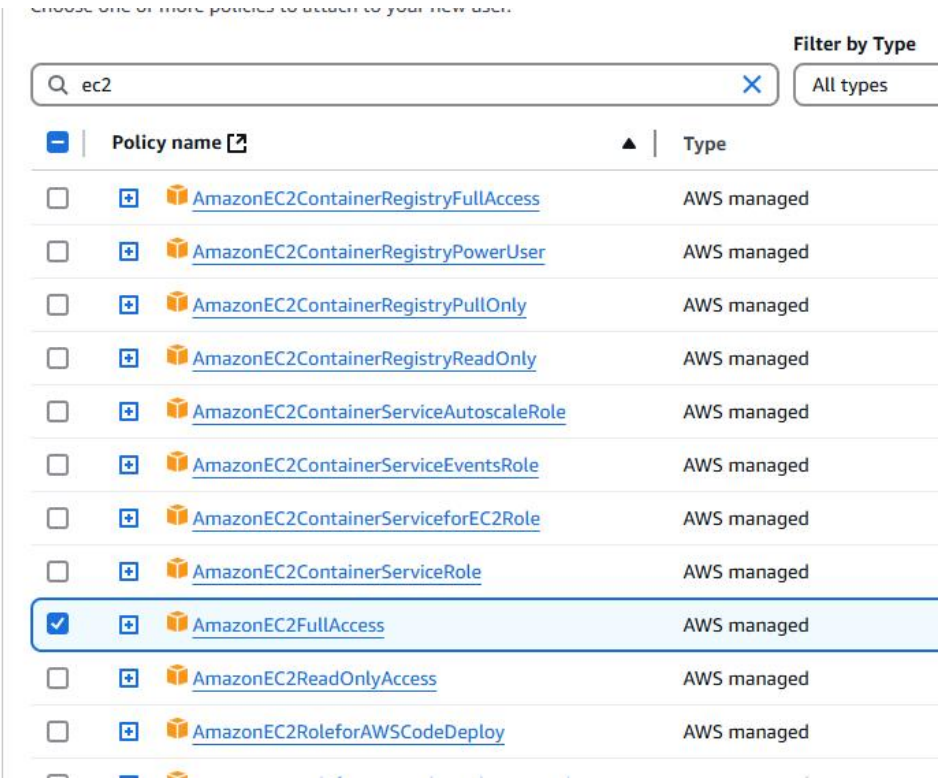
Lets use the second option



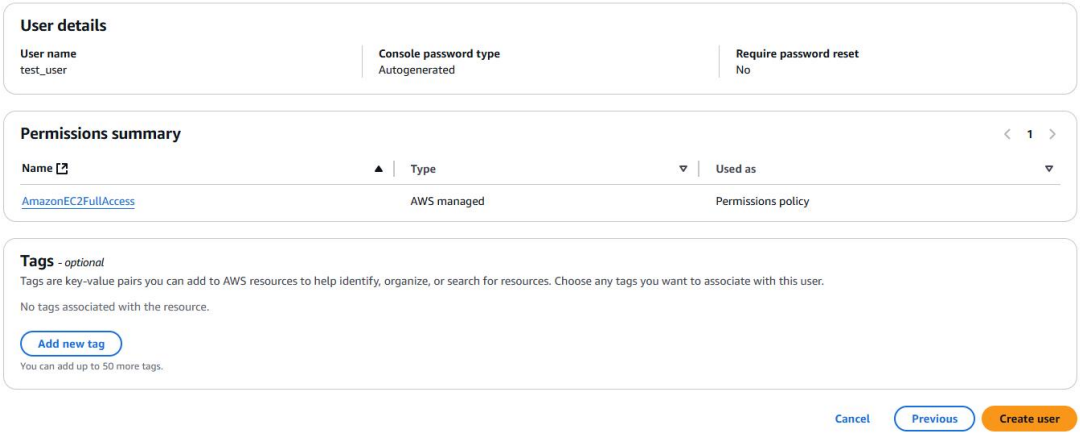
'Attach policies directly' means what the user can access



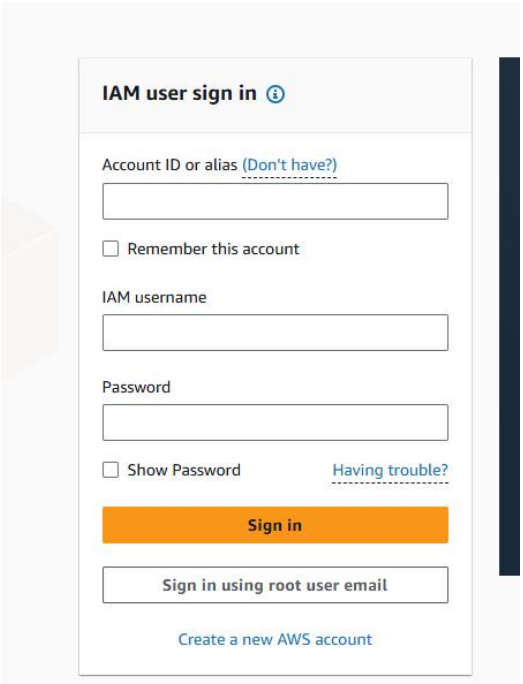
I want to give only EC2 access

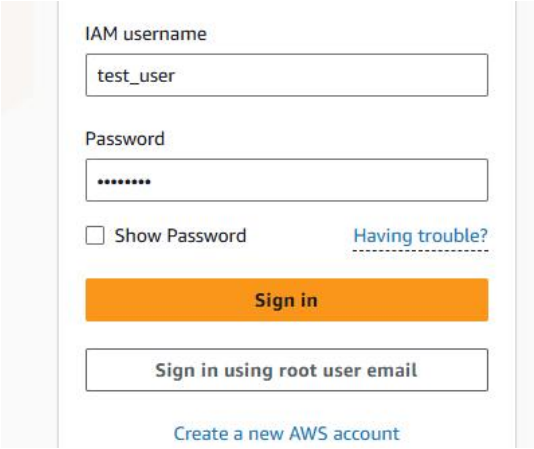


In total, we can give 1335 permission policies to your user ---> AWS managed policies

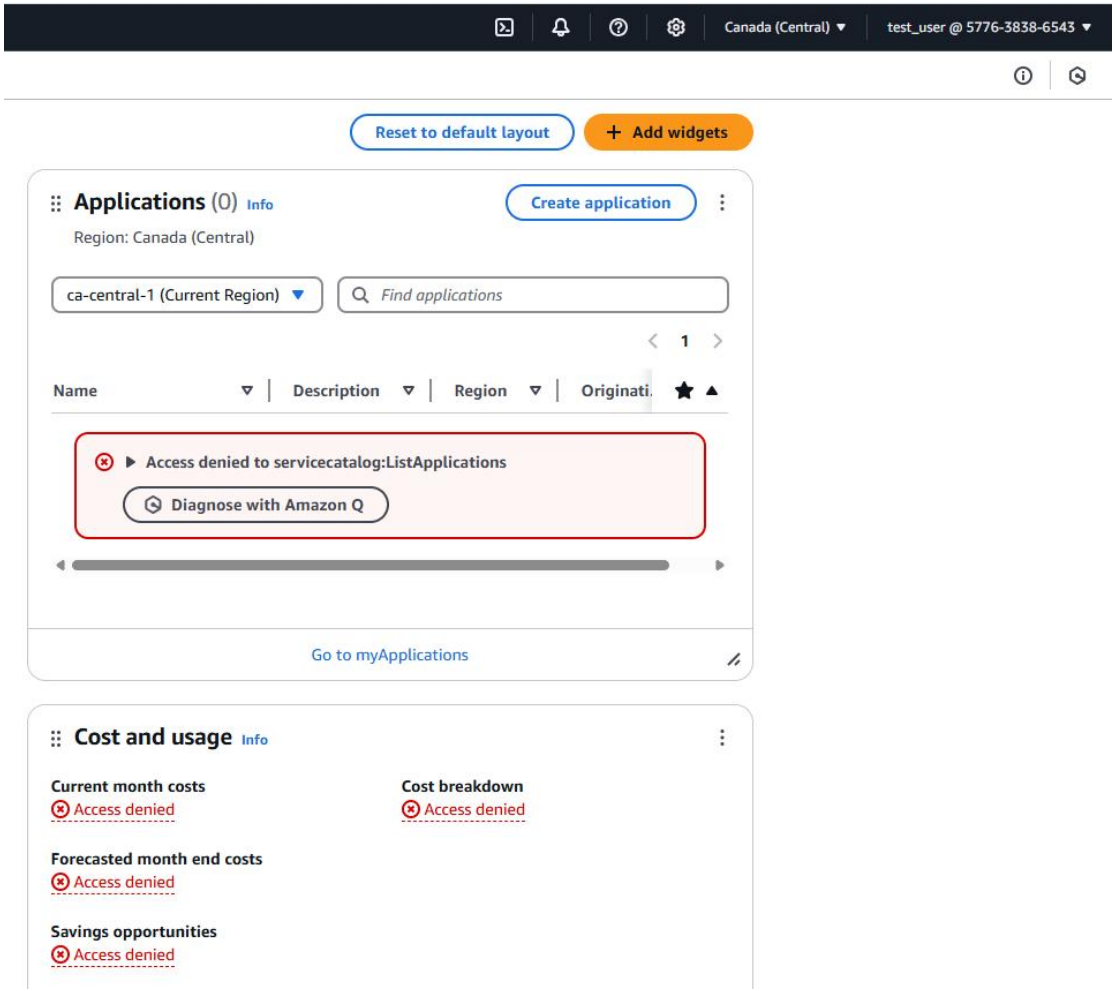


Click Create user
Sign out of account



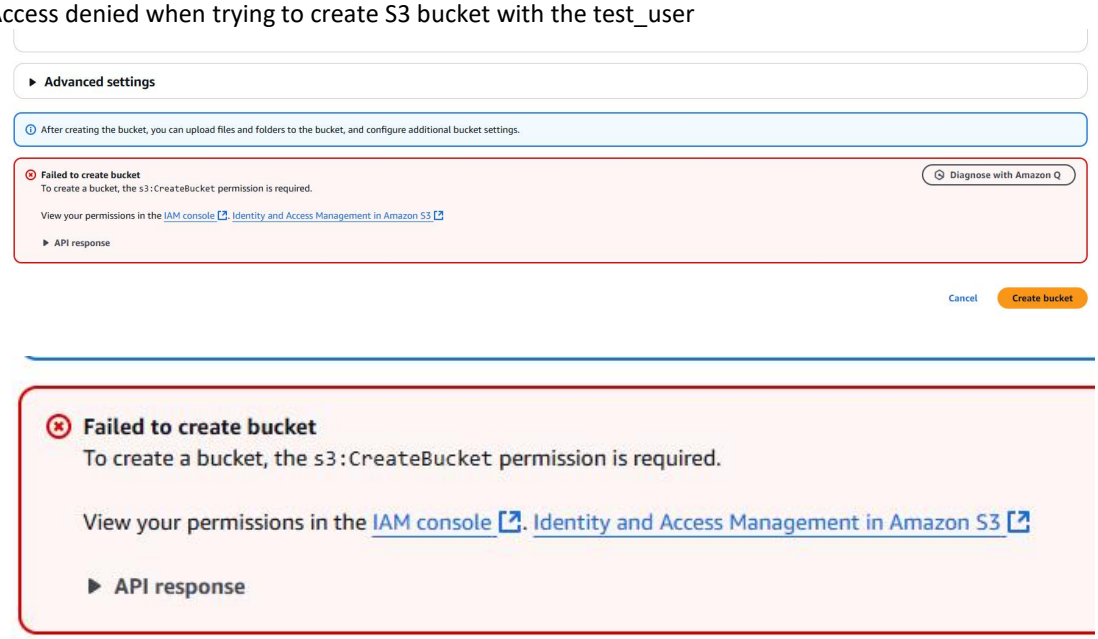


For example, Cost and usage is access denied

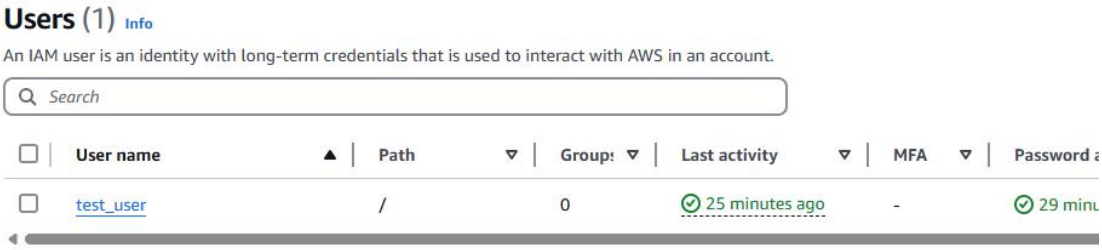


Only EC2 access is there for this user

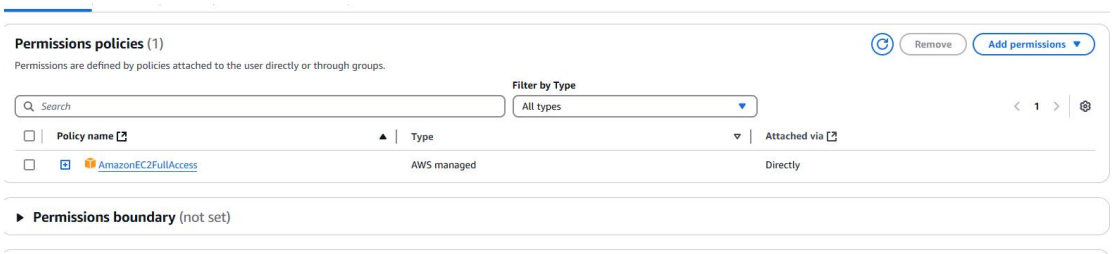
Access denied when trying to create S3 bucket with the test_user



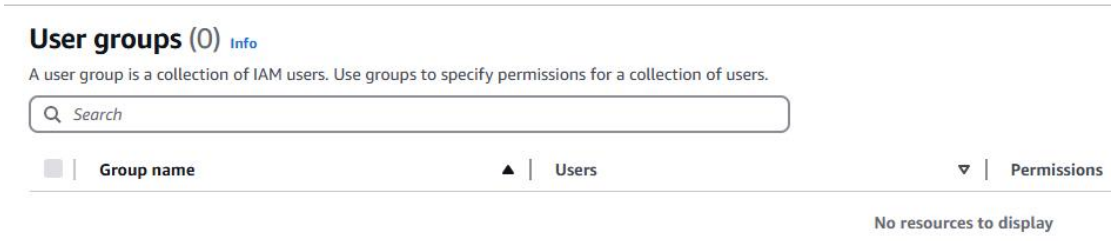
Login as root user



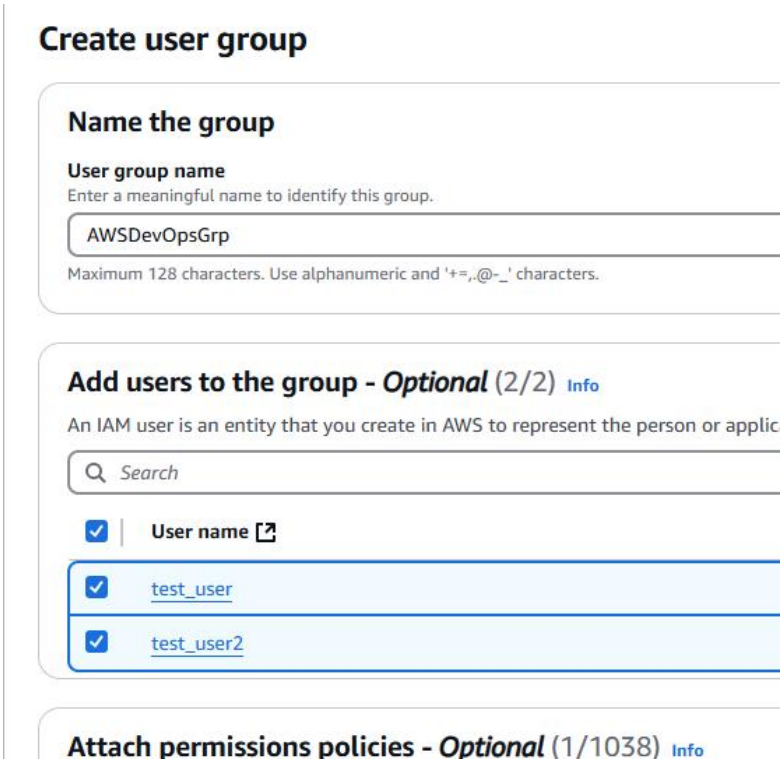
Later also I can modify Permissions policies



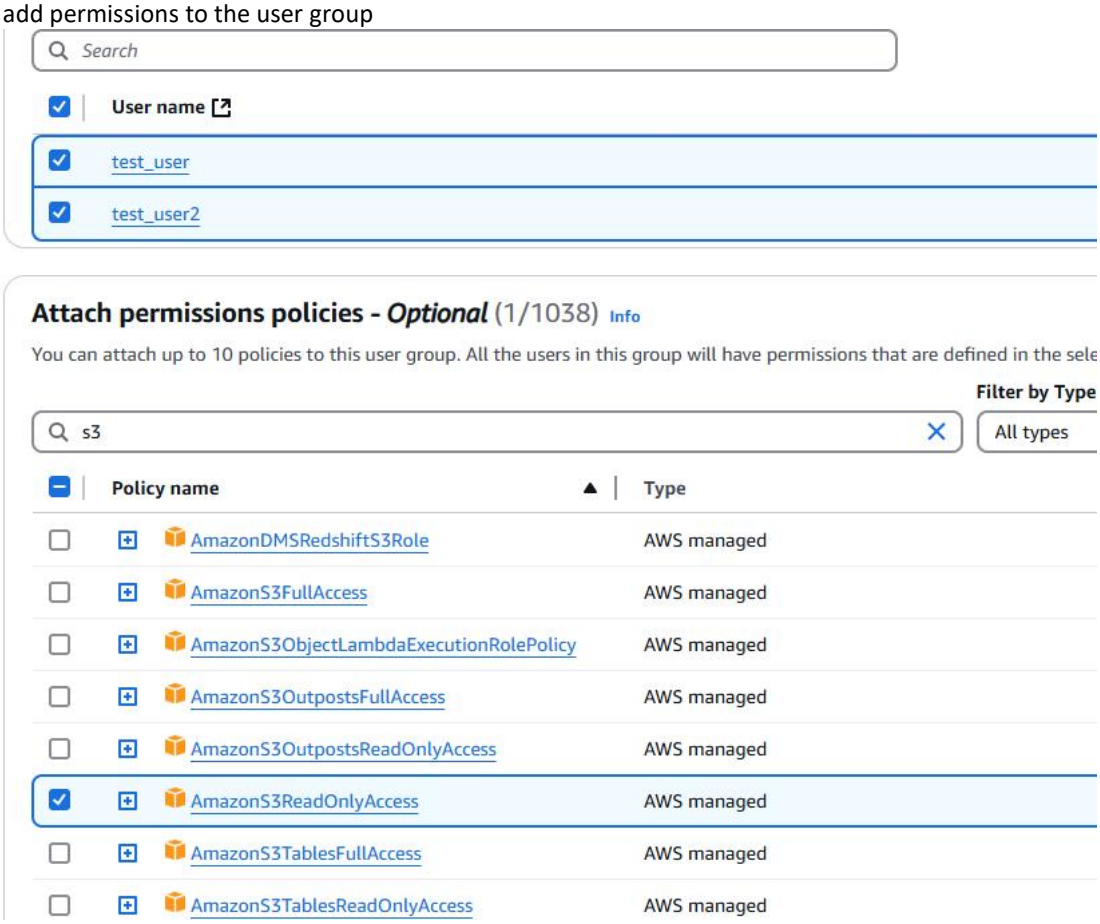
User groups are also there



I add users to user group



I add permissions to the user group



User group is created

AWSDevOpsGrp user group created.

User groups (1) Info

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

Search

Group name	Users	Permissions
AWSDevOpsGrp	2	Defined

Policies
AWS managed policies

Policies (1335) Info

A policy is an object in AWS that defines permissions.

Filter by Type

Search

All types

Policy name	Type	Us
AccessAnalyzerServiceRolePolicy	AWS managed	No
AdministratorAccess	AWS managed - job function	No
AdministratorAccess-Amplify	AWS managed	No
AdministratorAccess-AWSElasticBeanstalk	AWS managed	No
AIOpsAssistantPolicy	AWS managed	No
AIOpsConsoleAdminPolicy	AWS managed	No
AIOpsOperatorAccess	AWS managed	No
AIOpsReadOnlyAccess	AWS managed	No
AlexaForBusinessDeviceSetup	AWS managed	No

How to create my own policy?
Click Create policy

Create policy

Specify permissions Info

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

Policy editor

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.

Resources

Specify resource ARNs for these actions.

Request conditions - *optional*

Actions on resources are allowed or denied only when these conditions are met.

+ Add more permissions

I can select whatever policies I want to include

S3

Allow 6 Actions

Specify what actions can be performed on specific resources in S3.

Actions allowed

Specify actions from the service to be allowed.

Filter Actions

Manual actions | Add actions

- [ ] All S3 actions (s3:*)

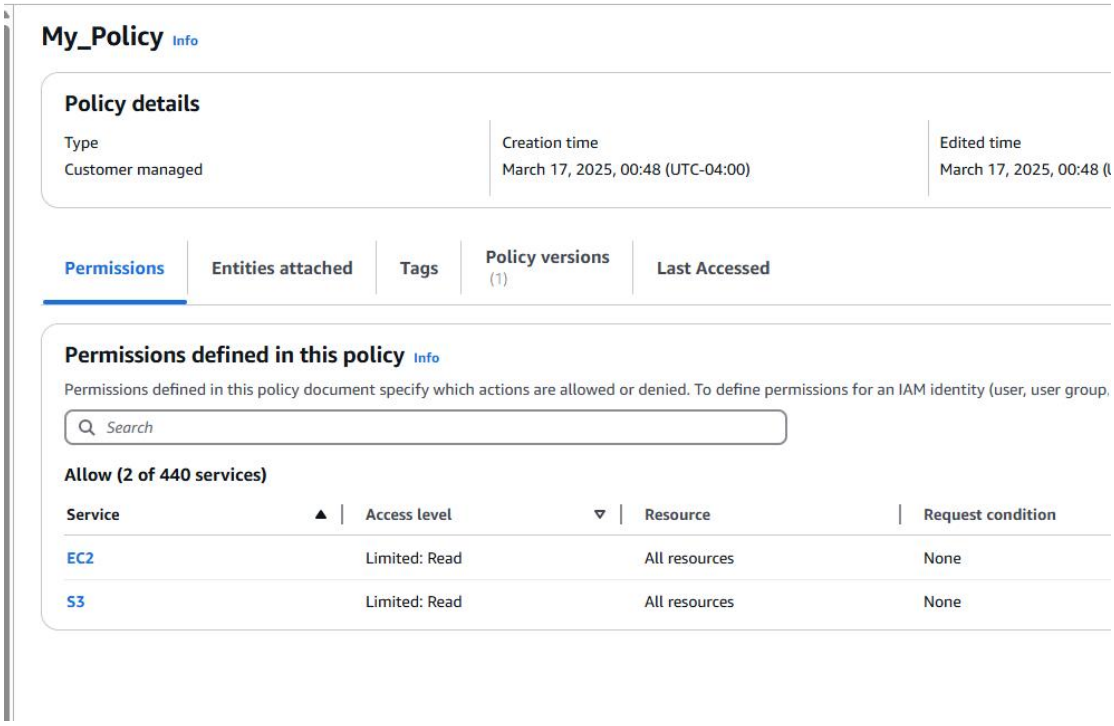
Access level

List (16)

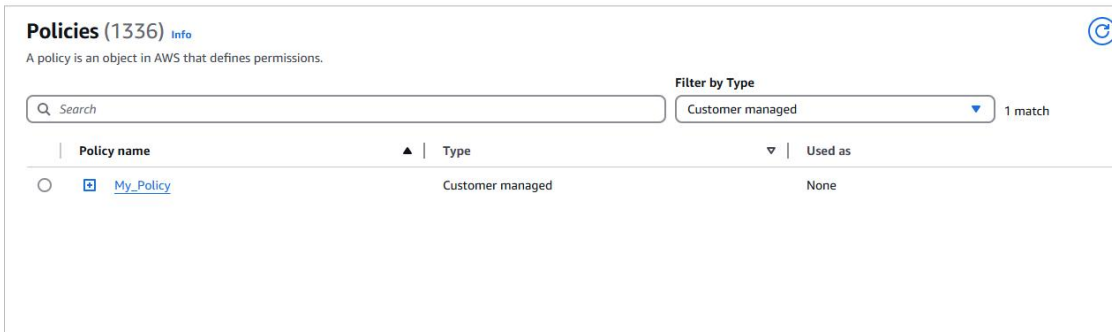
Read (Selected 6/61)

- [ ] All read actions
- [ ] DescribeJob Info
- [ ] GetAccessGrant Info
- [ ] GetAccessGrantsInstanceResourcePolicy Info
- [ ] GetAccessPointConfigurationForObjectLambda Info
- [ ] GetAccessPointPolicyForObjectLambda Info
- [ ] GetAccountPublicAccessBlock Info
- [ ] GetBucketCORS Info
- [ ] GetBucketMetadataTableConfiguration Info
- [ ] DescribeMultiRegionAccessPointOperation
- [ ] GetAccessGrantsInstance Info
- [ ] GetAccessGrantsLocation Info
- [ ] GetAccessPointForObjectLambda Info
- [ ] GetAccessPointPolicyStatus Info
- [ ] GetAnalyticsConfiguration Info
- [ ] GetBucketLocation Info
- [ ] GetBucketNotification Info

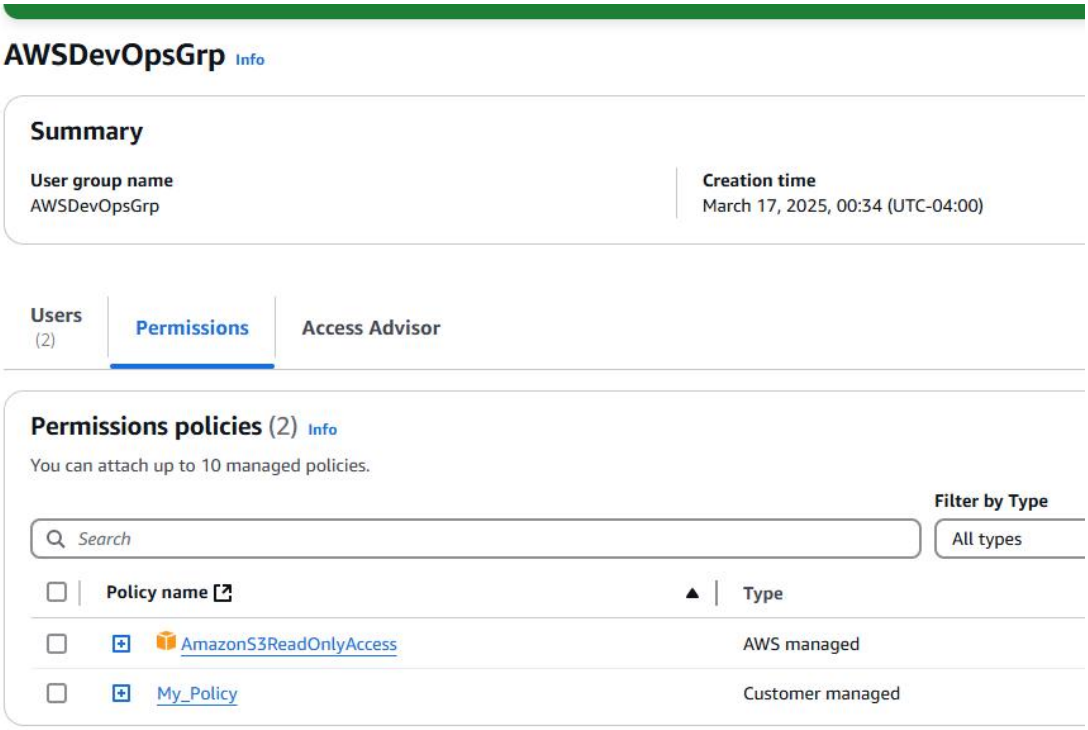
I have created the custom policy



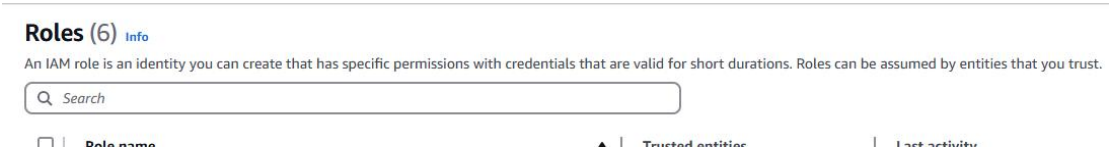
I have got an extra option here: Custom managed



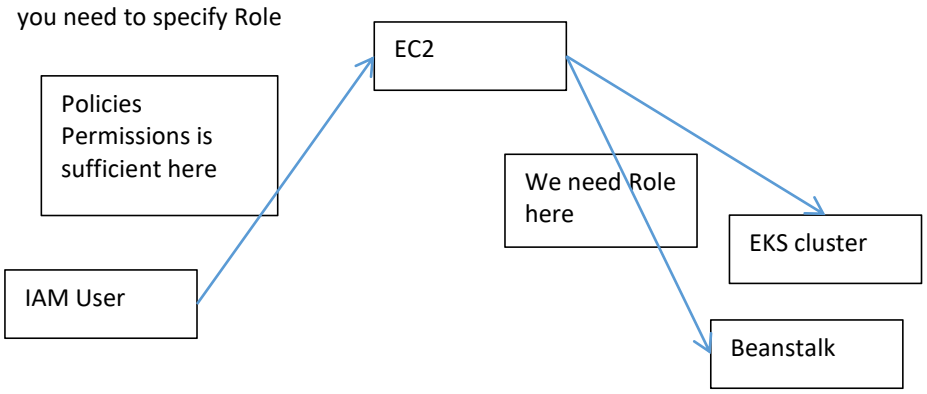
We have two policies for this user group
One is AWS managed and other is Customer managed



Roles



IAM Roles: Inside AWS cloud we have a service EC2, say if the root user has given you (IAM User) a policy to access EC2. can you access EC2? Yes. From EC2, IAM user is accessing EKS, Beanstack then you need to specify Role



We (IAM users) need 'Role' to access one service from another service.

Which Role you want to create?

> Create role

itity

d 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 federat web identity provide perform actions in t
- ( ) **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.

Service or use case

Use case

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

Service or use case

EC2

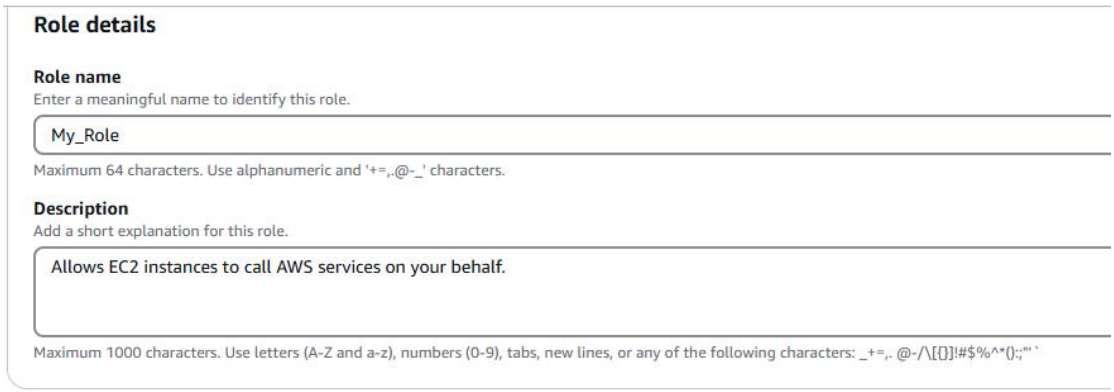
Choose a use case for the specified service.

Use case

- (●) **EC2** — Allows EC2 instances to call AWS services on your behalf.
- ( ) **EC2 Role for AWS Systems Manager** — Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.
- ( ) **EC2 Spot Fleet Role** — Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.
- ( ) **EC2 - Spot Fleet Auto Scaling** — Allows Auto Scaling to access and update EC2 spot fleets on your behalf.
- ( ) **EC2 - Spot Fleet Tagging** — Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.
- ( ) **EC2 - Spot Instances** — Allows EC2 Spot Instances to launch and manage spot instances on your behalf.
- ( ) **EC2 - Spot Fleet**

[ ]	AWSElasticBeanstalkRoleCore	AWS managed
[ ]	AWSElasticBeanstalkRoleCWL	AWS managed
[ ]	AWSElasticBeanstalkRoleECS	AWS managed
[ ]	AWSElasticBeanstalkRoleRDS	AWS managed
[ ]	AWSElasticBeanstalkRoleSNS	AWS managed
[ ]	AWSElasticBeanstalkRoleWorkerTier	AWS managed
[ ]	AWSElasticBeanstalkService	AWS managed
[x]	AWSElasticBeanstalkWebTier	AWS managed
[x]	AWSElasticBeanstalkWorkerTier	AWS managed

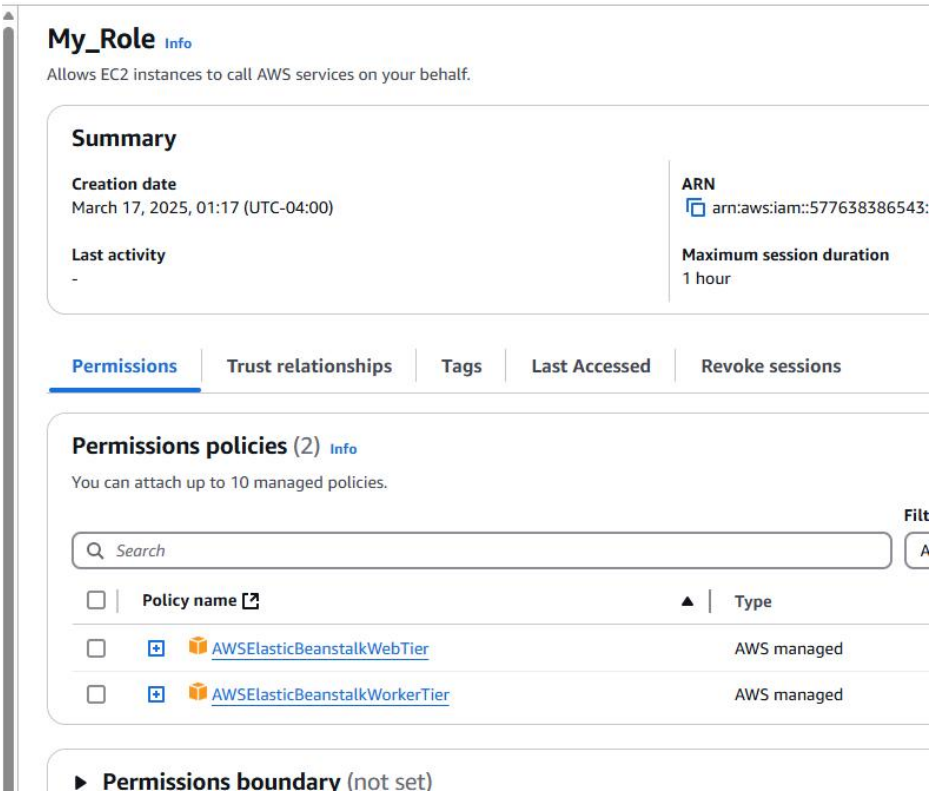
▸ Set permissions boundary - optional



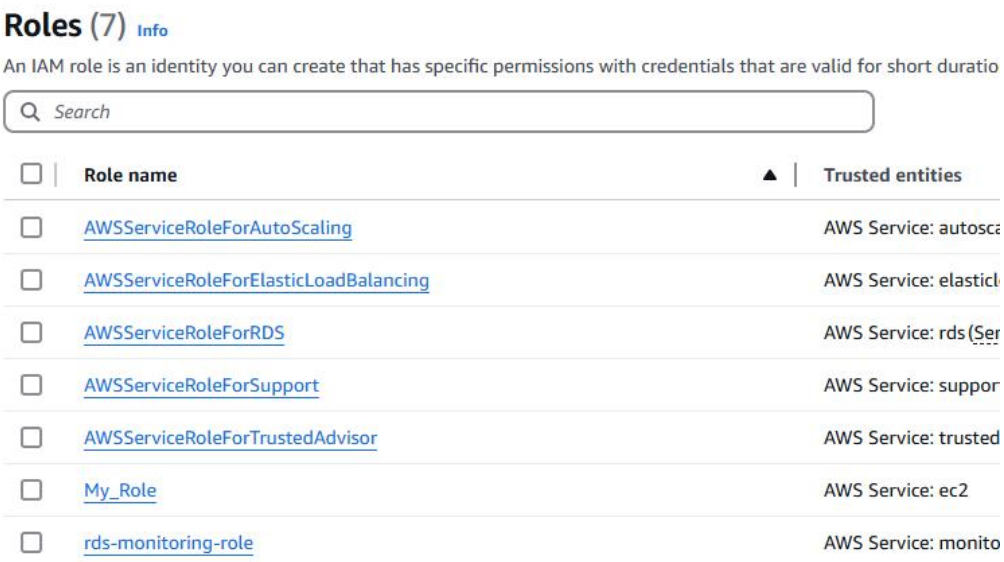
Step 1: Select trusted entities

Allows EC2 instances to call AWS services on your behalf

Click Create Role

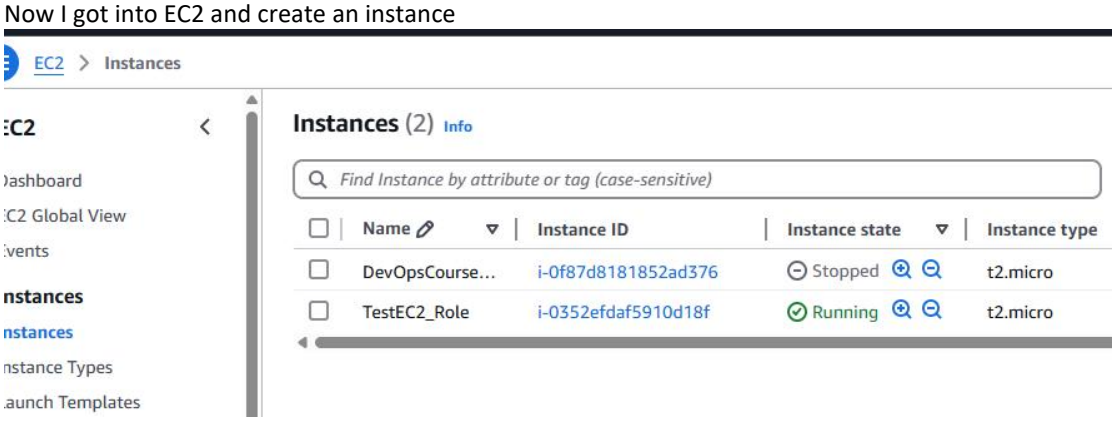


My_Role is added to the list of Roles

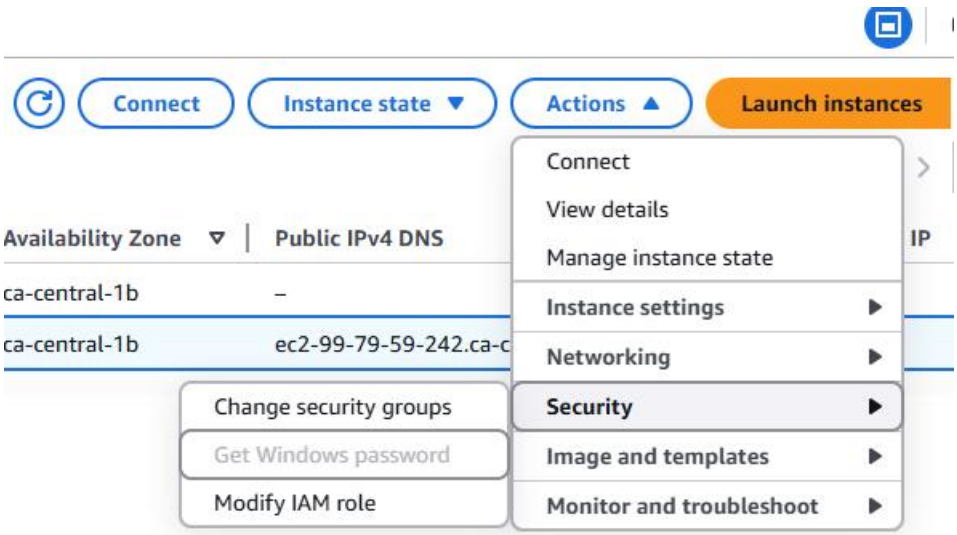


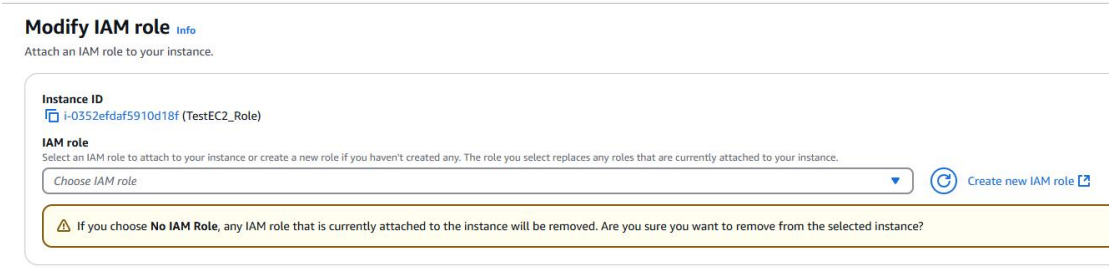
Role demo:

Now I got into EC2 and create an instance

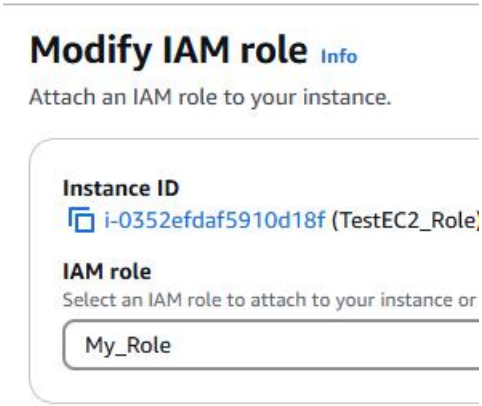


Click on Actions ---> Security ---> Modify IAM role





I select My_Role



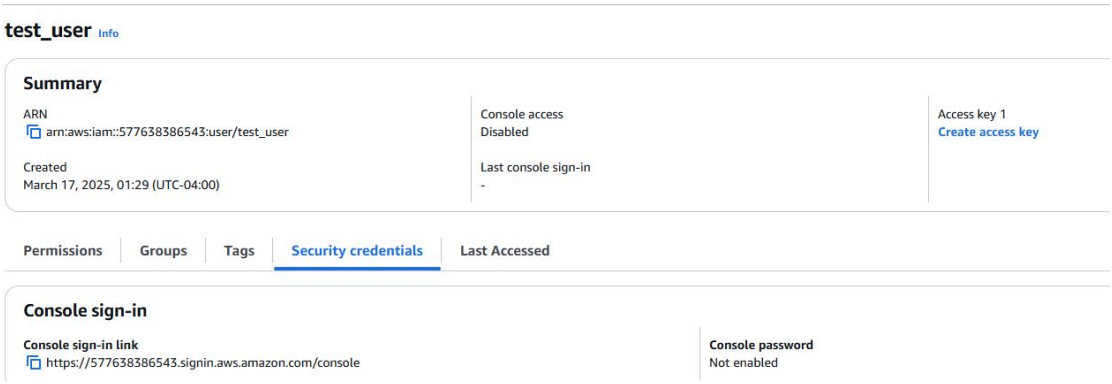
Click Modify IAM Role

Now can you access Elastic BeanStalk from this EC2? YES

Role provide Service to Service access

I create another User

Now, Create access key



Access key best practices & alternatives Info

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alte

Use case

- ○ Command Line Interface (CLI)
  You plan to use this access key to enable the AWS CLI to access your AWS account.
- ○ Local code
  You plan to use this access key to enable application code in a local development environment to access your AWS account.
- ○ Application running on an AWS compute service
  You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
- ○ Third-party service
  You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
- ○ Application running outside AWS
  You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.
- ○ Other
  Your use case is not listed here.

Application running outside AWS means SDK

CLI

- ◉ Command Line Interface (CLI)
  You plan to use this access key to enable the AWS CLI to access your AWS account.
- ○ Local code
  You plan to use this access key to enable application code in a local development environment to access your AWS account.
- ○ Application running on an AWS compute service
  You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
- ○ Third-party service
  You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
- ○ Application running outside AWS
  You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.
- ○ Other
  Your use case is not listed here.

⚠ **Alternatives recommended**

- Use AWS CloudShell, a browser-based CLI, to run commands. Learn more
- Use the AWS CLI V2 and enable authentication through a user in IAM Identity Center. Learn more

Confirmation

☑ I understand the above recommendation and want to proceed to create an access key.

Cancel Next

We need both Security access key and Access key for CLI access

Retrieve access keys Info

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key

Access key	Secret access key
AKIAYM7POJNXRPKLJ3RF	********** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

We have understood about User, UserGroups, Policies --> AWS managed policies, Customer managed policies, Roles -->