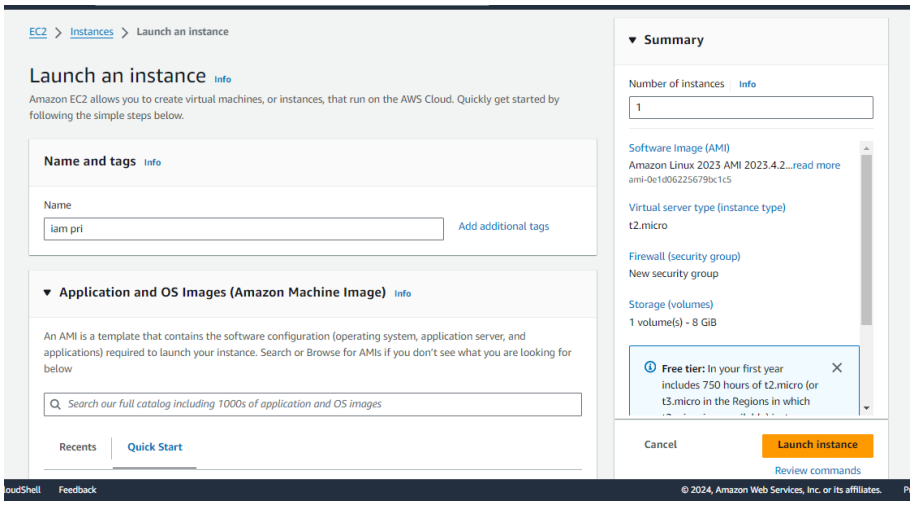
IDENTITY ACCESS MANAGEMENT

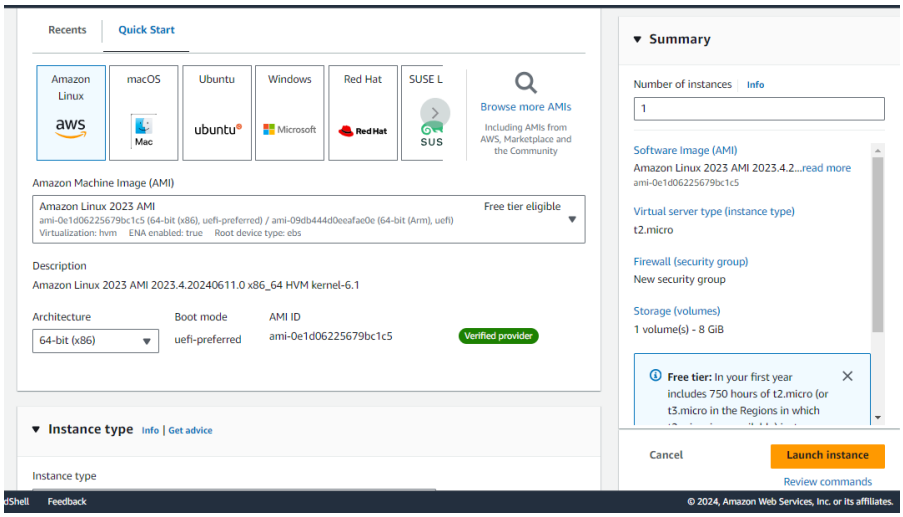
AWS IAM (Identity and Access Management) is a service that helps you securely control access to AWS services and resources. It allows you to manage users, groups, roles, and permissions to ensure that only authorized users can perform specific actions.

EC2 INSTANCES

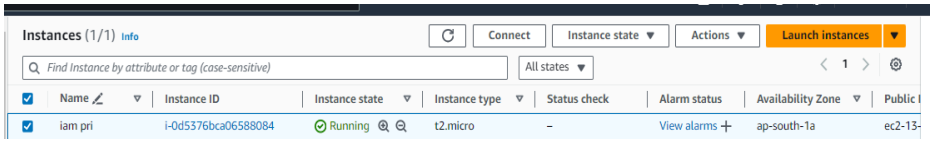
Amazon EC2 (Elastic Compute Cloud) is a service that lets you run virtual servers in the cloud, giving you flexible, scalable computing power. You can quickly launch and manage instances to meet your needs and only pay for the resources you use.

STEP 1 Launch EC2 instance – Amazon Linux





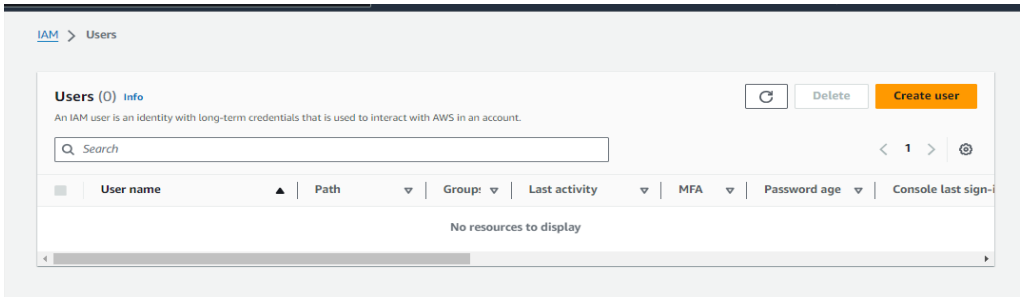
STEP 2 successfully created

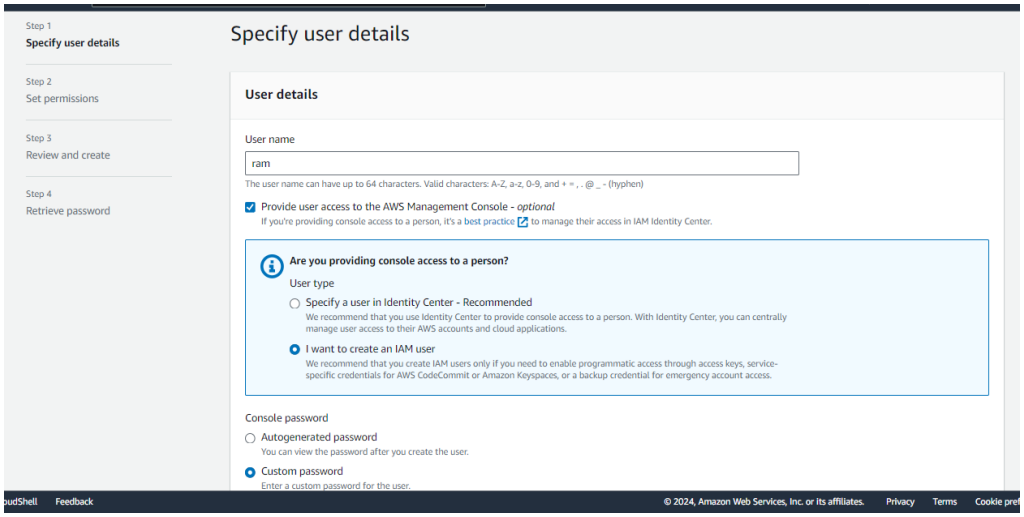


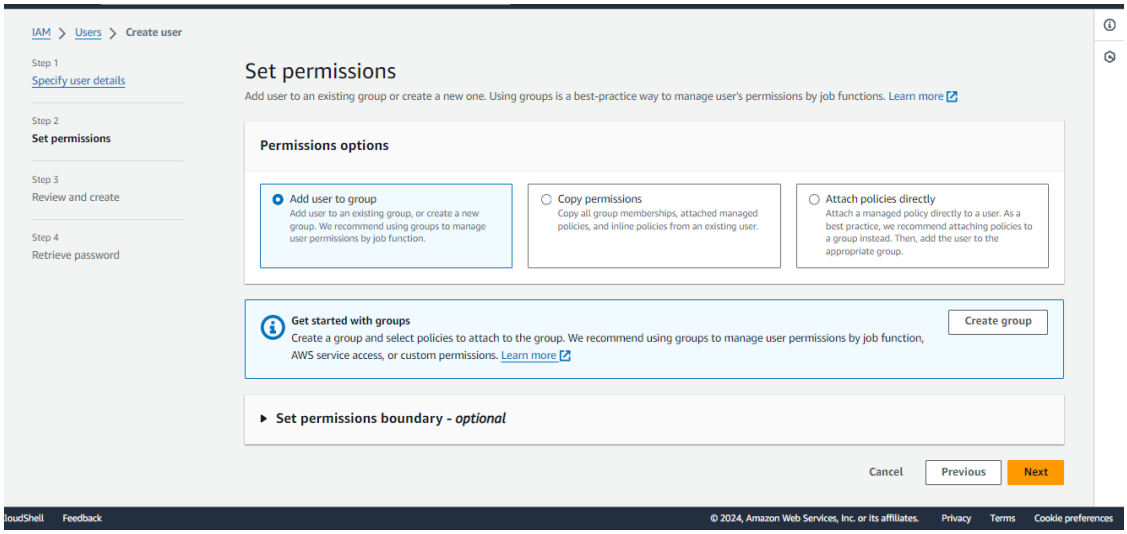
USER

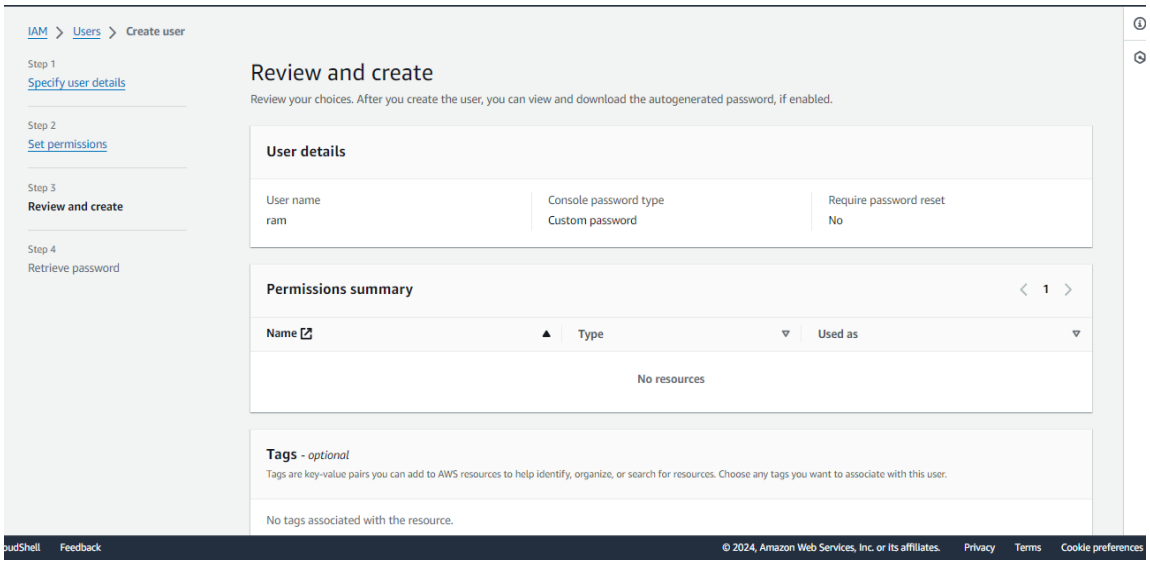
In AWS IAM, a user is an entity that represents a person or service needing access to AWS resources. Users are assigned permissions through policies, allowing them to perform specific actions within AWS.

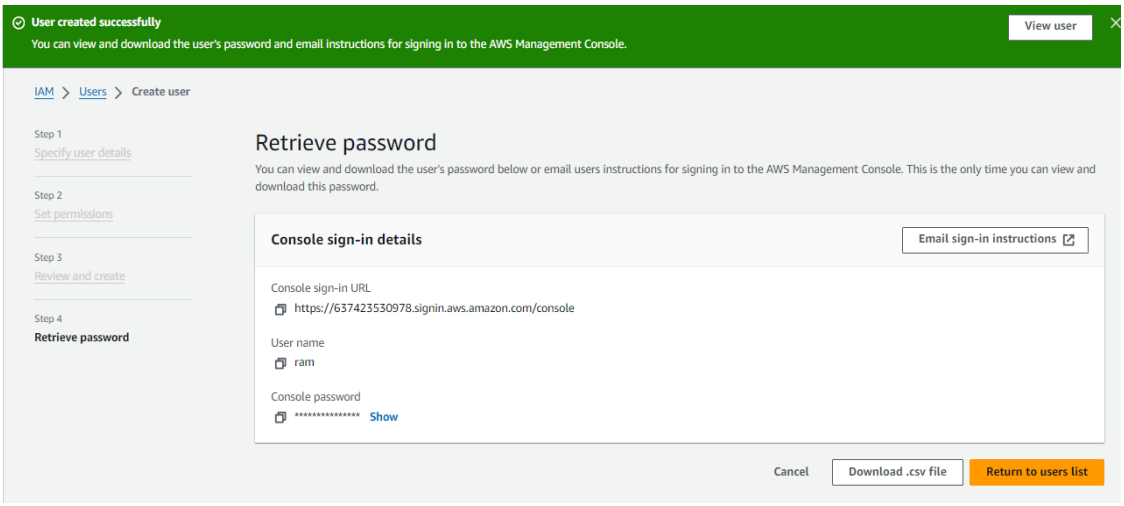
STEP 1 create user



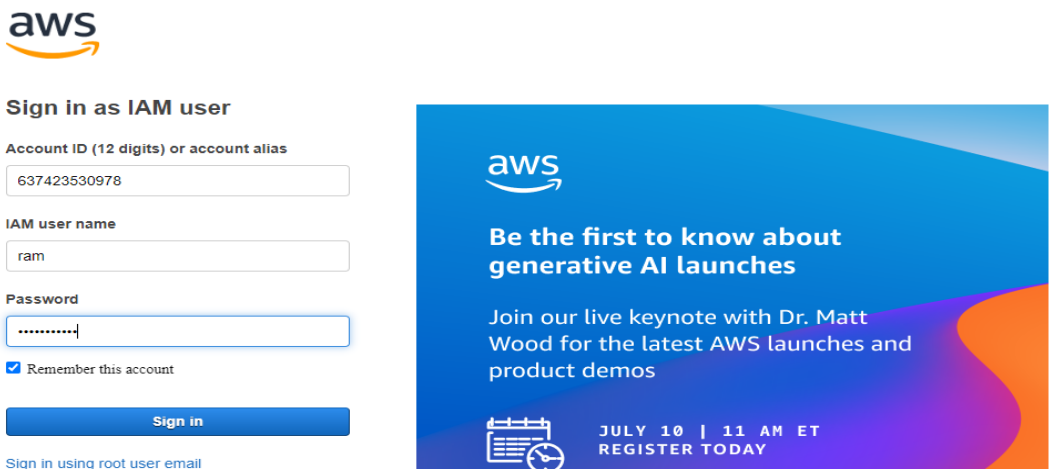

Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
Specify user details
User details
User name
ram
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)
Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.
Are you providing console access to a person?
User type
Specify a user in Identity Center - Recommended
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.
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.
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie pref

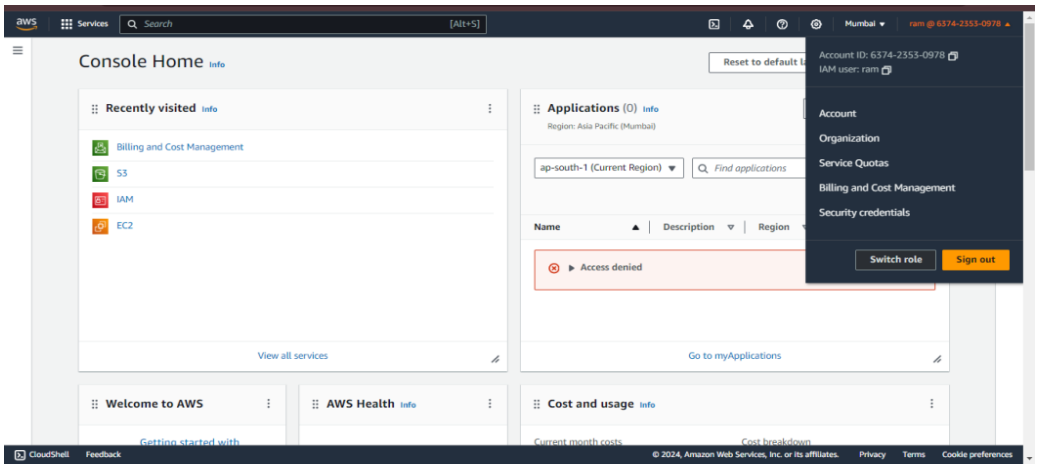

IAM
Users
Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
Set permissions
Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more
Permissions options
Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.
Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.
Get started with groups
Create a group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. Learn more
Create group
Set permissions boundary - optional
Cancel
Previous
Next
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


IAM
Users
Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Step 4
Retrieve password
Review and create
Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.
User details
User name
ram
Console password type
Custom password
Require password reset
No
Permissions summary
1
Name
Type
Used as
No resources
Tags - optional
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.
No tags associated with the resource.
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



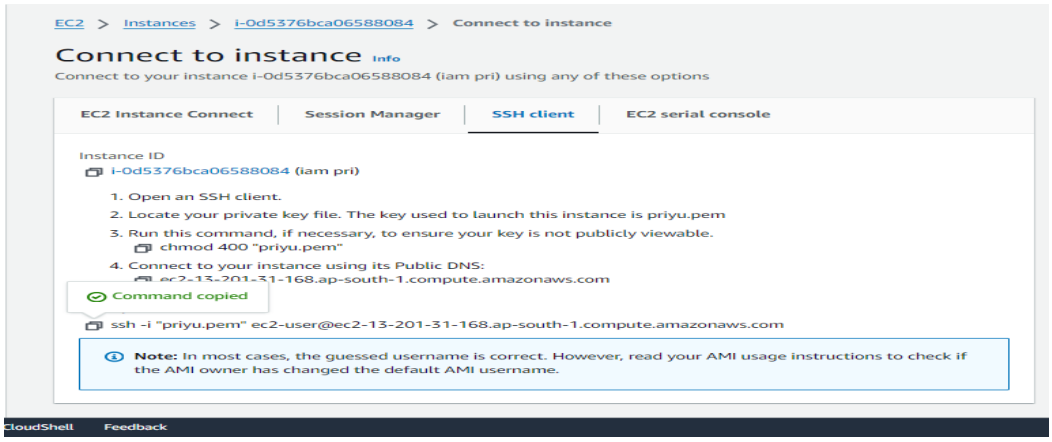
STEP 2 let's try to login using IAM user





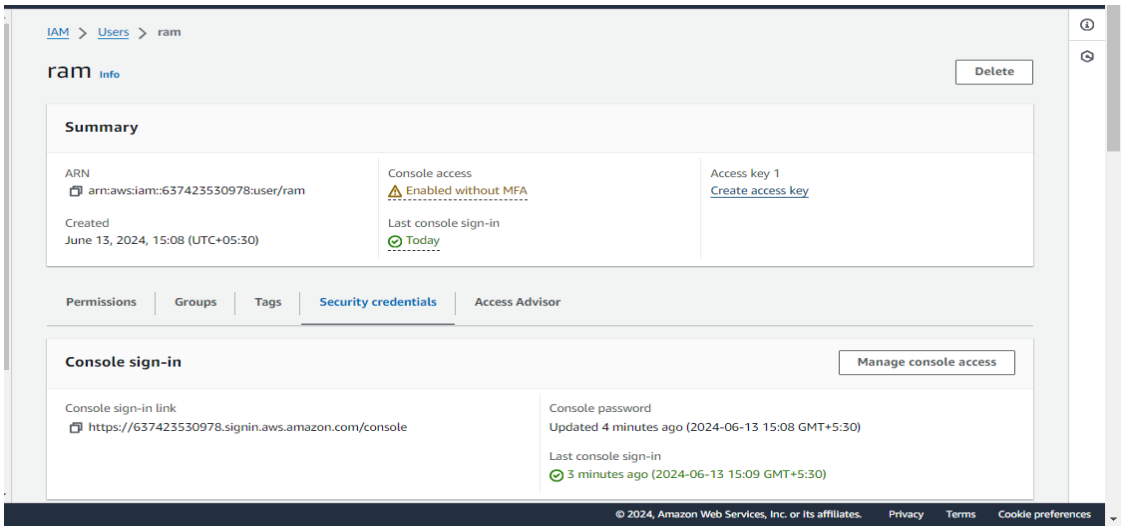
Now switch Root user

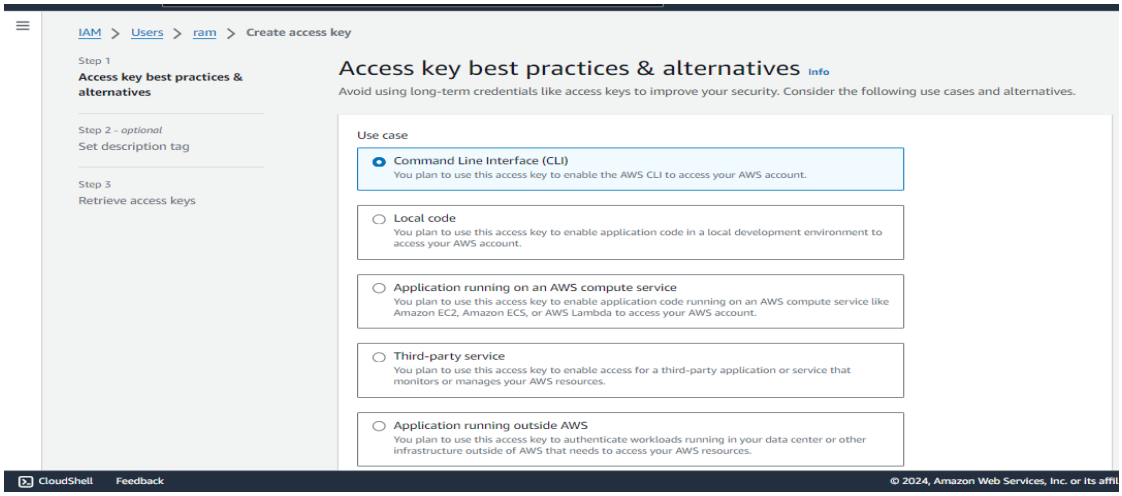
STEP 1 connect the EC2 instance using ssh client

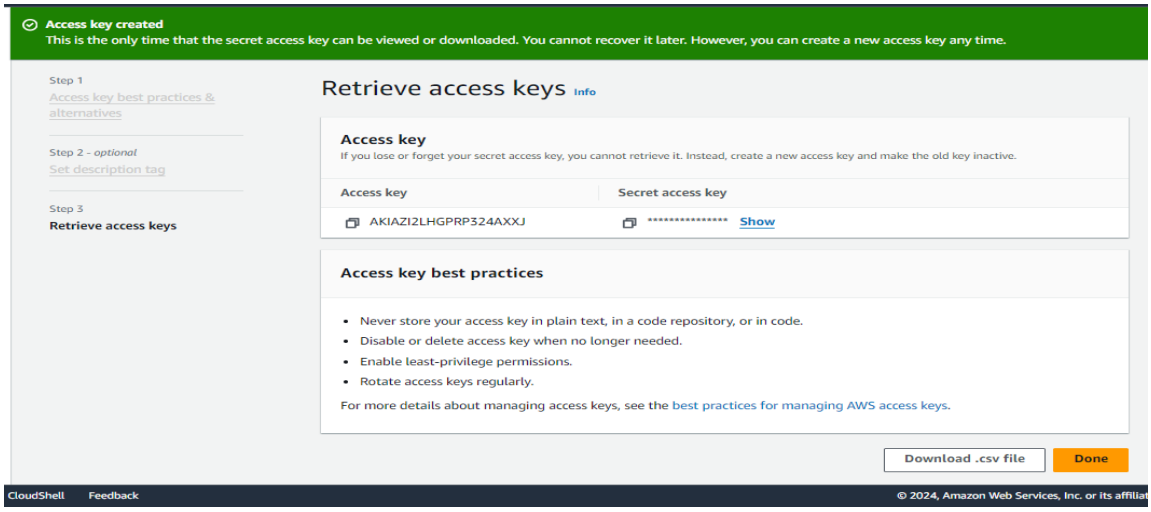


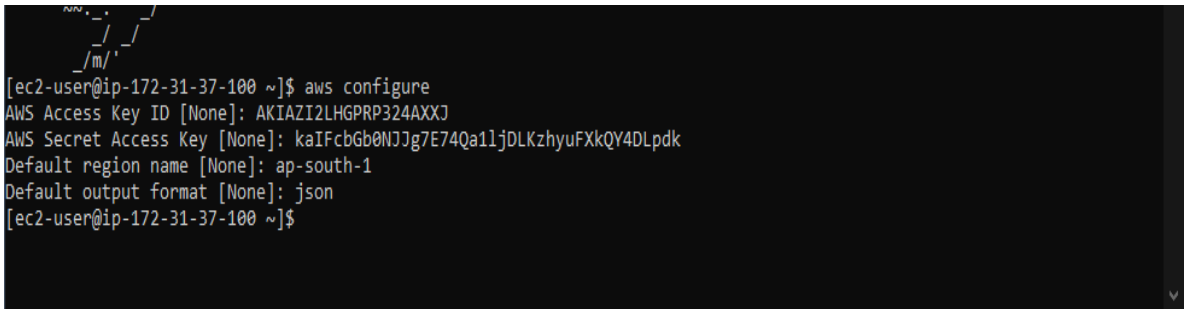

```
C:\Users\Sys>cd downloads

C:\Users\Sys\Downloads>ssh -i "priyu.pem" ec2-user@ec2-13-201-31-168.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-13-201-31-168.ap-south-1.compute.amazonaws.com (13.201.31.168)' can't be established.
ECDSA key fingerprint is SHA256:4Nzk/F2TwF4g+rcZKrkOjLRZJKXz+aQOYiVd6s43zFs.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-201-31-168.ap-south-1.compute.amazonaws.com,13.201.31.168' (ECDSA) to the list of kno
wn hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-37-100 ~]$
```

STEP 3 IAM-users-create access key





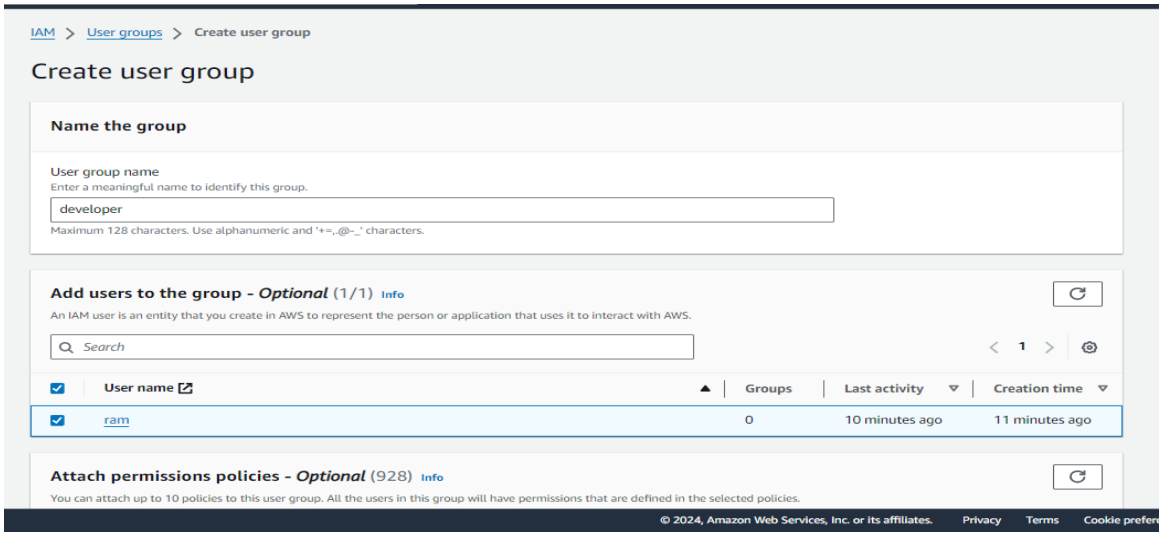


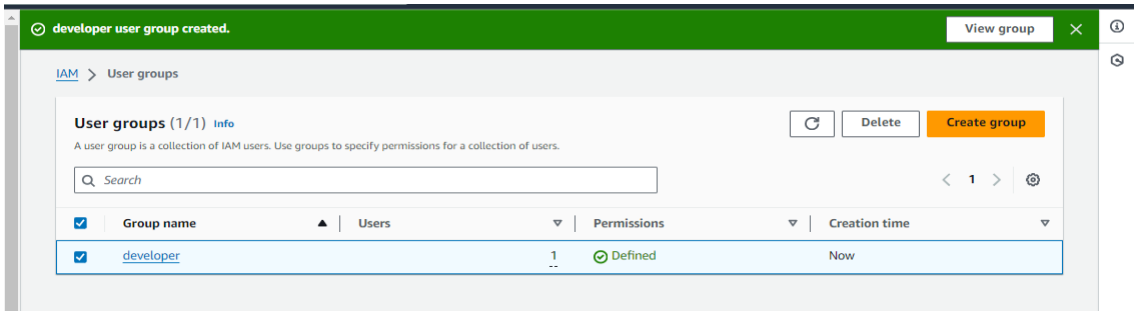


User Groups

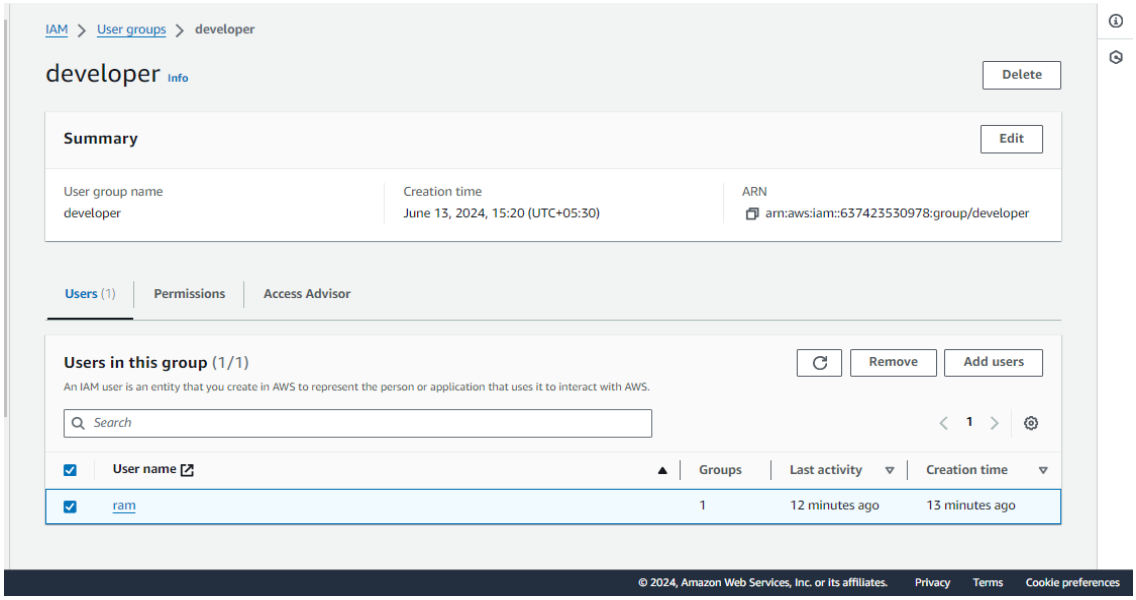
In AWS IAM, a user group is a collection of IAM users that you can manage as a single unit. By assigning permissions to a user group, all users in the group inherit those permissions, simplifying access management.

STEP 1 create user group





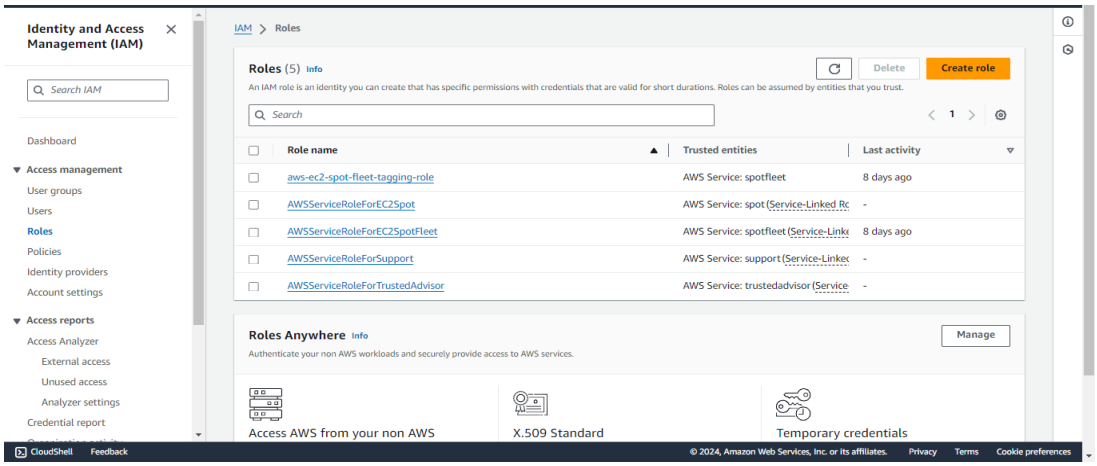
Add or remove users

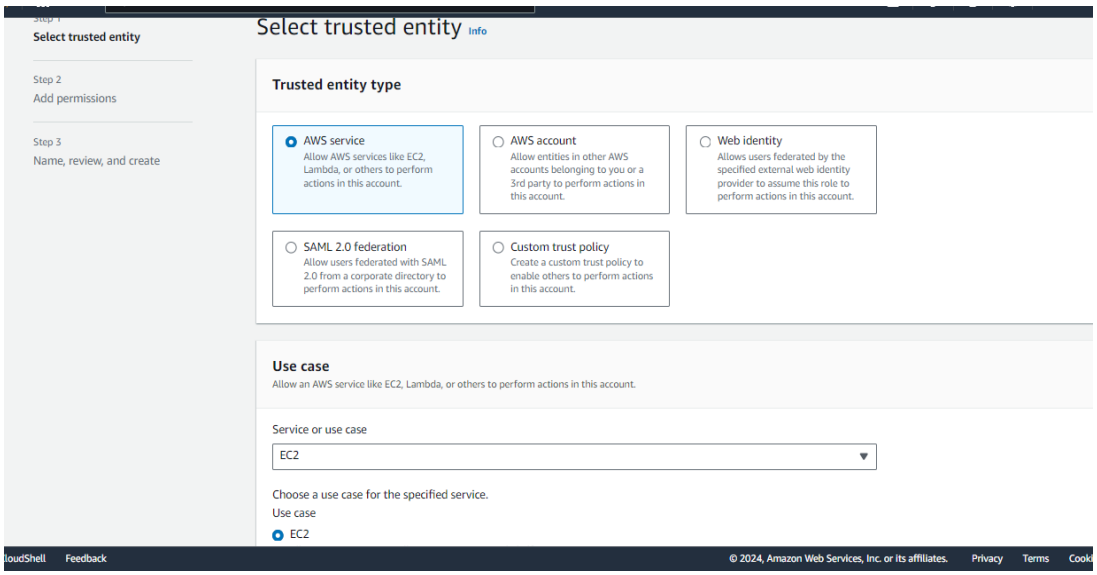


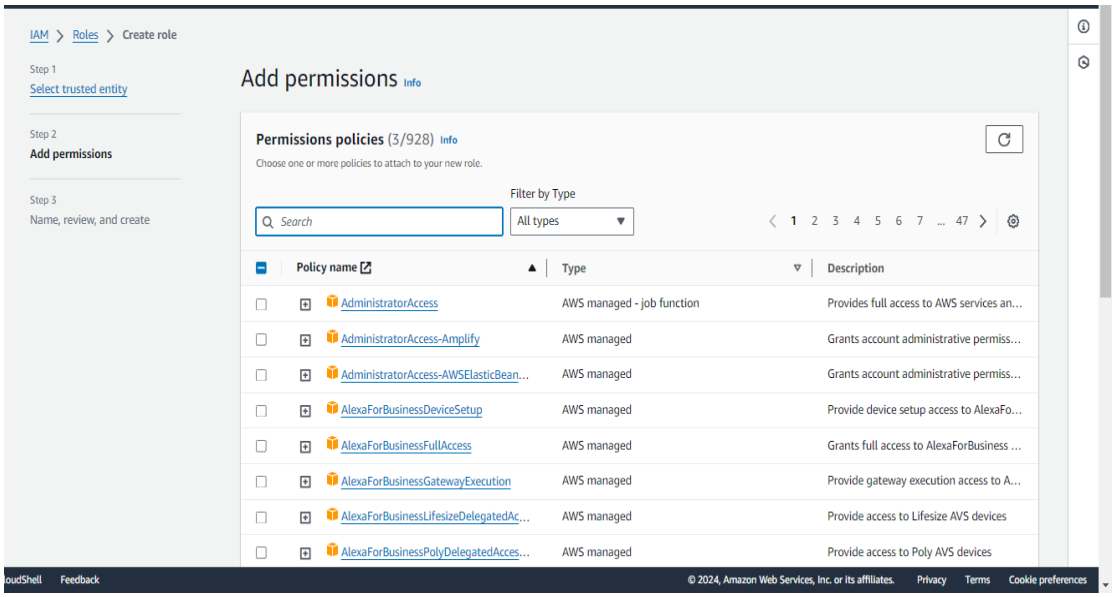
Roles

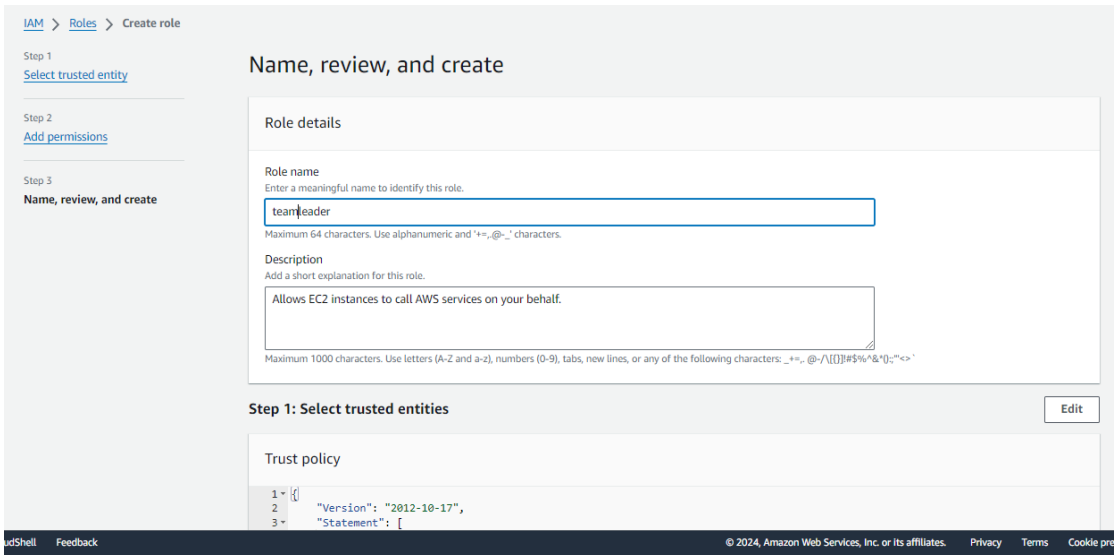
In AWS, roles are a way to grant permissions to entities like EC2 instances, Lambda functions, or users, allowing them to perform specific actions on AWS resources. They use temporary security credentials and can be assumed by trusted entities, enabling secure and controlled access.

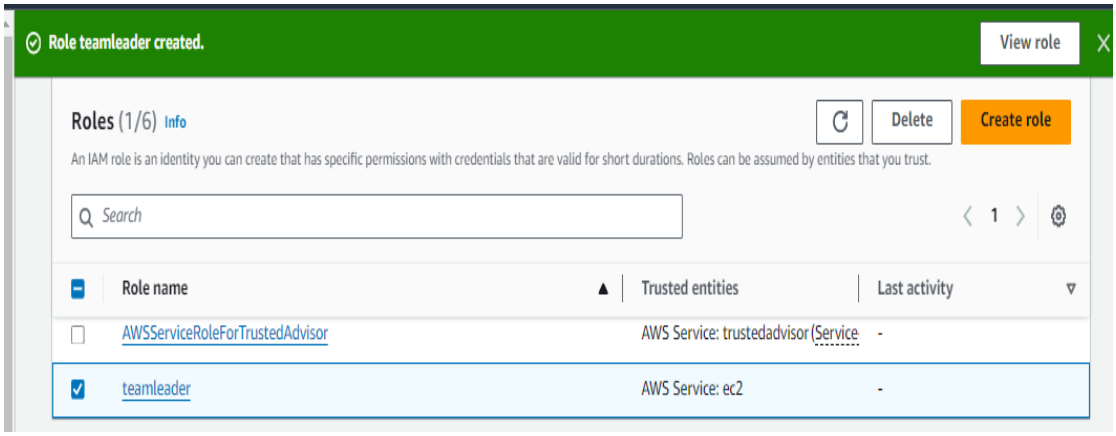
STEP 1 create role






IAM > Roles > Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Add permissions Info
Permissions policies (3/928) Info
Choose one or more policies to attach to your new role.
Filter by Type
Search
All types
1 2 3 4 5 6 7 ... 47
Policy name
Type
Description
AdministratorAccess
AWS managed - job function
Provides full access to AWS services an...
AdministratorAccess-Amplify
AWS managed
Grants account administrative permiss...
AdministratorAccess-AWSElasticBean...
AWS managed
Grants account administrative permiss...
AlexaForBusinessDeviceSetup
AWS managed
Provide device setup access to AlexaFo...
AlexaForBusinessFullAccess
AWS managed
Grants full access to AlexaForBusiness ...
AlexaForBusinessGatewayExecution
AWS managed
Provide gateway execution access to A...
AlexaForBusinessLifesizeDelegatedAc...
AWS managed
Provide access to Lifesize AVS devices
AlexaForBusinessPolyDelegatedAcces...
AWS managed
Provide access to Poly AVS devices
oudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

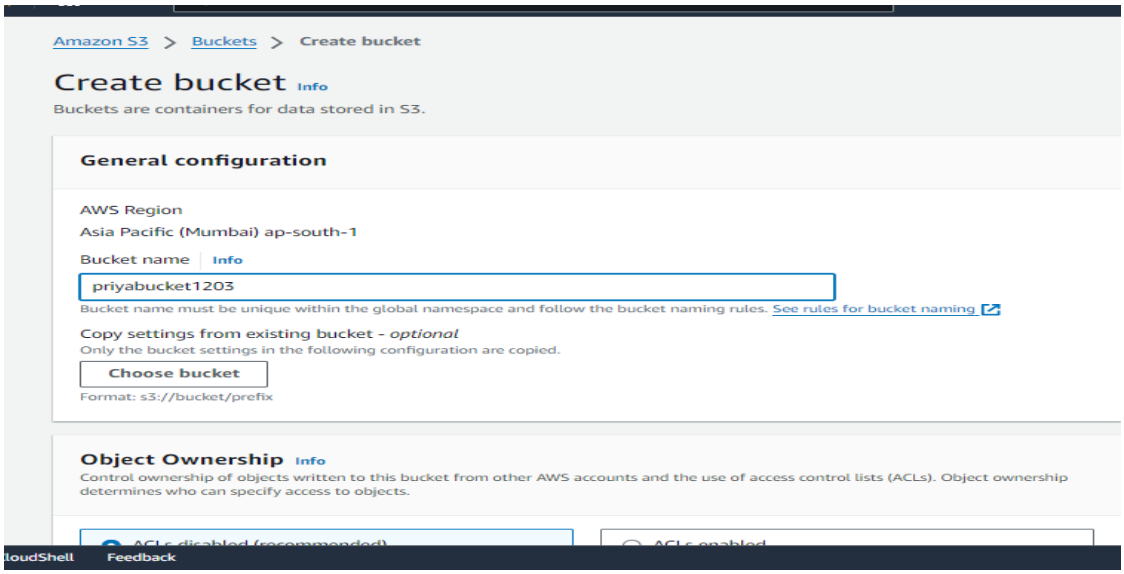

IAM > Roles > Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Name, review, and create
Role details
Role name
Enter a meaningful name to identify this role.
teamleader
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.
Description
Add a short explanation for this role.
Allows EC2 instances to call AWS services on your behalf.
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/\[{}]!#$%^&*();:"<>`
Step 1: Select trusted entities
Edit
Trust policy
{
"Version": "2012-10-17",
"Statement": [
udShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms

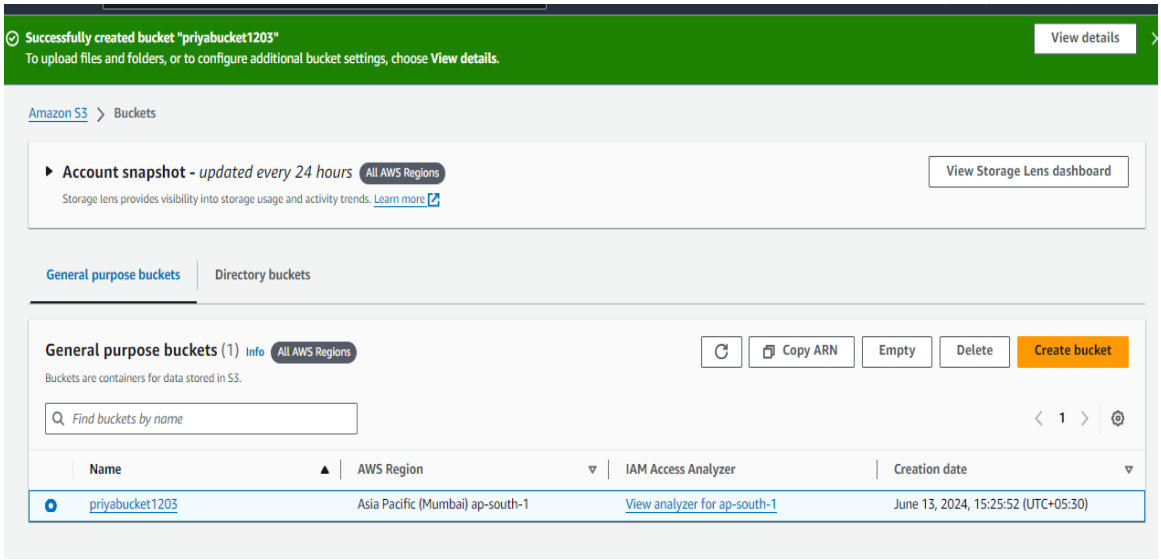

Role teamleader created.
View role
Roles (1/6) Info
Delete
Create role
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.
Search
1
Role name
Trusted entities
Last activity
AWSServiceRoleForTrustedAdvisor
AWS Service: trustedadvisor (Service
-
teamleader
AWS Service: ec2
-

AWS S3

Amazon S3 (Simple Storage Service) is a scalable cloud storage service that allows you to store and retrieve large amounts of data at any time. It offers secure, durable, and highly available object storage, ideal for backups, data archiving, and content distribution.

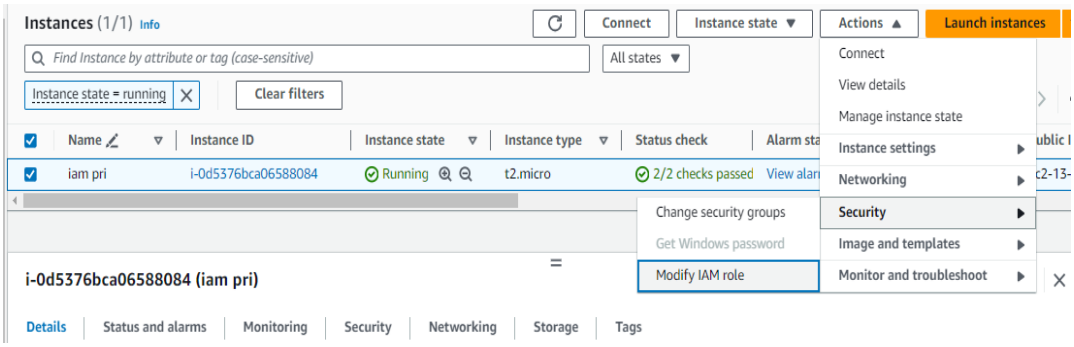
STEP 1 S3 - bucket - create bucket

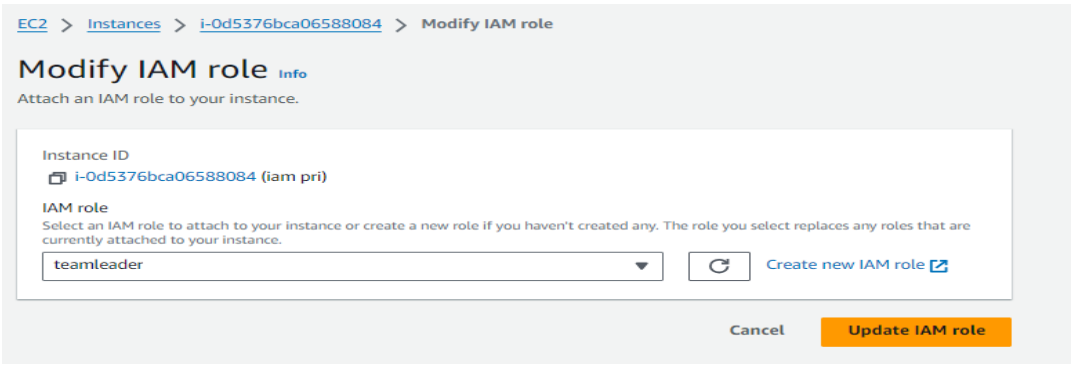


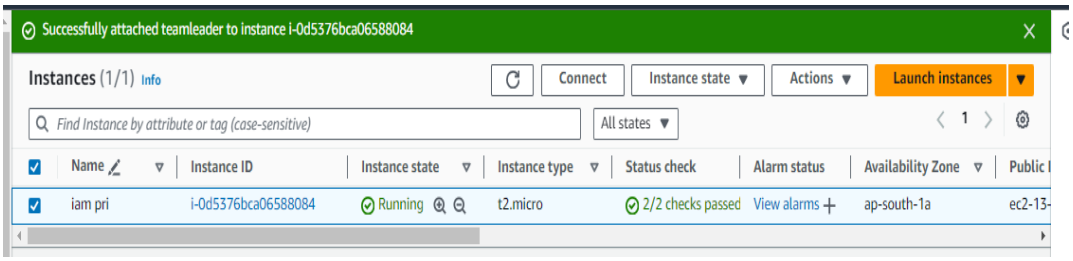


EC2- Modify IAM Role

STEP 1 modify IAM role







STEP 2 Open command prompt

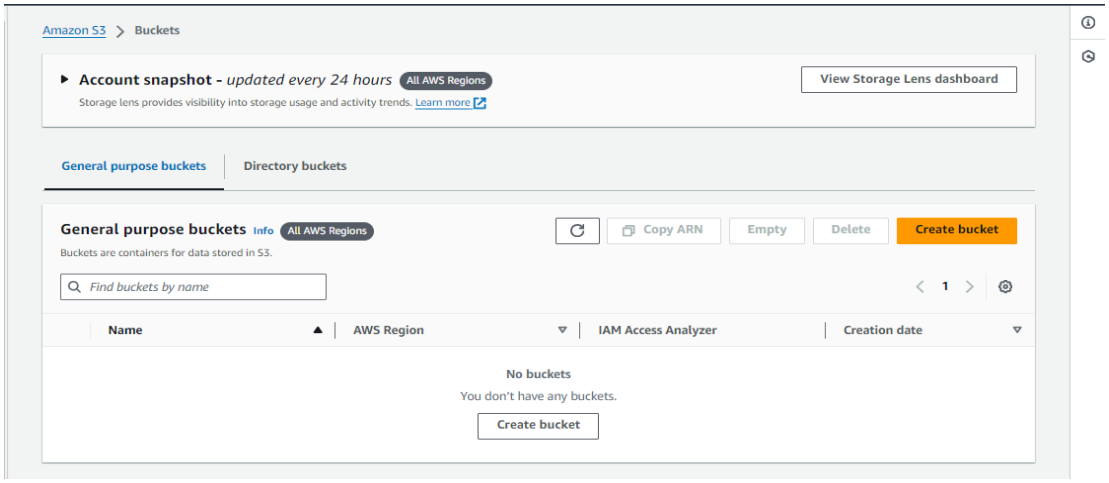
```
[ec2-user@ip-172-31-37-100 ~]$ aws configure
AWS Access Key ID [****************AXXJ]: AKIAZI2LHGPRP324AXXJ
AWS Secret Access Key [****************Lpdk]: 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
Default region name [ap-south-1]: ap-south-1
Default output format [json]: json
[ec2-user@ip-172-31-37-100 ~]$ sudo yum install awscli
Last metadata expiration check: 0:26:20 ago on Thu Jun 13 09:36:30 2024.
Package awscli-2-2.15.30-1.amzn2023.0.1.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-37-100 ~]$
```

```
[ec2-user@ip-172-31-37-100 ~]$ aws s3 ls
2024-06-13 09:55:53 priyabucket1203
[ec2-user@ip-172-31-37-100 ~]$
```

STEP 3 Removing bucket through commands

```
[ec2-user@ip-172-31-37-100 ~]$ aws s3 rb s3://priyabucket1203
remove_bucket: priyabucket1203
[ec2-user@ip-172-31-37-100 ~]$
```

STEP 4 open AWS S3 to check the bucket whether removed or not



After, completing the task terminate the instance