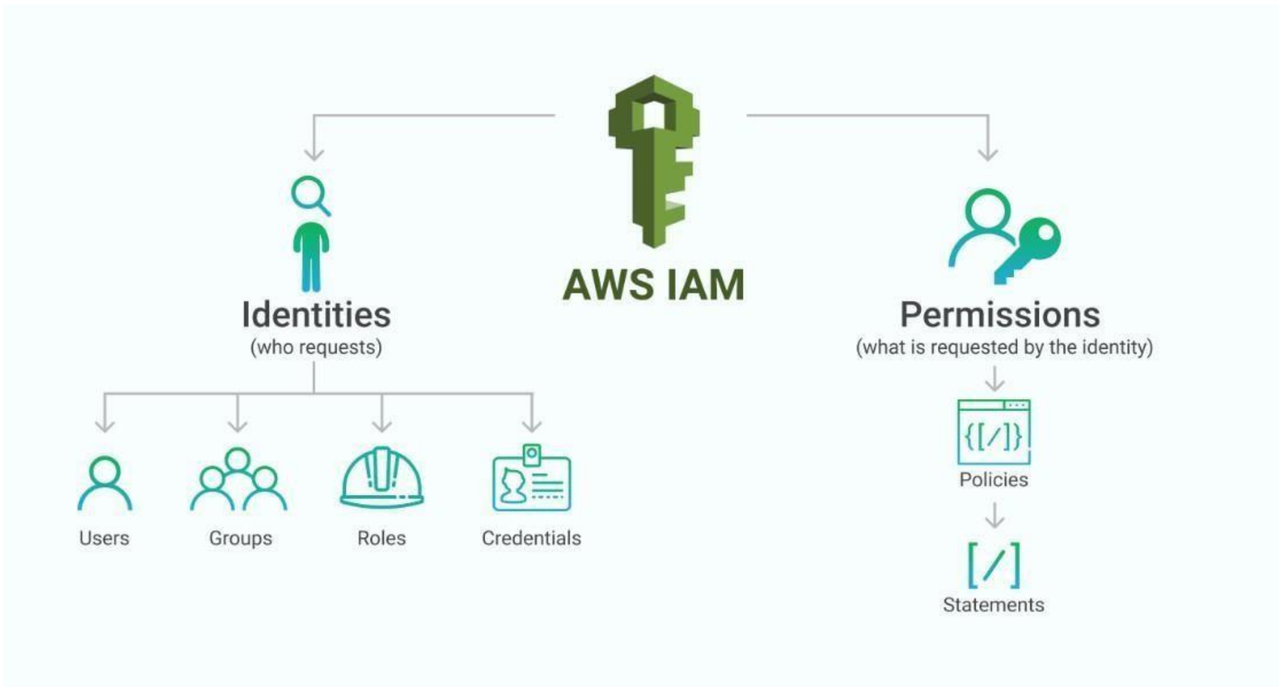
AWS Documentation

IAM (Identity Access Management):

Identity and Access Management (IAM) manages Amazon Web Services (AWS) users and their access to AWS accounts and services. It controls the level of access a user can have over an AWS account & set users, grant permission, and allows a user to use different features of an AWS account.



How IAM Works?

IAM verifies that a user or service has the necessary authorization to access a particular service in the AWS cloud. We can also use IAM to grant the right level of access to specific users, groups, or services. For example, we can use IAM to enable an EC2 instance to access S3 buckets by requesting fine-grained permissions.

IAM Users

- **What it is**: An IAM User is an individual person or service that interacts with AWS resources.

- **Example**: Imagine you have a developer named Alice who needs to manage EC2 instances. You create an IAM User called "Alice" and give her permissions to manage EC2 instances.

2. IAM Groups

- **What it is**: An IAM Group is a collection of IAM Users. Permissions assigned to a group are inherited by all users in that group.

- **Example**: You have a team of developers: Alice, Bob, and Carol. Instead of assigning permissions to each developer individually, you create an IAM Group called "Developers" and give it permissions to manage EC2 instances. Then, you add Alice, Bob, and Carol to the "Developers" group, so they all get the necessary permissions.

3. IAM Roles

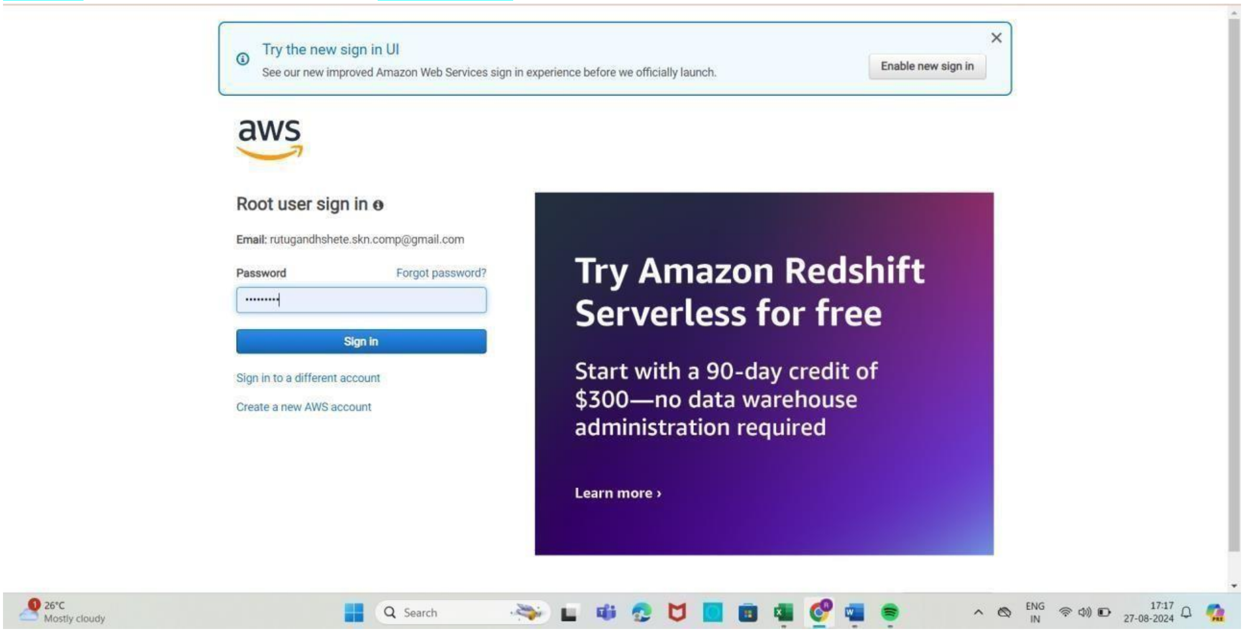
- **What it is**: An IAM Role is like a user, but instead of being assigned to a specific person, it's meant to be assumed by anyone who needs it (including AWS services).

- **Example**: You have an application running on an EC2 instance that needs to access an S3 bucket. Instead of embedding credentials in the application, you create an IAM Role called "S3AccessRole" that has permissions to access the S3 bucket. You then assign this role to the EC2 instance, so the application can securely access the S3 bucket without needing hardcoded credentials.

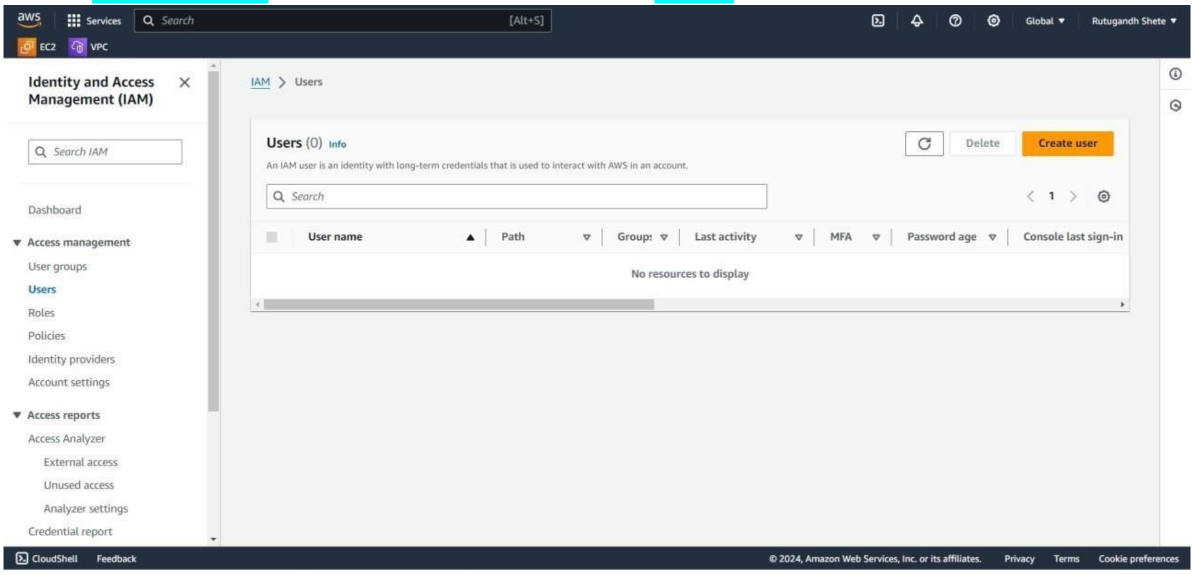
Launching instance using CLI:

Steps:

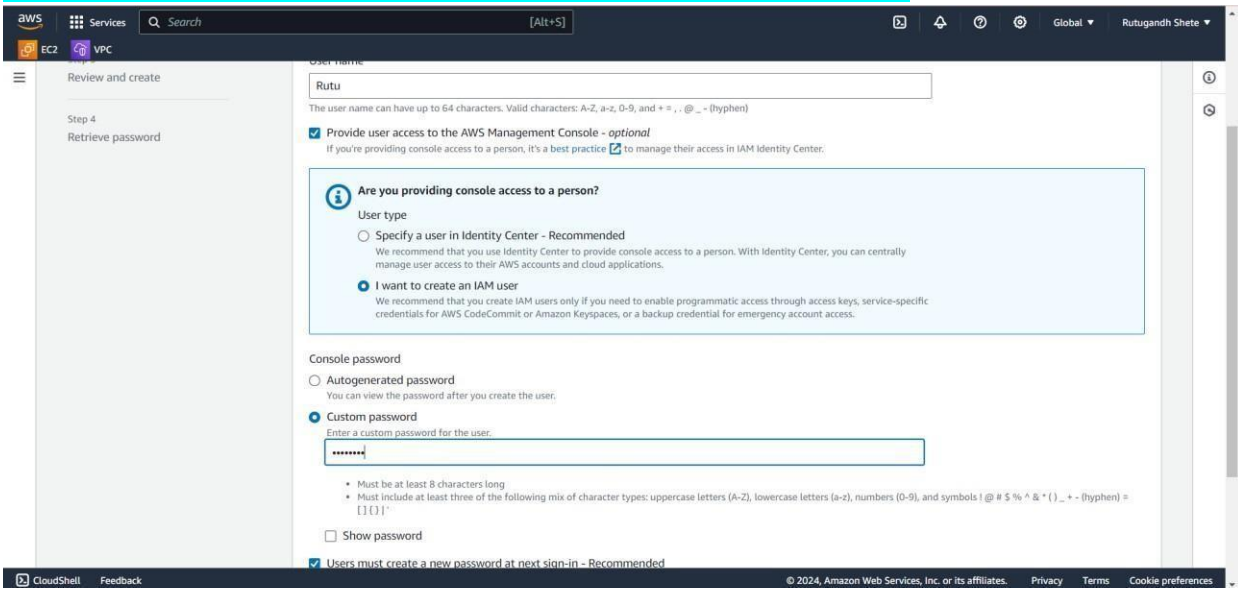
- Login AWS console using root user.


Try the new sign in UI
See our new improved Amazon Web Services sign in experience before we officially launch.
Enable new sign in
aws
Root user sign in
Email: rutugandhshete.skn.comp@gmail.com
Password
Forgot password?
Sign in
Sign in to a different account
Create a new AWS account
Try Amazon Redshift Serverless for free
Start with a 90-day credit of $300—no data warehouse administration required
Learn more ›

- Go to IAM service→ Access management→users



- Click on create user →give user name→click Provide user access to the AWS Management Console→click on I want to create IAM user →we can consider default password or we can create our passoword.



Review and create

Step 4
Retrieve password

Rutu

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.
••••••••

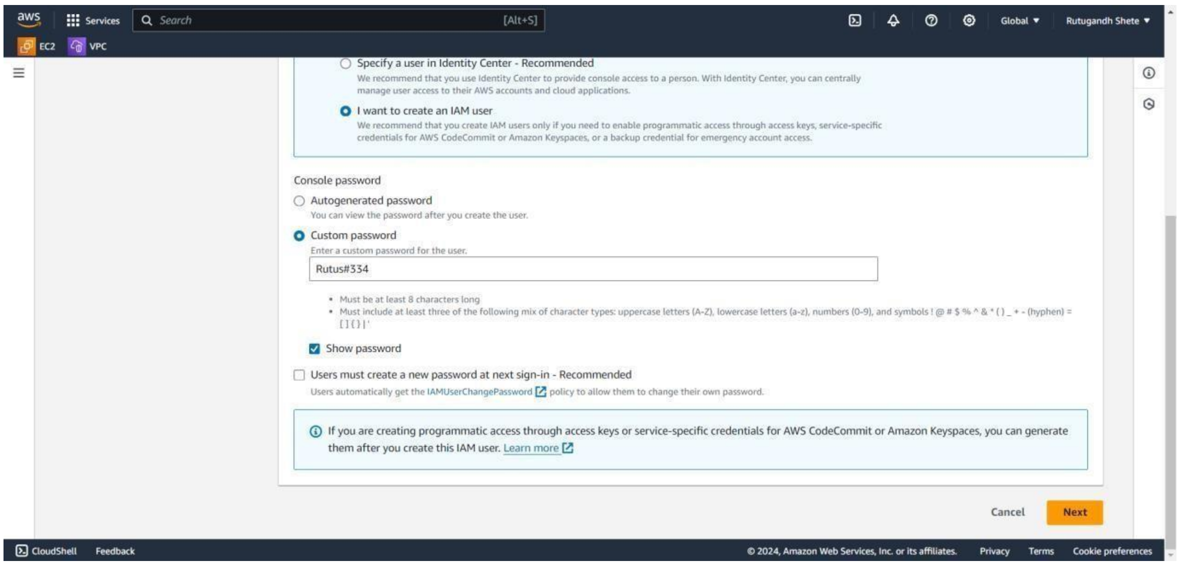
- Must be at least 8 characters long
- Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # $ % ^ & * ( ) _ + - (hyphen) = [ ] { } | '

[ ] Show password

[✓] Users must create a new password at next sign-in - Recommended

© 2024, Amazon Web Services, Inc. or its affiliates.

- If user want to create new password at the time of login then we can select that option.



( ) 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.
Rutus#334

- 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) = [ ] { } | '

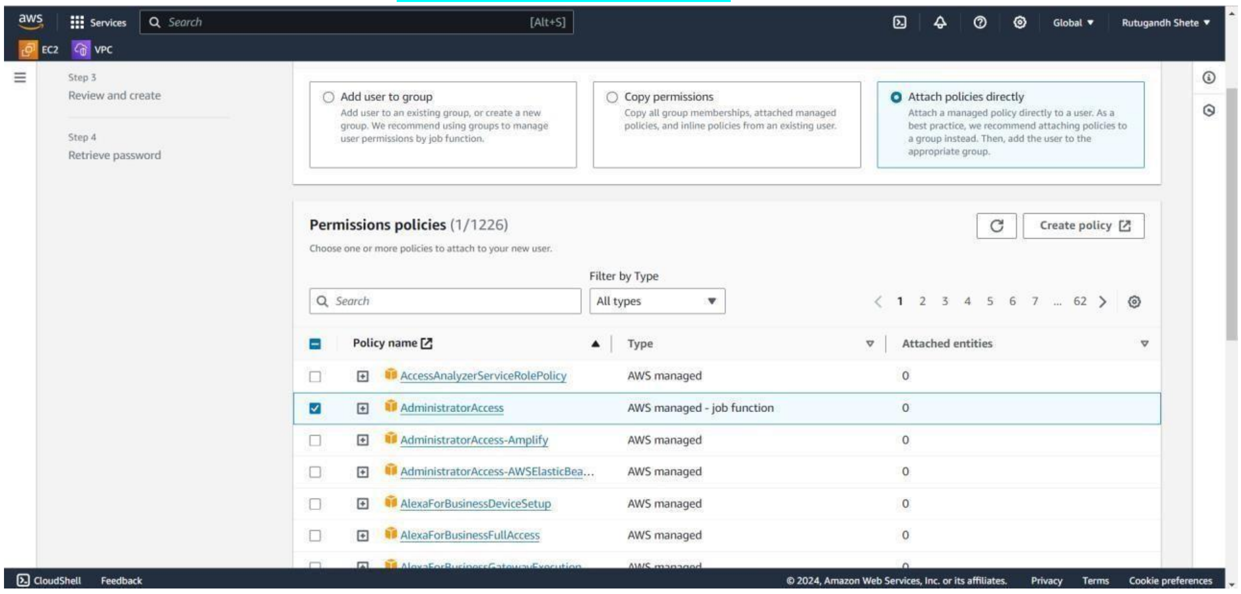
[✓] Show password

[ ] 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.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more

© 2024, Amazon Web Services, Inc. or its affiliates.

- Is set permissions→select attach policies directly→select policy



○ 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.

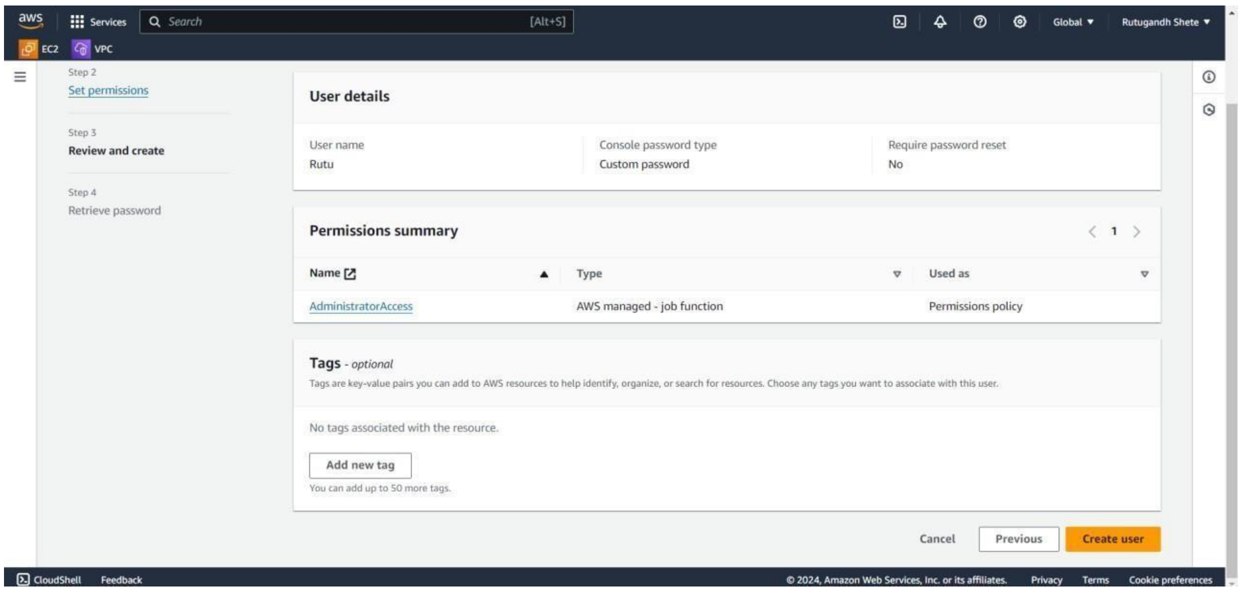
● Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1226)
Choose one or more policies to attach to your new user.

Filter by Type: All types

	Policy name	Type	Attached entities
☐	AccessAnalyzerServiceRolePolicy	AWS managed	0
☑	AdministratorAccess	AWS managed - job function	0
☐	AdministratorAccess-Amplify	AWS managed	0
☐	AdministratorAccess-AWSElasticBea...	AWS managed	0
☐	AlexaForBusinessDeviceSetup	AWS managed	0
☐	AlexaForBusinessFullAccess	AWS managed	0

- Review all the details and create user



User details

User name	Console password type	Require password reset
Rutu	Custom password	No

Permissions summary

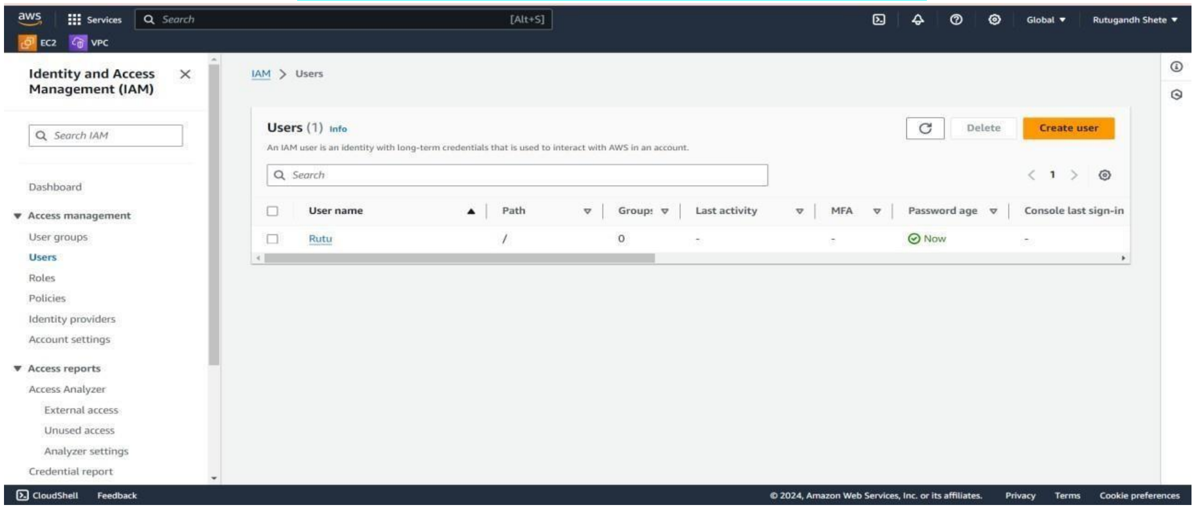
Name	Type	Used as
AdministratorAccess	AWS managed - job function	Permissions policy

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.

Add new tag
You can add up to 50 more tags.

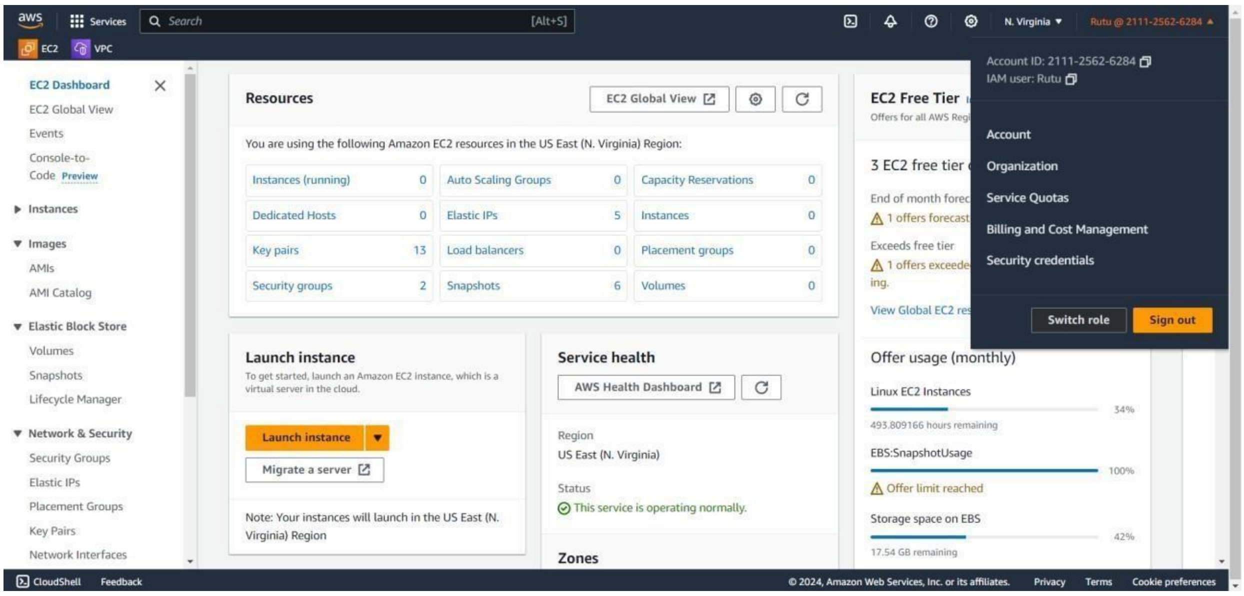
- After creating user. Copy ARN 12digit number(211125626284).



Users (1) Info
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

	User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in
☐	Rutu	/	0	-	-	✓ Now	-

- Sign in to AWS using IAM user.



Account ID: 2111-2562-6284
IAM user: Rutu

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	0	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	5	Instances	0
Key pairs	13	Load balancers	0	Placement groups	0
Security groups	2	Snapshots	6	Volumes	0

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the US East (N. Virginia) Region

Region
US East (N. Virginia)

Status
This service is operating normally.

Offers for all AWS Regi

3 EC2 free tier

End of month forec
1 offers forecast

Exceeds free tier
1 offers exceede ing.

View Global EC2 re

Offer usage (monthly)

Linux EC2 Instances 34%
493.809166 hours remaining

EBS:SnapshotUsage 100%
Offer limit reached

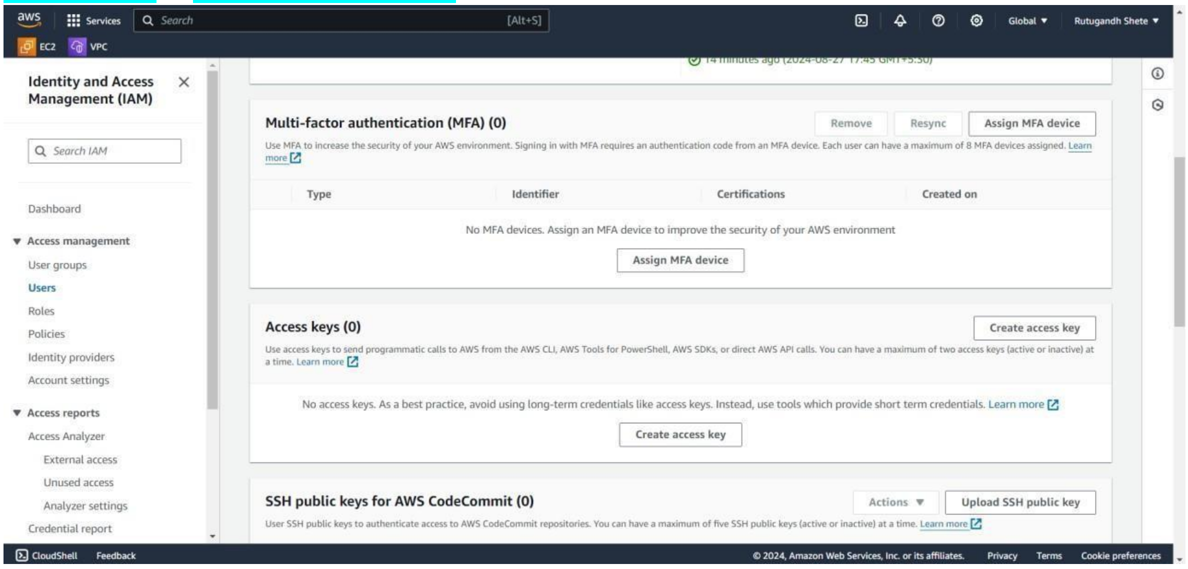
Storage space on EBS 42%
17.54 GB remaining

© 2024, Amazon Web Services, Inc. or its affiliates.

Launching instance using CLI:

Steps:

- Go to AWS root user→IAM user→click on User name→security credential→create access key



14 minutes ago (2024-08-27 17:45 GMT+5:30)

Multi-factor authentication (MFA) (0)

Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. Learn more

Type	Identifier	Certifications	Created on

No MFA devices. Assign an MFA device to improve the security of your AWS environment

Access keys (0)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more

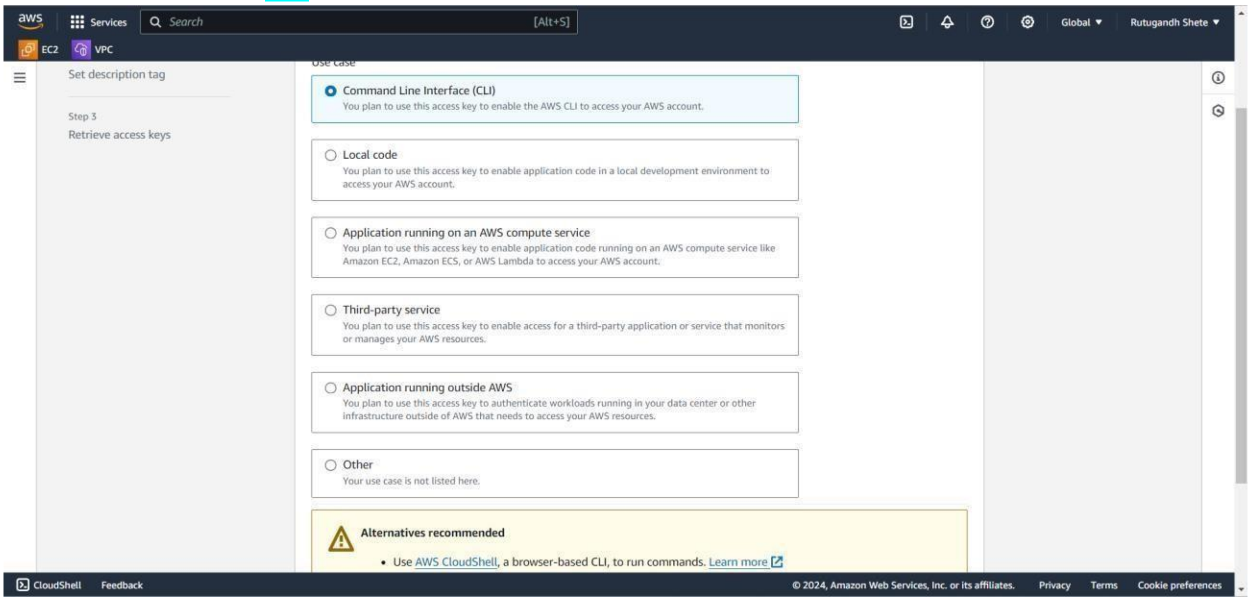
No access keys. As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. Learn more

SSH public keys for AWS CodeCommit (0)

User SSH public keys to authenticate access to AWS CodeCommit repositories. You can have a maximum of five SSH public keys (active or inactive) at a time. Learn more

© 2024, Amazon Web Services, Inc. or its affiliates.

- Select use case→CLI→next



Set description tag

Step 3
Retrieve access keys

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.

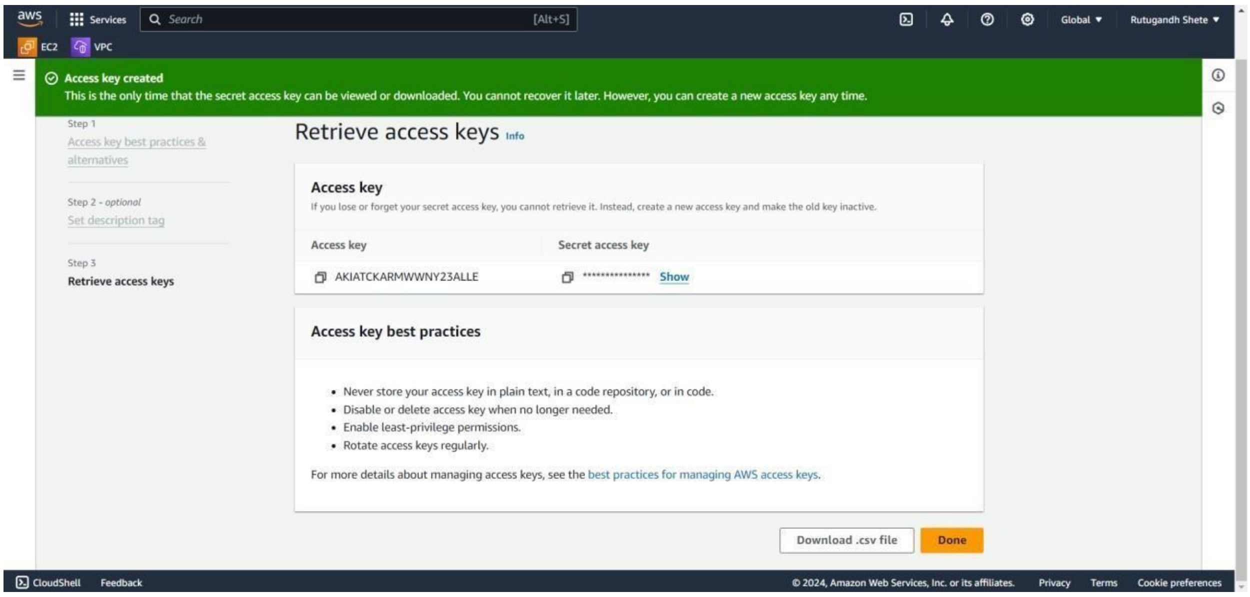
Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

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.

Alternatives recommended
- Use AWS CloudShell, a browser-based CLI, to run commands. Learn more

- After creating key



Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Step 1
Access key best practices & alternatives

Step 2 - optional
Set description tag

Step 3
Retrieve access keys

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 inactive.

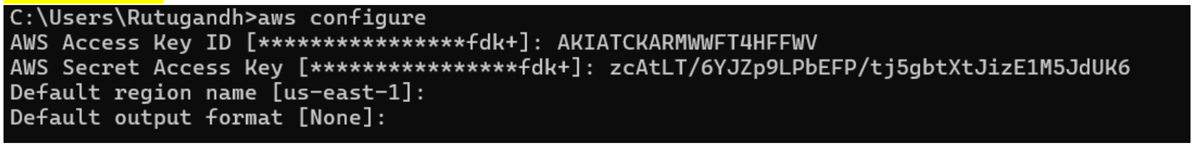
Access key	Secret access key
AKIATCKARMWWNY23ALLE	*************** Show

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.

For more details about managing access keys, see the best practices for managing AWS access keys.

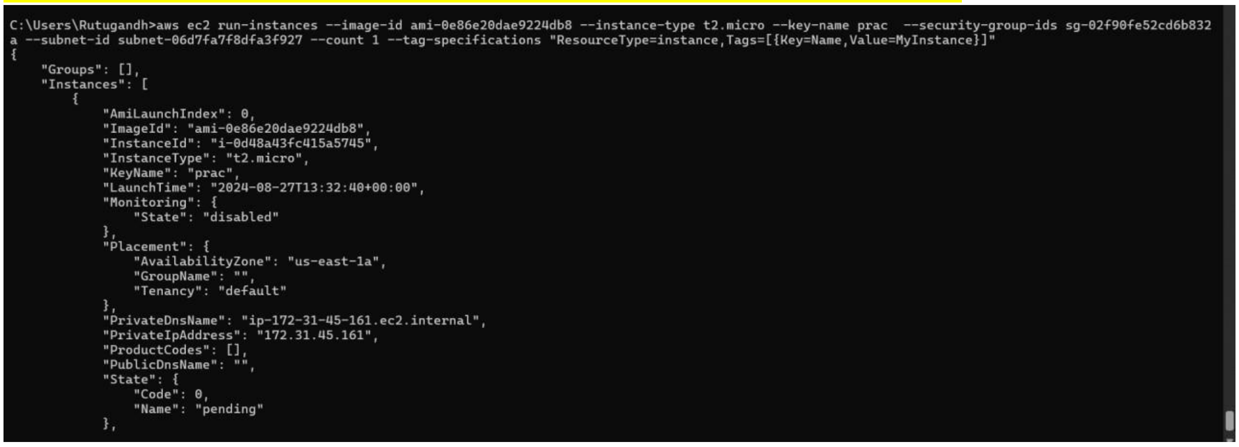
- Copy that access key→open CMD in machine and add command "aws configure"→paste that key in access and secret key place→region→.



```
C:\Users\Rutugandh>aws configure
AWS Access Key ID [****************fdk+]: AKIATCKARMWWFT4HFFWV
AWS Secret Access Key [****************fdk+]: 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
Default region name [us-east-1]:
Default output format [None]:
```

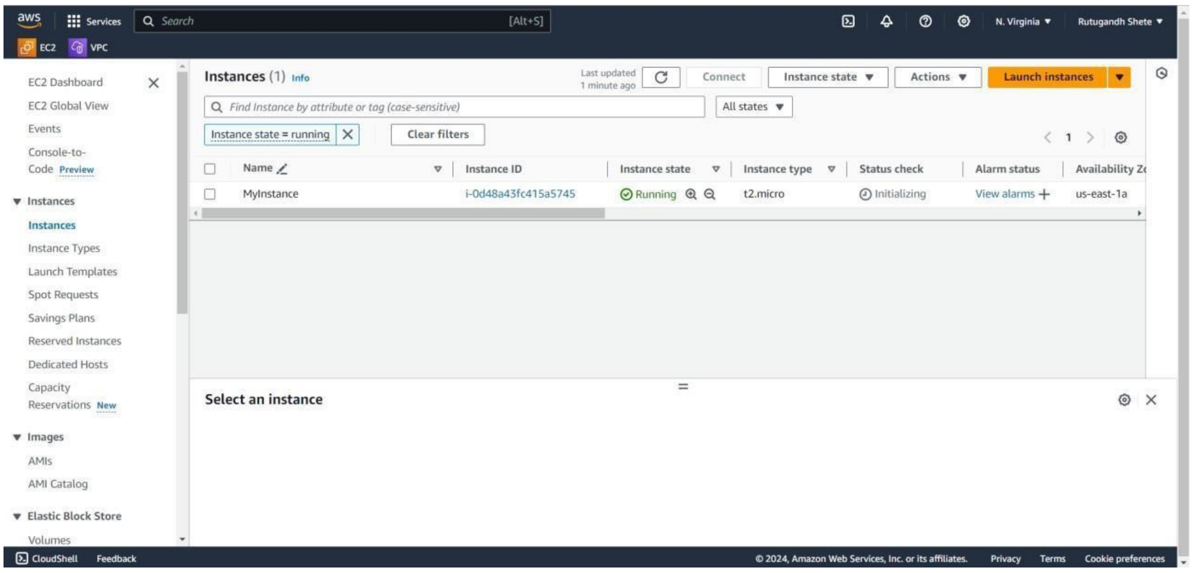
- This is one basic structure of launching instance

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
   --instance-type t2.micro \
   --key-name YourKeyName \
    --security-group-ids sg-12345678 \
    --subnet-id subnet-12345678 \
   --count 1 \
  --tag-specifications
'ResourceType=instance,Tags=[{Key=Name,Value=MyInstance}]'
```



```
C:\Users\Rutugandh>aws ec2 run-instances --image-id ami-0e86e20dae9224db8 --instance-type t2.micro --key-name prac  --security-group-ids sg-02f90fe52cd6b832
a --subnet-id subnet-06d7fa7f8dfa3f927 --count 1 --tag-specifications "ResourceType=instance,Tags=[{Key=Name,Value=MyInstance}]"
{
    "Groups": [],
    "Instances": [
        {
            "AmiLaunchIndex": 0,
            "ImageId": "ami-0e86e20dae9224db8",
            "InstanceId": "i-0d48a43fc415a5745",
            "InstanceType": "t2.micro",
            "KeyName": "prac",
            "LaunchTime": "2024-08-27T13:32:40+00:00",
            "Monitoring": {
                "State": "disabled"
            },
            "Placement": {
                "AvailabilityZone": "us-east-1a",
                "GroupName": "",
                "Tenancy": "default"
            },
            "PrivateDnsName": "ip-172-31-45-161.ec2.internal",
            "PrivateIpAddress": "172.31.45.161",
            "ProductCodes": [],
            "PublicDnsName": "",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
```

- One instance is launched.



Instances (1) Info

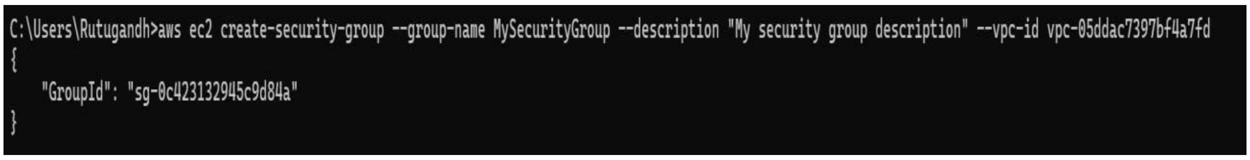
Instance state = running

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zo
☐	MyInstance	i-0d48a43fc415a5745	Running	t2.micro	Initializing	View alarms +	us-east-1a

Creating security groups using CLI:

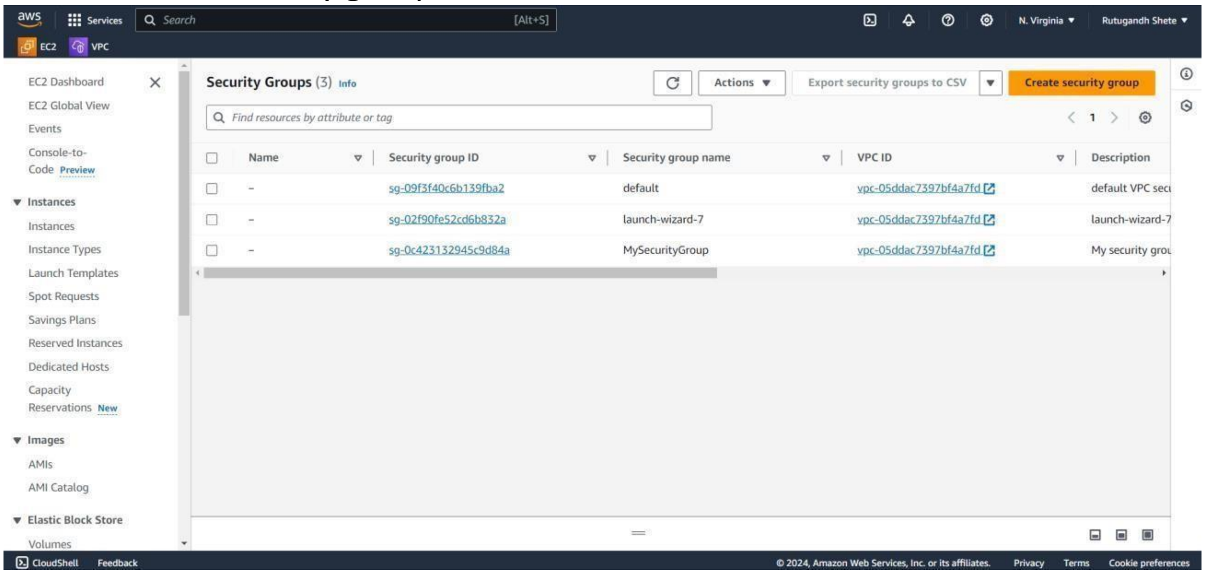
Steps:

- Go to CMD and type command "aws ec2 create-security-group --group-name MySecurityGroup --description "My security group description" --vpc-id vpc-05ddac7397bf4a7fd"



```
C:\Users\Rutugandh>aws ec2 create-security-group --group-name MySecurityGroup --description "My security group description" --vpc-id vpc-05ddac7397bf4a7fd
{
    "GroupId": "sg-0c423132945c9d84a"
}
```

- As we can see security group is created



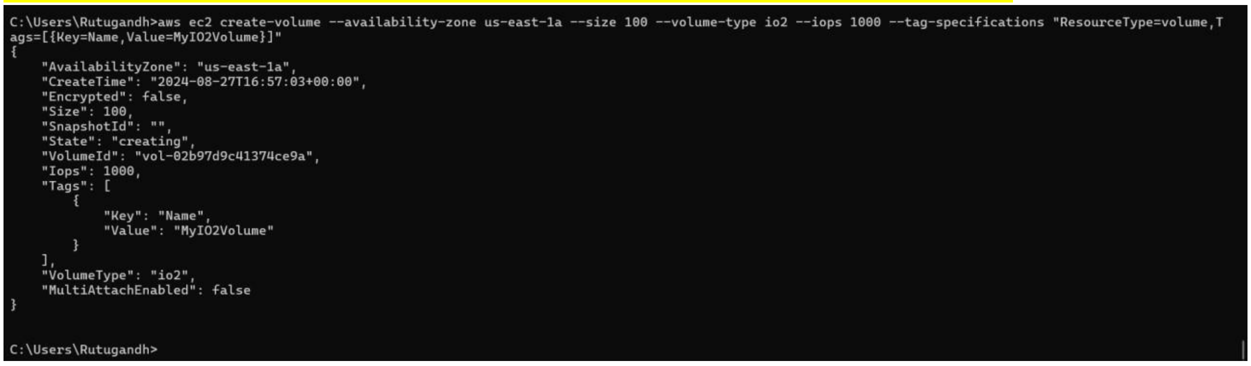
Security Groups (3)

	Name	Security group ID	Security group name	VPC ID	Description
☐	-	sg-09f3f40c6b139fba2	default	vpc-05ddac7397bf4a7fd	default VPC secu
☐	-	sg-02f90fe52cd6b832a	launch-wizard-7	vpc-05ddac7397bf4a7fd	launch-wizard-7
☐	-	sg-0c423132945c9d84a	MySecurityGroup	vpc-05ddac7397bf4a7fd	My security grou

Creating EBS using CLI:

Steps:

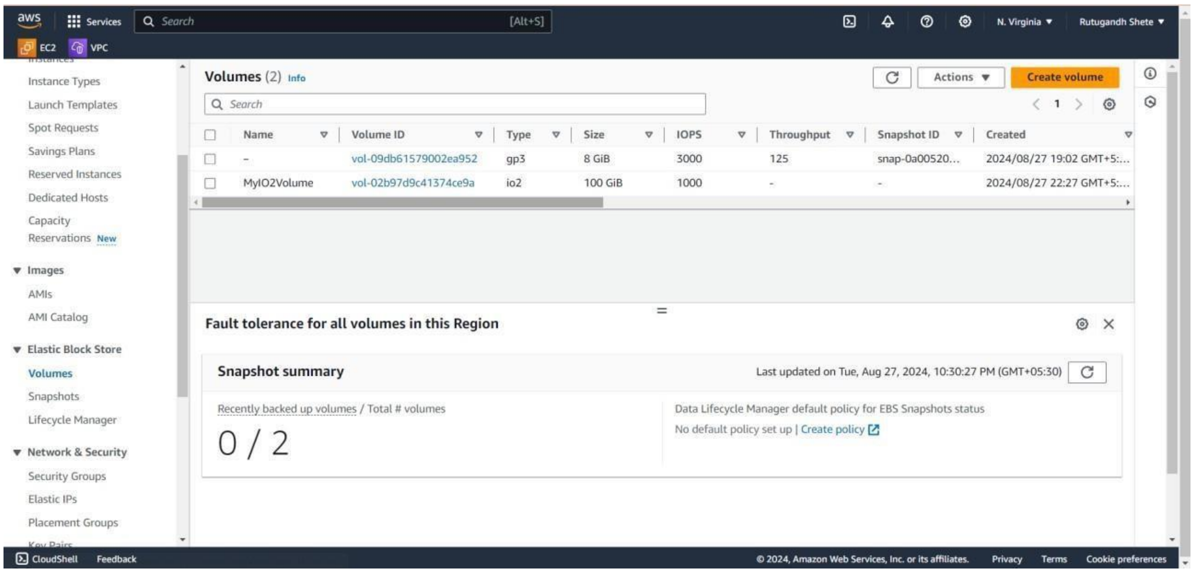
- In earlier step we have already launched one ec2 instance in this step we will attach one volume to ec2 instance.
- aws ec2 create-volume --availability-zone us-east-1a --size 100 --volume-type io2 --iops 1000 --tag-specifications "ResourceType=volume,Tags=[{Key=Name,Value=MyIO2Volume}]".



```
C:\Users\Rutugandh>aws ec2 create-volume --availability-zone us-east-1a --size 100 --volume-type io2 --iops 1000 --tag-specifications "ResourceType=volume,T
ags=[{Key=Name,Value=MyIO2Volume}]"
{
    "AvailabilityZone": "us-east-1a",
    "CreateTime": "2024-08-27T16:57:03+00:00",
    "Encrypted": false,
    "Size": 100,
    "SnapshotId": "",
    "State": "creating",
    "VolumeId": "vol-02b97d9c41374ce9a",
    "Iops": 1000,
    "Tags": [
        {
            "Key": "Name",
            "Value": "MyIO2Volume"
        }
    ],
    "VolumeType": "io2",
    "MultiAttachEnabled": false
}

C:\Users\Rutugandh>
```

- We can see that we have created one EBS volume on GUI.



Volumes (2) Info

	Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot ID	Created
☐	-	vol-09db61579002ea952	gp3	8 GiB	3000	125	snap-0a00520...	2024/08/27 19:02 GMT+5:...
☐	MyIO2Volume	vol-02b97d9c41374ce9a	io2	100 GiB	1000	-	-	2024/08/27 22:27 GMT+5:...

Fault tolerance for all volumes in this Region

Snapshot summary

Last updated on Tue, Aug 27, 2024, 10:30:27 PM (GMT+05:30)

Recently backed up volumes / Total # volumes

0 / 2

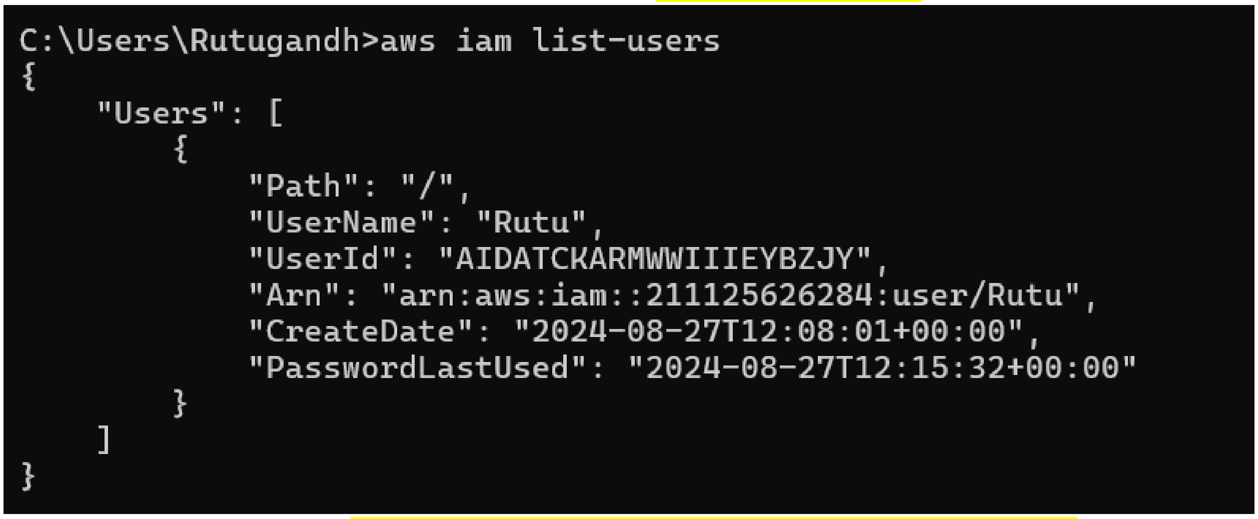
Data Lifecycle Manager default policy for EBS Snapshots status

No default policy set up | Create policy

Adding user using CLI:

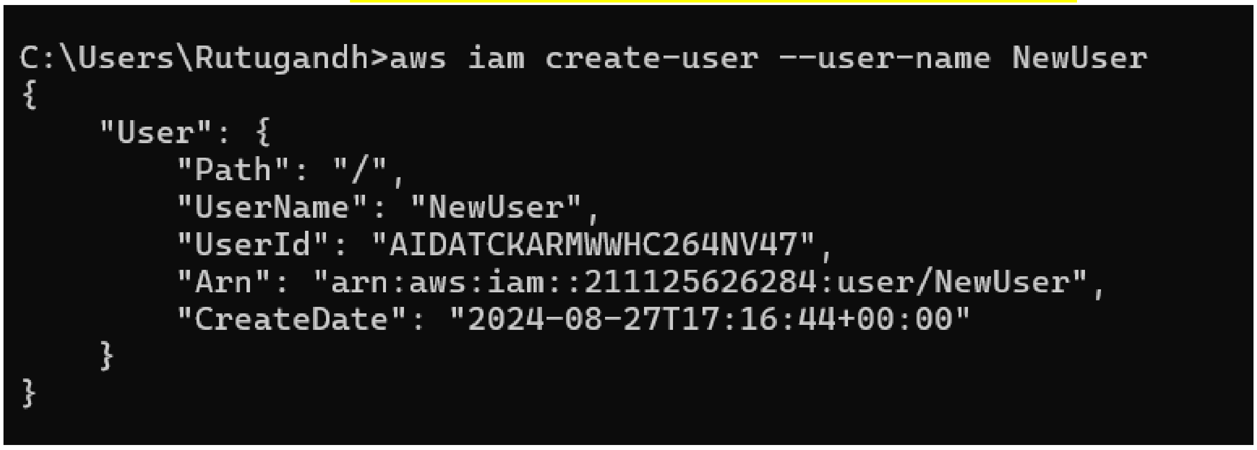
Steps:

- See all the IAM user that are created using “aws iam list-users”.



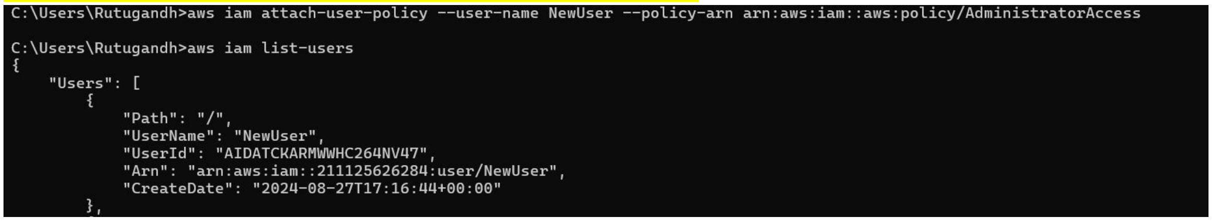
```
C:\Users\Rutugandh>aws iam list-users
{
    "Users": [
        {
            "Path": "/",
            "UserName": "Rutu",
            "UserId": "AIDATCKARMWWIIIEYBZJY",
            "Arn": "arn:aws:iam::211125626284:user/Rutu",
            "CreateDate": "2024-08-27T12:08:01+00:00",
            "PasswordLastUsed": "2024-08-27T12:15:32+00:00"
        }
    ]
}
```

- create new user using “aws iam create-user --user-name NewUserName”.



```
C:\Users\Rutugandh>aws iam create-user --user-name NewUser
{
    "User": {
        "Path": "/",
        "UserName": "NewUser",
        "UserId": "AIDATCKARMWWHC264NV47",
        "Arn": "arn:aws:iam::211125626284:user/NewUser",
        "CreateDate": "2024-08-27T17:16:44+00:00"
    }
}
```

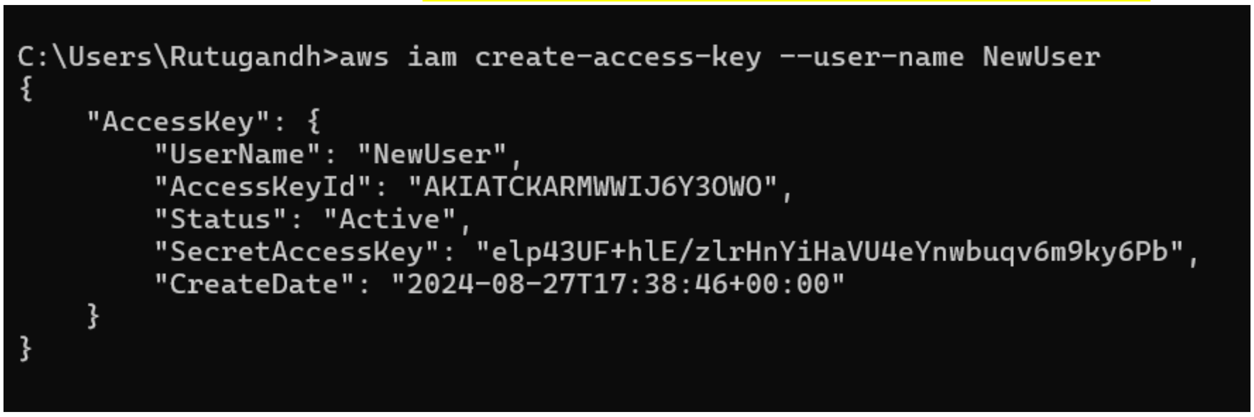
- Then we have ARN for every policy copy this “arn:aws:iam::aws:policy/AdministratorAccess” paste in ARN part ” aws iam attach-user-policy --user-name NewUser --policy-arn arn:aws:iam::aws:policy/AdministratorAccess”



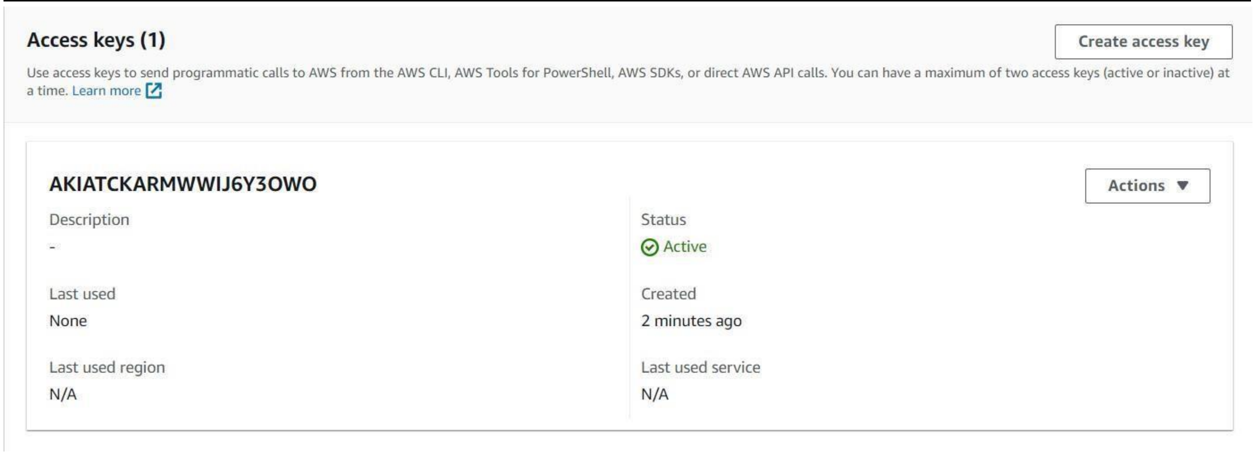
```
C:\Users\Rutugandh>aws iam attach-user-policy --user-name NewUser --policy-arn arn:aws:iam::aws:policy/AdministratorAccess

C:\Users\Rutugandh>aws iam list-users
{
    "Users": [
        {
            "Path": "/",
            "UserName": "NewUser",
            "UserId": "AIDATCKARMWWHC264NV47",
            "Arn": "arn:aws:iam::211125626284:user/NewUser",
            "CreateDate": "2024-08-27T17:16:44+00:00"
        },
```

- To create secret access key “aws iam create-access-key --user-name NewUser”



```
C:\Users\Rutugandh>aws iam create-access-key --user-name NewUser
{
    "AccessKey": {
        "UserName": "NewUser",
        "AccessKeyId": "AKIATCKARMWWIJ6Y3OWO",
        "Status": "Active",
        "SecretAccessKey": "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",
        "CreateDate": "2024-08-27T17:38:46+00:00"
    }
}
```



Access keys (1)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more

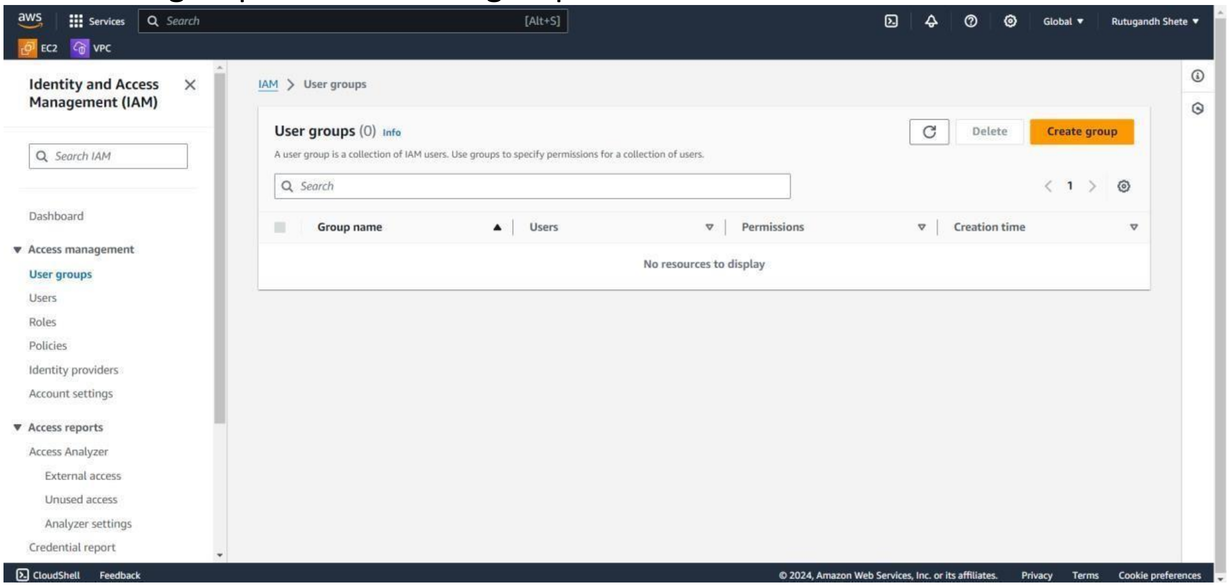
AKIATCKARMWWIJ6Y3OWO

Description -	Status Active
Last used None	Created 2 minutes ago
Last used region N/A	Last used service N/A

User Groups:

Steps:

- Go to user groups→click create group.

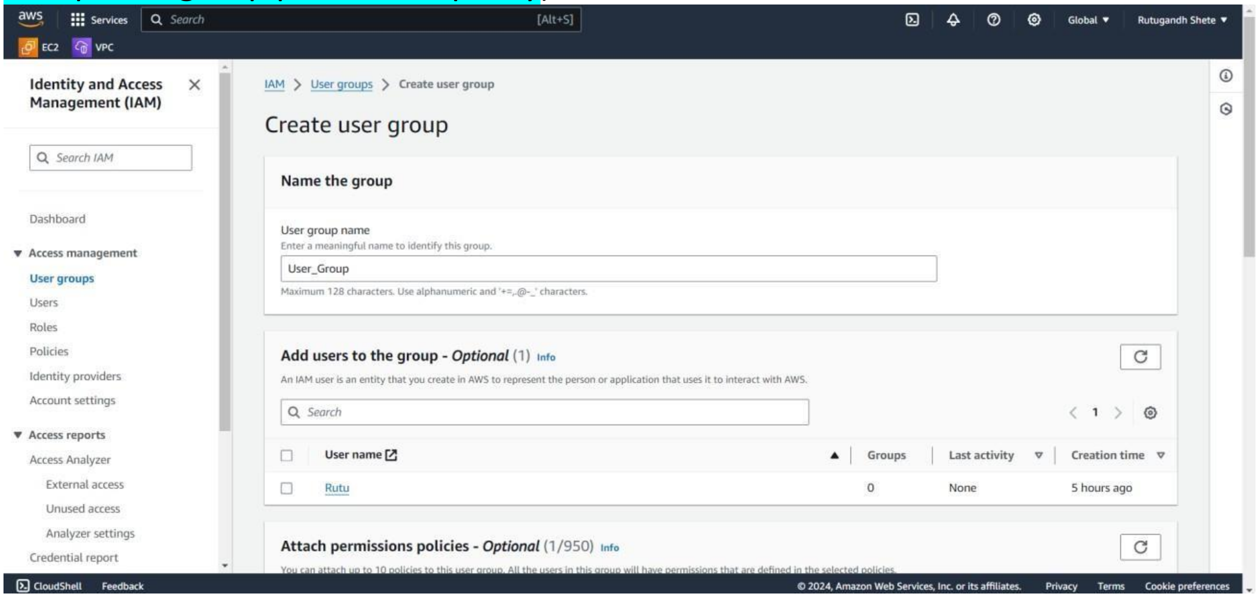


User groups (0) Info

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
No resources to display			

- Add name→we can add user to group(if user is added then default permissions will as equal to group permission policy)



IAM > User groups > Create user group

Create user group

Name the group

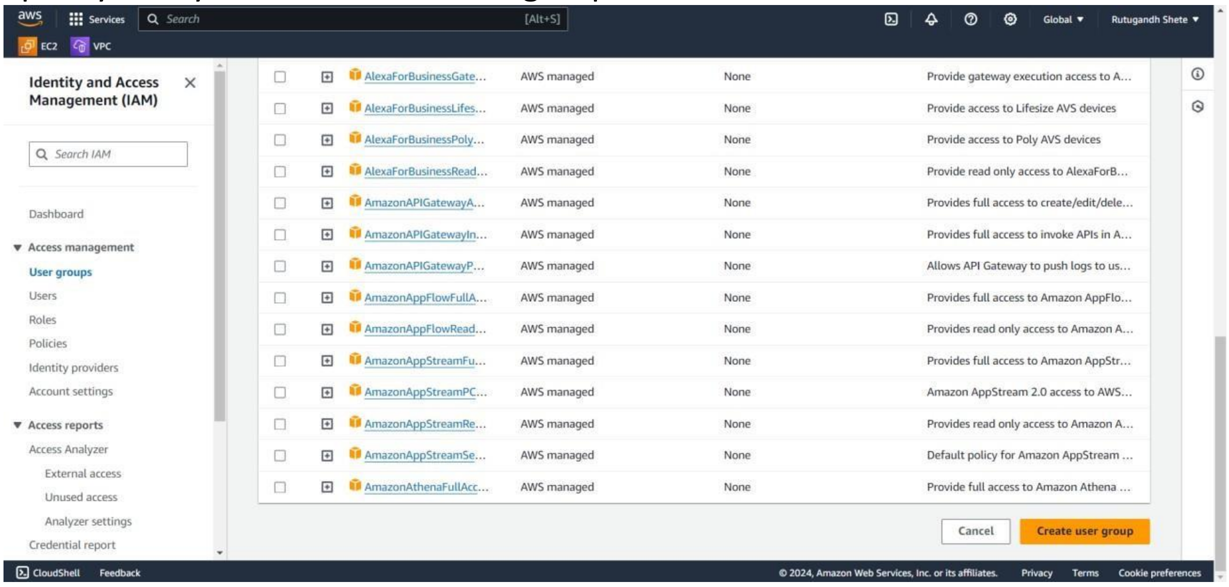
User group name
Enter a meaningful name to identify this group.
User_Group
Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

Add users to the group - *Optional* (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.

	User name	Groups	Last activity	Creation time
☐	Rutu	0	None	5 hours ago

Attach permissions policies - *Optional* (1/950) 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.

- Specify Policy name →create user group

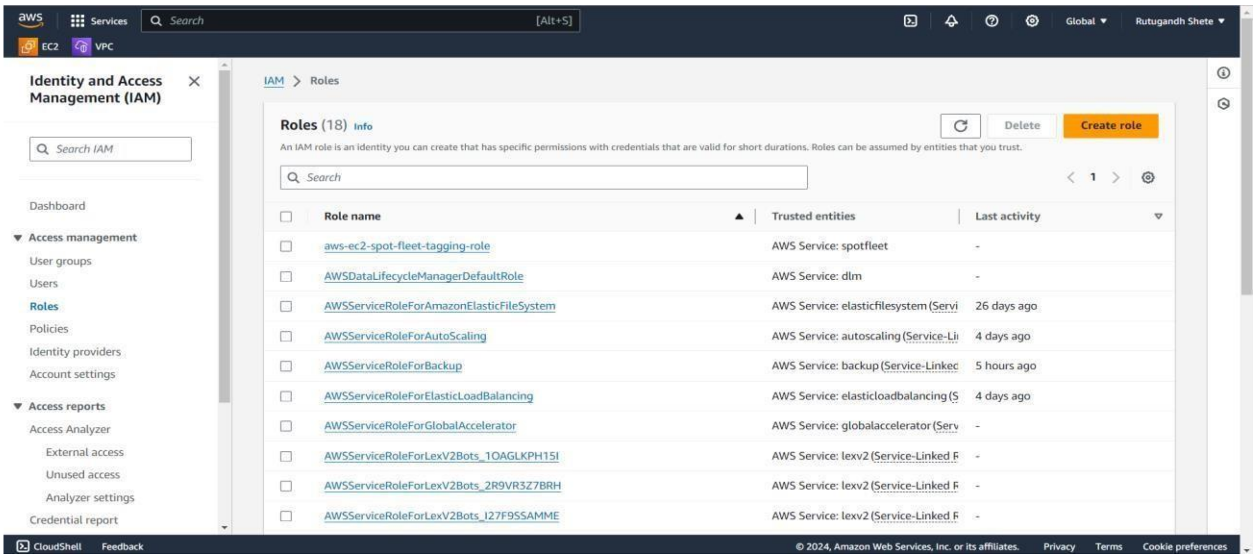


☐	⊞	AlexaForBusinessGate...	AWS managed	None	Provide gateway execution access to A...
☐	⊞	AlexaForBusinessLifes...	AWS managed	None	Provide access to Lifesize AVS devices
☐	⊞	AlexaForBusinessPoly...	AWS managed	None	Provide access to Poly AVS devices
☐	⊞	AlexaForBusinessRead...	AWS managed	None	Provide read only access to AlexaForB...
☐	⊞	AmazonAPIGatewayA...	AWS managed	None	Provides full access to create/edit/dele...
☐	⊞	AmazonAPIGatewayIn...	AWS managed	None	Provides full access to invoke APIs in A...
☐	⊞	AmazonAPIGatewayP...	AWS managed	None	Allows API Gateway to push logs to us...
☐	⊞	AmazonAppFlowFullA...	AWS managed	None	Provides full access to Amazon AppFlo...
☐	⊞	AmazonAppFlowRead...	AWS managed	None	Provides read only access to Amazon A...
☐	⊞	AmazonAppStreamFu...	AWS managed	None	Provides full access to Amazon AppStr...
☐	⊞	AmazonAppStreamPC...	AWS managed	None	Amazon AppStream 2.0 access to AWS...
☐	⊞	AmazonAppStreamRe...	AWS managed	None	Provides read only access to Amazon A...
☐	⊞	AmazonAppStreamSe...	AWS managed	None	Default policy for Amazon AppStream ...
☐	⊞	AmazonAthenaFullAcc...	AWS managed	None	Provide full access to Amazon Athena ...

Role:

Steps:

- Go to Role →click on create role



Identity and Access Management (IAM)

IAM > Roles

Roles (18) 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.

Role name	Trusted entities	Last activity
aws-ec2-spot-fleet-tagging-role	AWS Service: spotfleet	-
AWSDataLifecycleManagerDefaultRole	AWS Service: dlm	-
AWSServiceRoleForAmazonElasticFileSystem	AWS Service: elasticfilesystem (Servi	26 days ago
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Li	4 days ago
AWSServiceRoleForBackup	AWS Service: backup (Service-Linked	5 hours ago
AWSServiceRoleForElasticLoadBalancing	AWS Service: elasticloadbalancing (S	4 days ago
AWSServiceRoleForGlobalAccelerator	AWS Service: globalaccelerator (Serv	-
AWSServiceRoleForLexV2Bots_1OAGLKPH15I	AWS Service: lexv2 (Service-Linked F	-
AWSServiceRoleForLexV2Bots_2R9VR3Z7BRH	AWS Service: lexv2 (Service-Linked F	-
AWSServiceRoleForLexV2Bots_I27F9SSAMME	AWS Service: lexv2 (Service-Linked F	-

© 2024, Amazon Web Services, Inc. or its affiliates.

- Select trusted entity→

  AWS Service: This option allows you to grant permissions to AWS services like EC2, Lambda, or S3 to perform actions on your behalf. For example, if you want an EC2 instance to access an S3 bucket, you would create a role with S3 permissions and specify "AWS Service" (EC2) as the trusted entity.

  Another AWS Account:

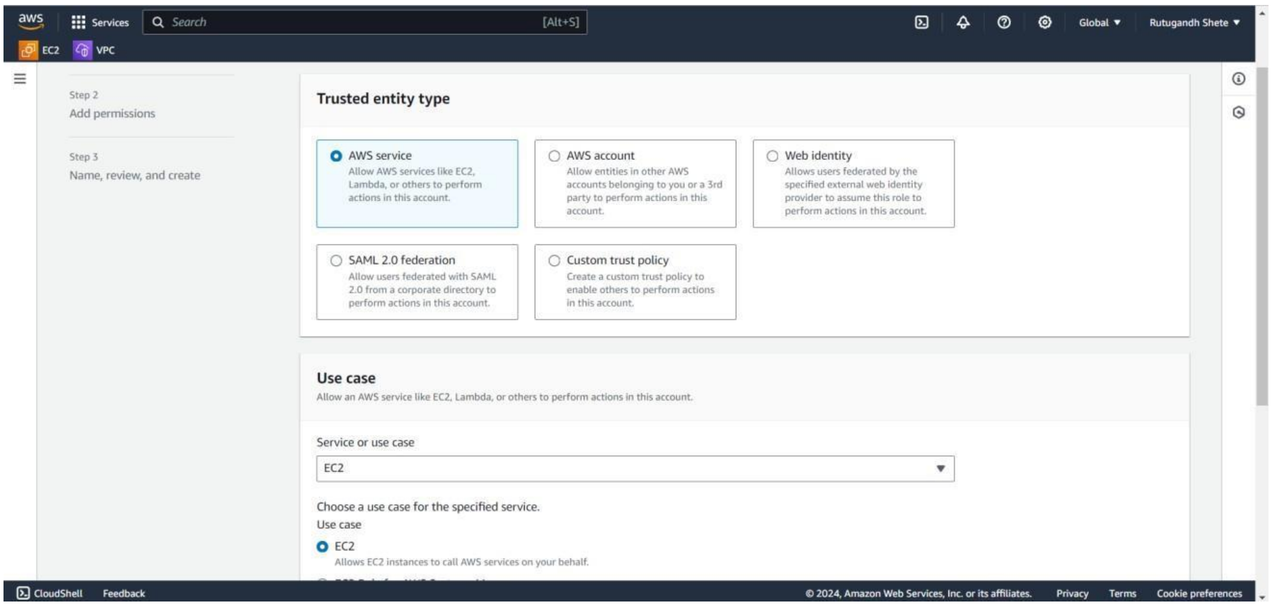
  This allows an IAM user or role in a different AWS account to assume the role. This is useful in scenarios where multiple AWS accounts need to share resources securely.

  Web Identity:

  his allows external identity providers (like Google, Facebook, or Amazon Cognito) to grant temporary security credentials to users, enabling them to access AWS resources. It's commonly used in mobile or web applications that use social sign-in.

  SAML 2.0 Federation:

  This allows your role to be assumed by users from your enterprise identity provider (IdP) using SAML (Security Assertion Markup Language). It's used for single sign-on (SSO) scenarios where users from your corporate directory (like Active Directory) need access to AWS resources→select use case (You have a web application running on an EC2 instance that processes images. After processing, the images need to be uploaded to an S3 bucket for storage and later retrieval.)



Step 2
Add permissions

Step 3
Name, review, and create

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.

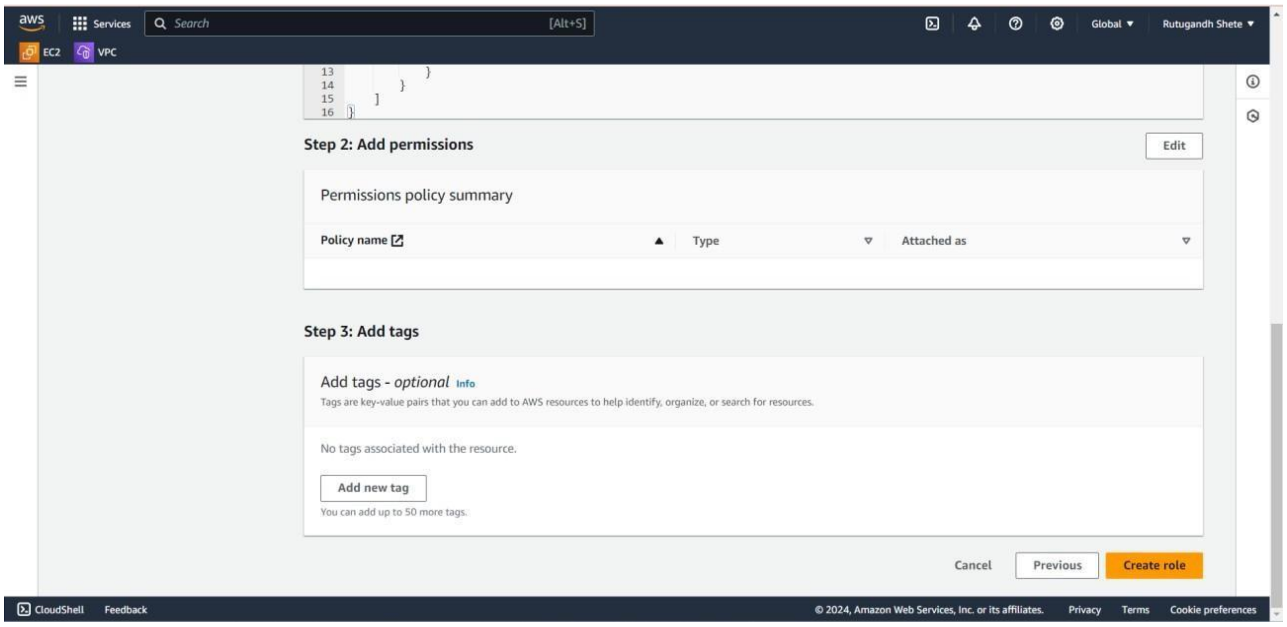
Service or use case: EC2

Choose a use case for the specified service.

Use case

[✓ EC2] Allows EC2 instances to call AWS services on your behalf.

- Provide name →create role



```
            }
        }
    ]
}
```

Step 2: Add permissions Edit

Permissions policy summary

Policy name	Type	Attached as

Step 3: Add tags

Add tags - *optional* Info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

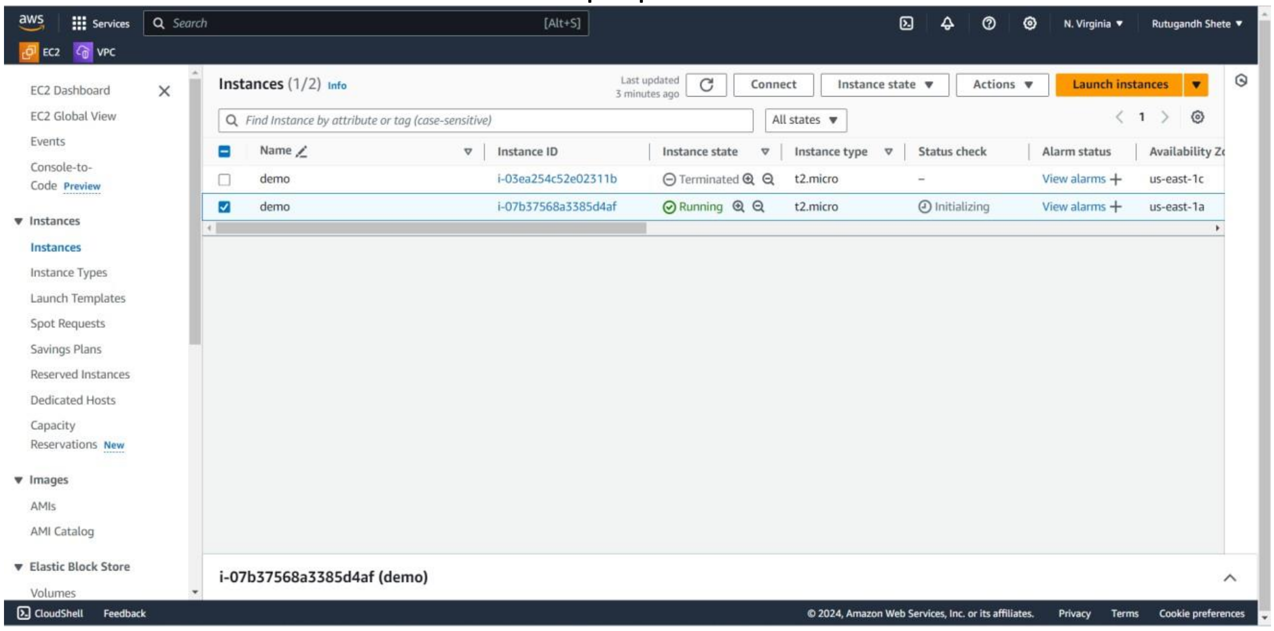
No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel Previous Create role

- Create Ec2 Amazon instance for demo purpose



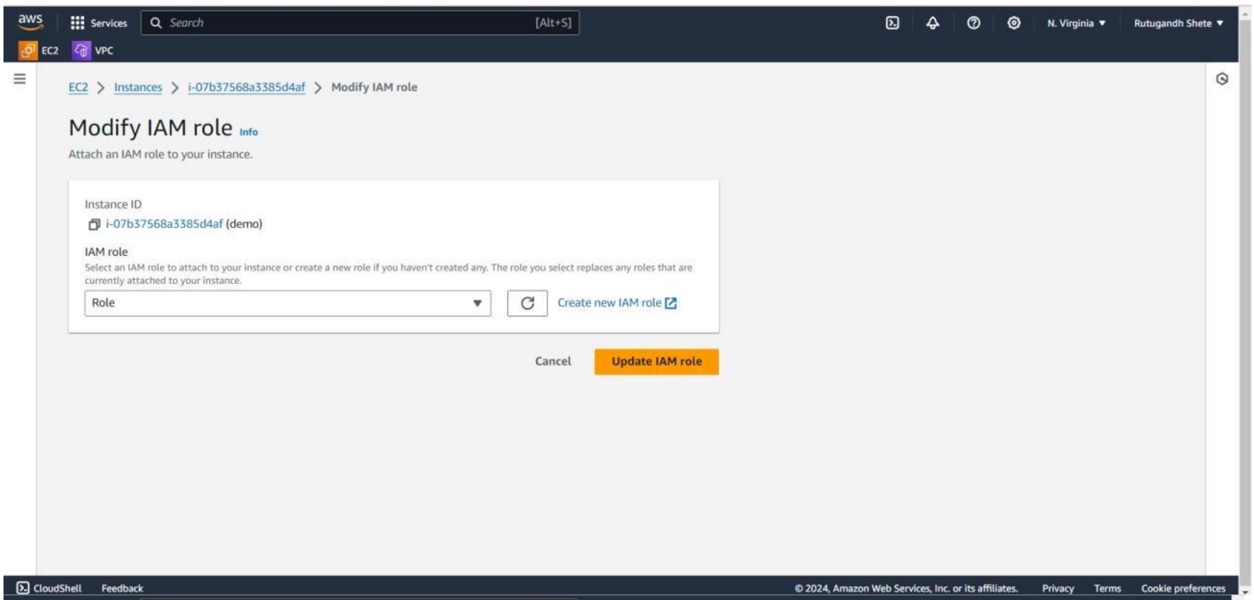
Instances (1/2) Info — Last updated 3 minutes ago — Connect — Instance state — Actions — Launch instances

Find Instance by attribute or tag (case-sensitive) — All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☐	demo	i-03ea254c52e02311b	Terminated	t2.micro	–	View alarms +	us-east-1c
☑	demo	i-07b37568a3385d4af	Running	t2.micro	Initializing	View alarms +	us-east-1a

i-07b37568a3385d4af (demo)

- Ec2 instance→actions→security→modify IAM role→choose IAM role that we have created →update.



EC2 > Instances > i-07b37568a3385d4af > Modify IAM role

Modify IAM role Info

Attach an IAM role to your instance.

Instance ID
i-07b37568a3385d4af (demo)

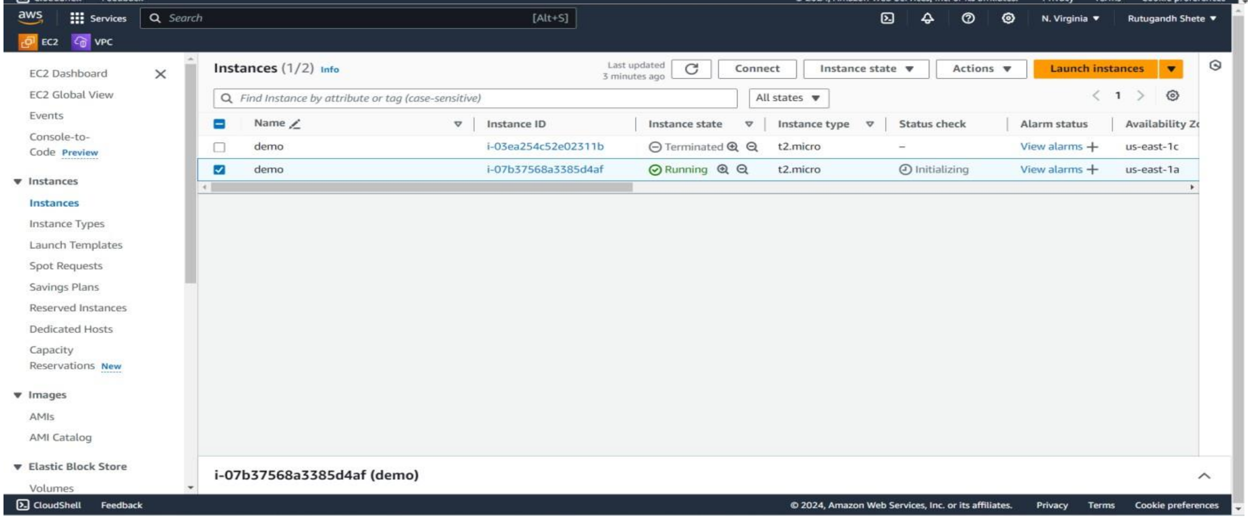
IAM role
Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

Role

Create new IAM role

Cancel   Update IAM role

© 2024, Amazon Web Services, Inc. or its affiliates.

- 

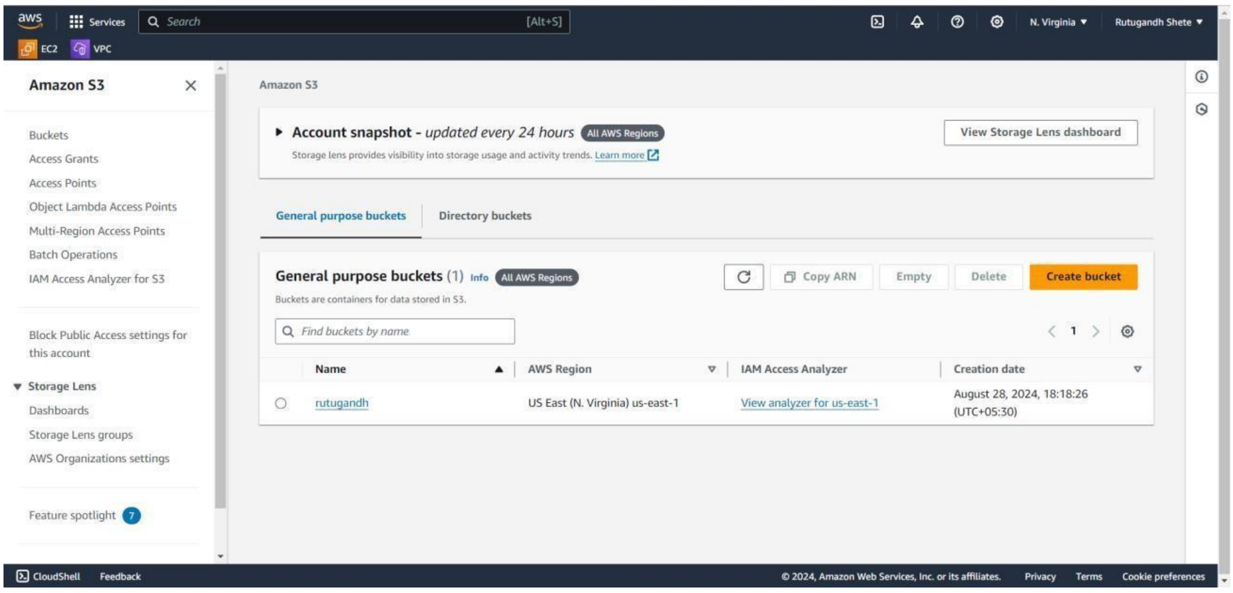
Instances (1/2) Info   Last updated 3 minutes ago   Connect   Instance state   Actions   Launch instances

Find Instance by attribute or tag (case-sensitive)   All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zo
☐	demo	i-03ea254c52e02311b	Terminated	t2.micro	–	View alarms +	us-east-1c
☑	demo	i-07b37568a3385d4af	Running	t2.micro	Initializing	View alarms +	us-east-1a

i-07b37568a3385d4af (demo)

- Go to s3 bucket service and create bucket.



Amazon S3

Account snapshot - *updated every 24 hours* All AWS Regions   View Storage Lens dashboard

Storage lens provides visibility into storage usage and activity trends. Learn more

General purpose buckets   Directory buckets

General purpose buckets (1) Info All AWS Regions   Copy ARN   Empty   Delete   Create bucket

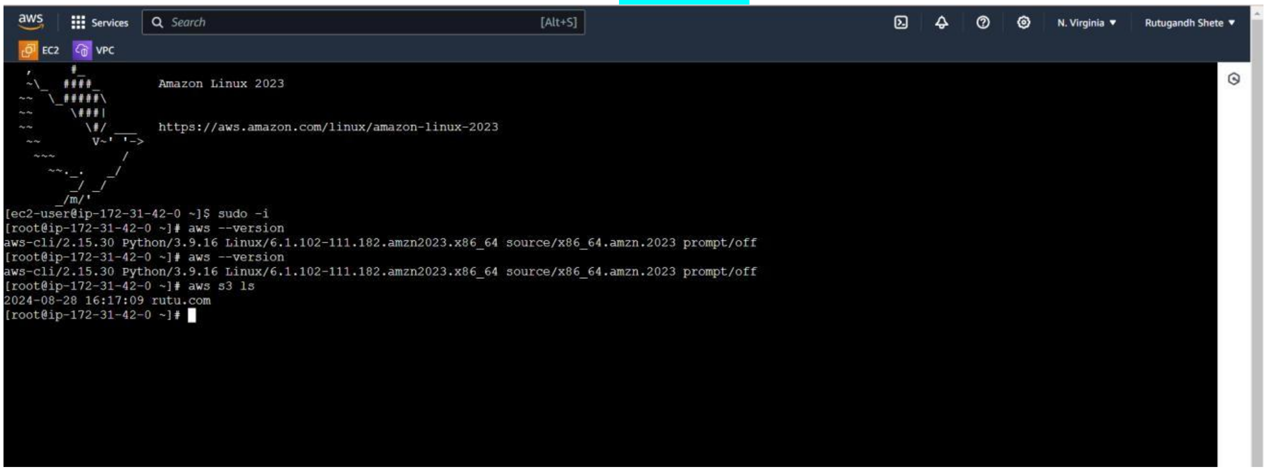
Buckets are containers for data stored in S3.

Find buckets by name

	Name	AWS Region	IAM Access Analyzer	Creation date
○	rutugandh	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 28, 2024, 18:18:26 (UTC+05:30)

Buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 7

- Connect to terminal and add command “aws s3 ls”.

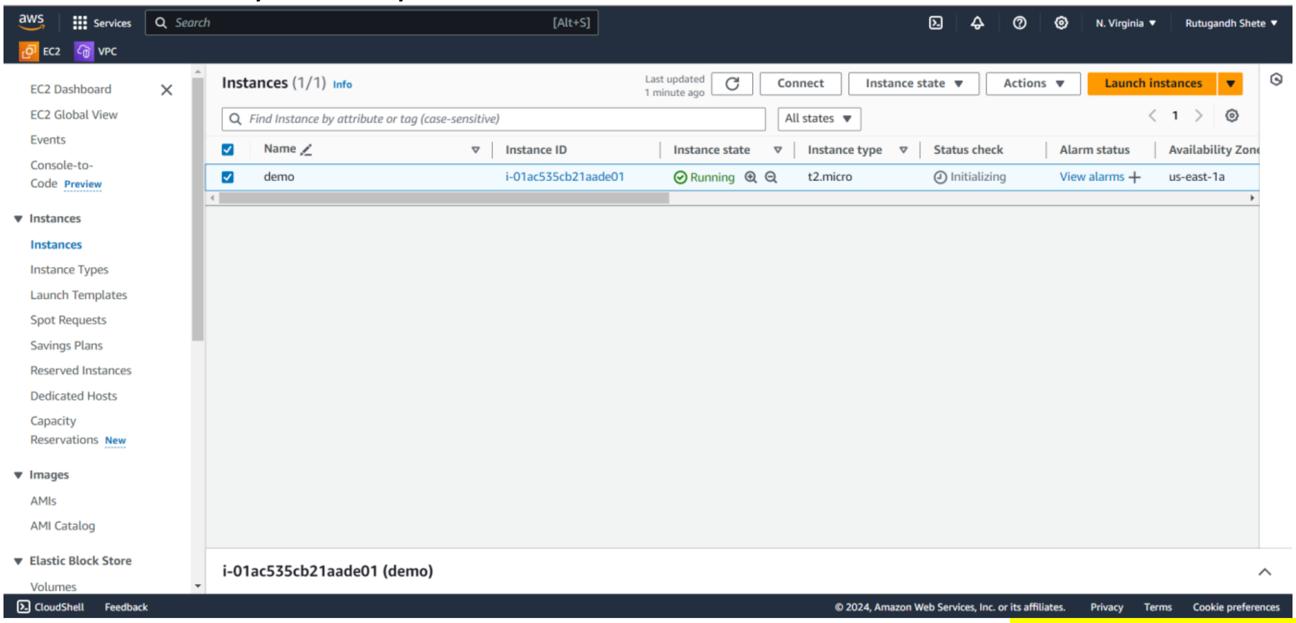


```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-42-0 ~]$ sudo -i
[root@ip-172-31-42-0 ~]# aws --version
aws-cli/2.15.30 Python/3.9.16 Linux/6.1.102-111.182.amzn2023.x86_64 source/x86_64.amzn.2023 prompt/off
[root@ip-172-31-42-0 ~]# aws --version
aws-cli/2.15.30 Python/3.9.16 Linux/6.1.102-111.182.amzn2023.x86_64 source/x86_64.amzn.2023 prompt/off
[root@ip-172-31-42-0 ~]# aws s3 ls
2024-08-28 16:17:09 rutu.com
[root@ip-172-31-42-0 ~]# 
```

Creating role using ubuntu linux machine:

Steps:

- Launch one ubuntu instance and modify IAM Role by going to actions→security→modify IAM role.



Instances (1/1)

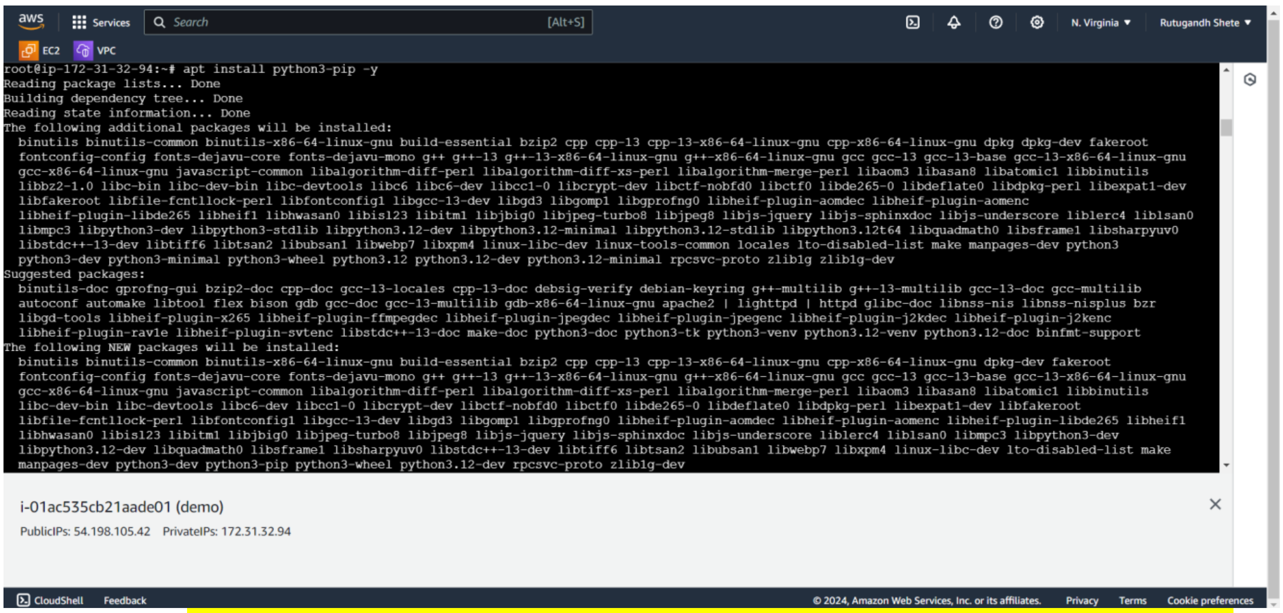
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
demo	i-01ac535cb21aade01	Running	t2.micro	Initializing	View alarms	us-east-1a

i-01ac535cb21aade01 (demo)

- connect it to terminal →update the system “apt update” →install “apt install python3-pip”.

  Package Management: pip is a tool that allows you to install and manage additional libraries and dependencies that are not part of the Python standard library. Many Python projects use pip to install necessary packages.

  Python Environment Setup: If you’re working with Python projects, especially in development or data science, pip is essential for installing packages like requests, flask, numpy, and many others.

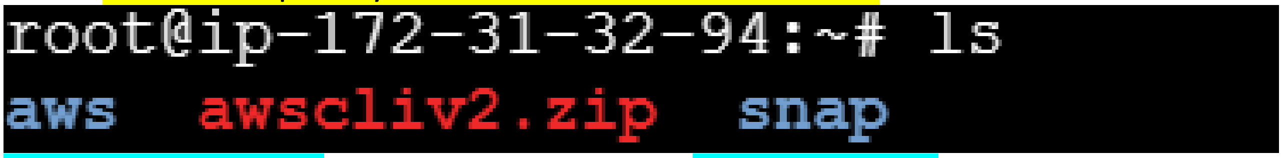


```
root@ip-172-31-32-94:~# apt install python3-pip -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential bzip2 cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu dpkg dpkg-dev fakeroot
  fontconfig-config fonts-dejavu-core fonts-dejavu-mono g++ g++-13 g++-13-x86-64-linux-gnu g++-x86-64-linux-gnu gcc gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu
  gcc-x86-64-linux-gnu javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libaom3 libasan8 libatomic1 libbinutils
  libbz2-1.0 libc-bin libc-dev-bin libc-devtools libc6 libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libde265-0 libdeflate0 libdpkg-perl libexpat1-dev
  libfakeroot libfile-fcntllock-perl libfontconfig1 libgcc-13-dev libgd3 libgomp1 libgprofng0 libheif-plugin-aomdec libheif-plugin-aomenc
  libheif-plugin-libde265 libheif1 libhwasan0 libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 libjs-jquery libjs-sphinxdoc libjs-underscore liblerc4 liblsan0
  libmpc3 libpython3-dev libpython3-stdlib libpython3.12-dev libpython3.12-minimal libpython3.12-stdlib libpython3.12t64 libquadmath0 libsframe1 libsharpyuv0
  libstdc++-13-dev libtiff6 libtsan2 libubsan1 libwebp7 libxpm4 linux-libc-dev linux-tools-common locales lto-disabled-list make manpages-dev python3
  python3-dev python3-minimal python3-wheel python3.12 python3.12-dev python3.12-minimal rpcsvc-proto zlib1g zlib1g-dev
Suggested packages:
  binutils-doc gprofng-gui bzip2-doc cpp-doc gcc-13-locales cpp-13-doc debsig-verify debian-keyring g++-multilib g++-13-multilib gcc-13-doc gcc-multilib
  autoconf automake libtool flex bison gdb gcc-doc gcc-13-multilib gdb-x86-64-linux-gnu apache2 | lighttpd | httpd glibc-doc libnss-nis libnss-nisplus bzr
  libgd-tools libheif-plugin-x265 libheif-plugin-ffmpegdec libheif-plugin-jpegdec libheif-plugin-jpegenc libheif-plugin-j2kdec libheif-plugin-j2kenc
  libheif-plugin-rav1e libheif-plugin-svtenc libstdc++-13-doc make-doc python3-doc python3-tk python3-venv python3.12-venv python3.12-doc binfmt-support
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential bzip2 cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu dpkg-dev fakeroot
  fontconfig-config fonts-dejavu-core fonts-dejavu-mono g++ g++-13 g++-13-x86-64-linux-gnu g++-x86-64-linux-gnu gcc gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu
  gcc-x86-64-linux-gnu javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libaom3 libasan8 libatomic1 libbinutils
  libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libde265-0 libdeflate0 libdpkg-perl libexpat1-dev libfakeroot
  libfile-fcntllock-perl libfontconfig1 libgcc-13-dev libgd3 libgomp1 libgprofng0 libheif-plugin-aomdec libheif-plugin-aomenc libheif-plugin-libde265 libheif1
  libhwasan0 libisl23 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 libjs-jquery libjs-sphinxdoc libjs-underscore liblerc4 liblsan0 libmpc3 libpython3-dev
  libpython3.12-dev libquadmath0 libsframe1 libsharpyuv0 libstdc++-13-dev libtiff6 libtsan2 libubsan1 libwebp7 libxpm4 linux-libc-dev lto-disabled-list make
  manpages-dev python3-dev python3-pip python3-wheel python3.12-dev rpcsvc-proto zlib1g-dev
```

i-01ac535cb21aade01 (demo)

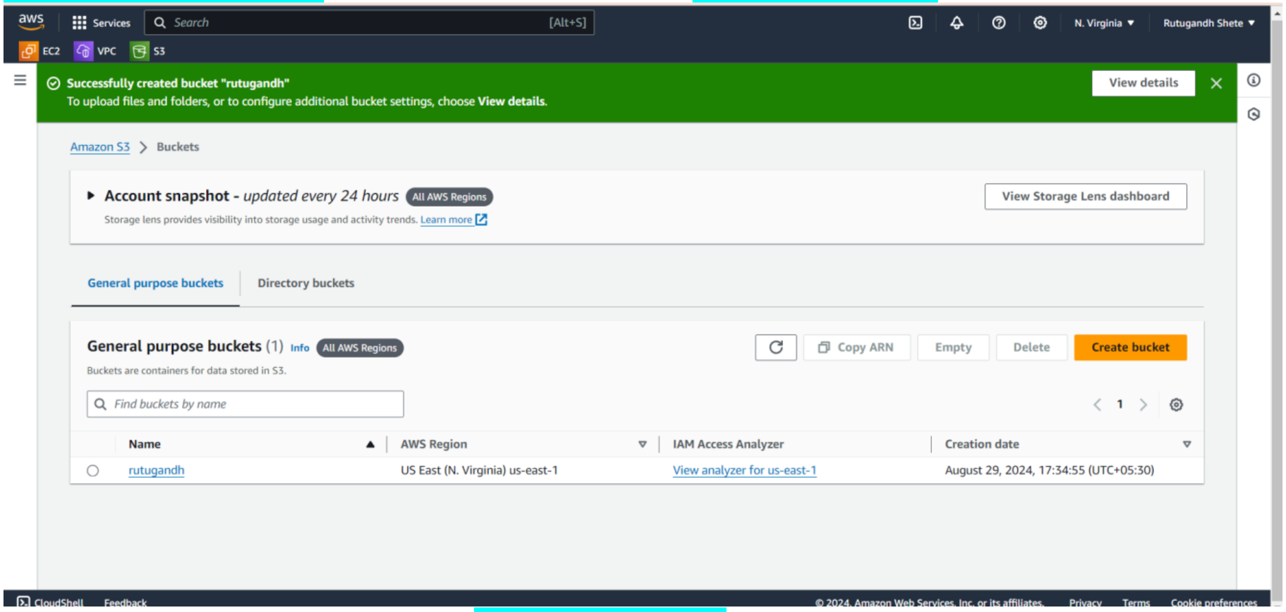
PublicIPs: 54.198.105.42 PrivateIPs: 172.31.32.94

- Then install curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip" unzip awscliv2.zip sudo ./aws/install. This CLI command for installing.
- After the installation is done then use command "ls", there will be configure "aws" file which can lead to privacy threat so we can delete that file. Use "rm -rf aws".



```
root@ip-172-31-32-94:~# ls
aws   awscliv2.zip   snap
```

- Create one s3 bucket for example purpose in us-east-1 region



Successfully created bucket "rutugandh"
To upload files and folders, or to configure additional bucket settings, choose **View details**.

Amazon S3 > Buckets

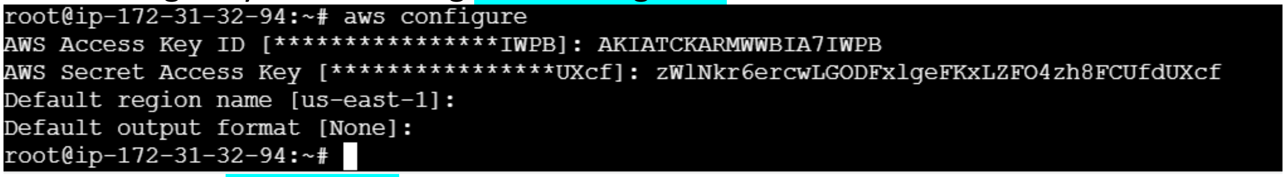
Account snapshot - *updated every 24 hours* All AWS Regions
Storage lens provides visibility into storage usage and activity trends. Learn more

General purpose buckets | Directory buckets

General purpose buckets (1) Info All AWS Regions
Buckets are containers for data stored in S3.

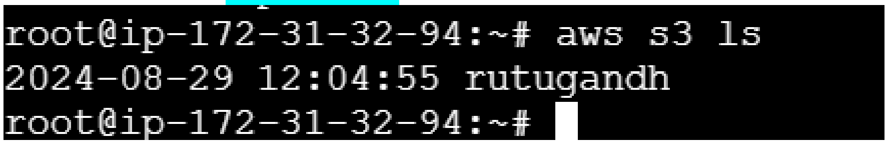
Name	AWS Region	IAM Access Analyzer	Creation date
rutugandh	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 29, 2024, 17:34:55 (UTC+05:30)

- Then configure your aws using "aws configure".



```
root@ip-172-31-32-94:~# aws configure
AWS Access Key ID [****************IWPB]: AKIATCKARMWWBIA7IWPB
AWS Secret Access Key [****************UXcf]: zWlNkr6ercwLGODFxlgeFKxLZFO4zh8FCUfdUXcf
Default region name [us-east-1]:
Default output format [None]:
root@ip-172-31-32-94:~#
```

- use command "aws s3 ls".



```
root@ip-172-31-32-94:~# aws s3 ls
2024-08-29 12:04:55 rutugandh
root@ip-172-31-32-94:~#
```