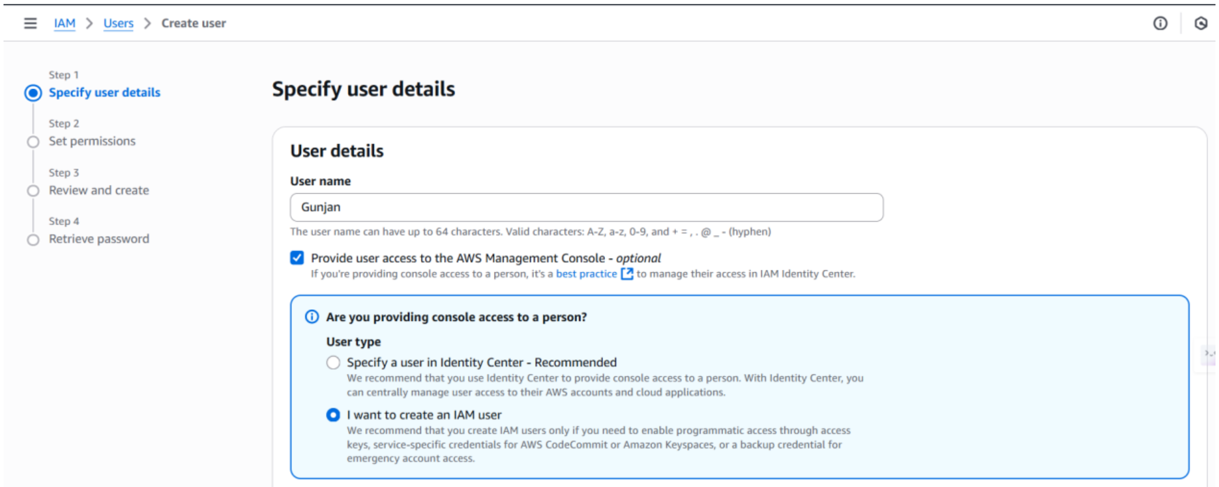
IAM Resources Task

IAM (Identity and Access Management) is a service in AWS that helps you decide who can use your AWS account and what they can do.

1. **User:** Individual identities

   User Creation steps-



Specify user details

User details

User name

Gunjan

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?

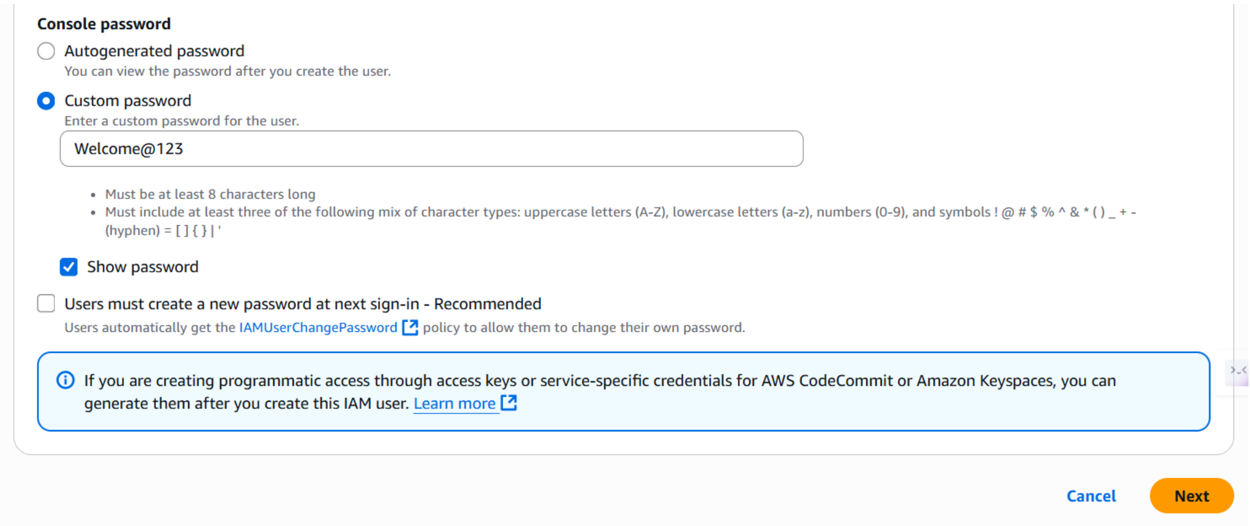
User type

○ 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.

Welcome@123

- 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) = [ ] { } | '

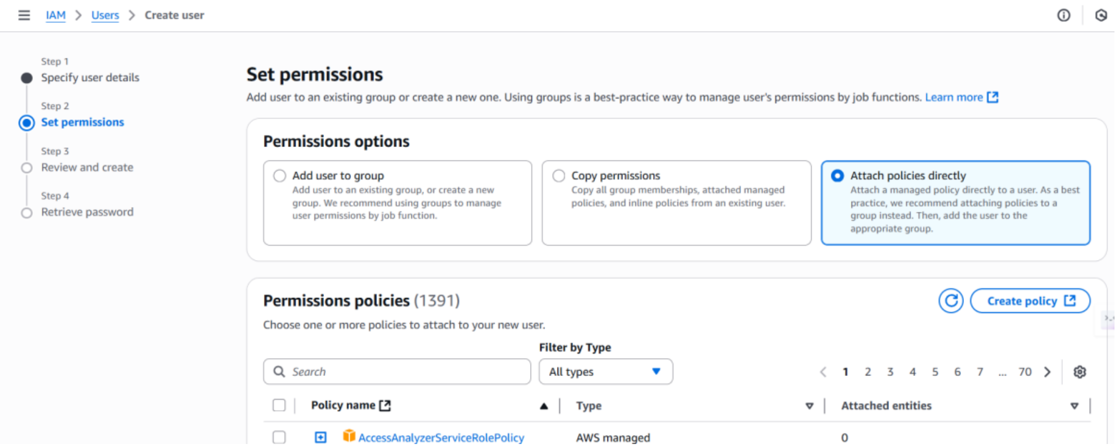
☑ Show password

□ 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

Attach Permission to the user



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.

Permissions policies (1391)

Create policy

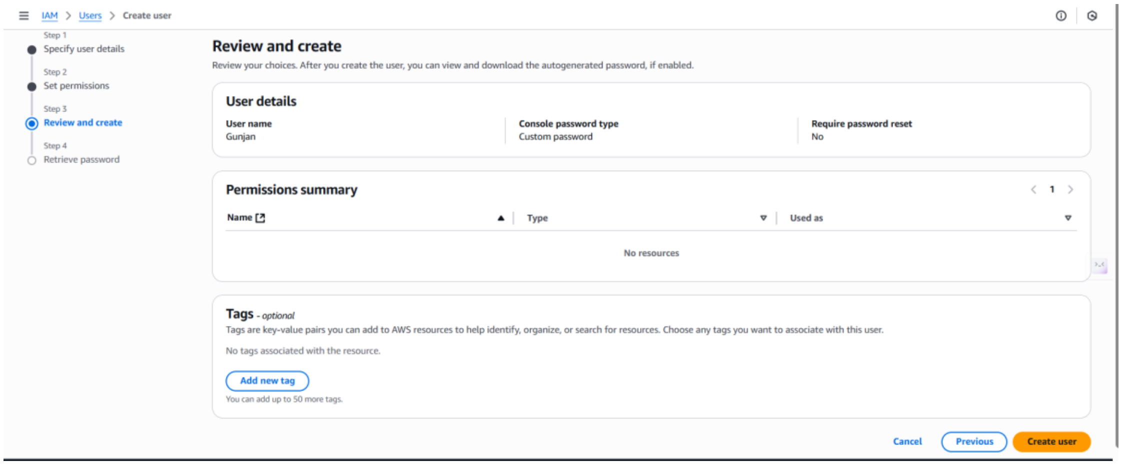
Choose one or more policies to attach to your new user.

Filter by Type

Search | All types

< 1 2 3 4 5 6 7 ... 70 >

□	Policy name	Type	Attached entities
□	AccessAnalyzerServiceRolePolicy	AWS managed	0



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	Console password type	Require password reset
Gunjan	Custom password	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.

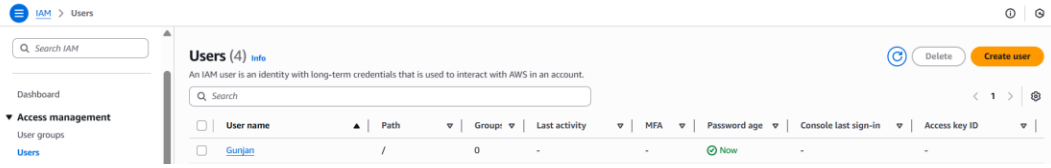
No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel | Previous | Create user

You can set MFA to user(optional)



IAM > Users

Search IAM

Dashboard

▼ Access management

User groups

Users

Users (4) Info

Delete | Create user

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Search

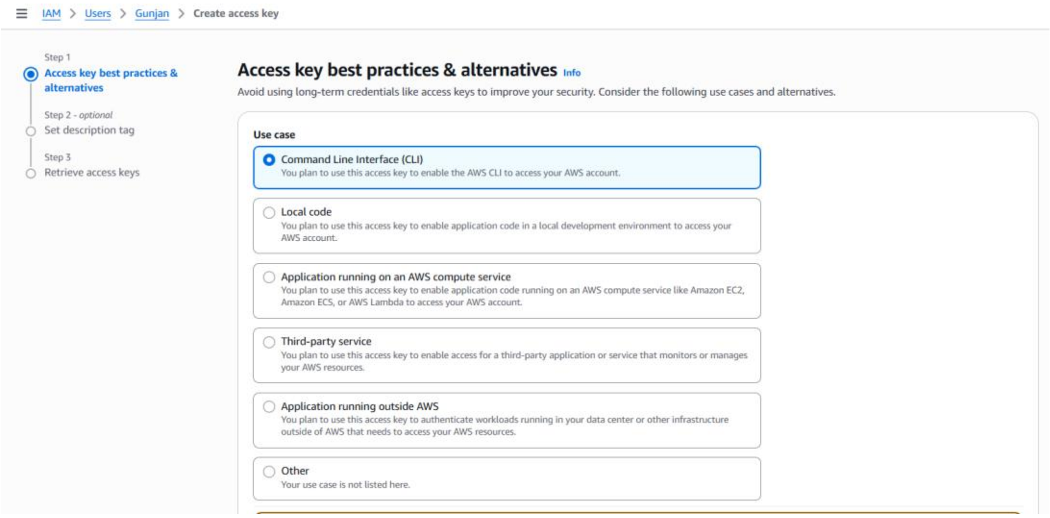
< 1 >

□	User name	Path	Group:	Last activity	MFA	Password age	Console last sign-in	Access key ID
□	Gunjan	/	0	-	-	✓ Now	-	-

Why do we need Access Key for a user?

- Because **applications, scripts, or CLI (Command Line Interface)** cannot type passwords like humans.
- Access keys act like a **username + password pair** for software/programs.

Create Access key for user. You can create only 2 keys of single user



Step 1
◉ Access key best practices & alternatives

Step 2 - *optional*
○ Set description tag

Step 3
○ Retrieve access keys

Access key best practices & alternatives Info

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

Use case

● Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.

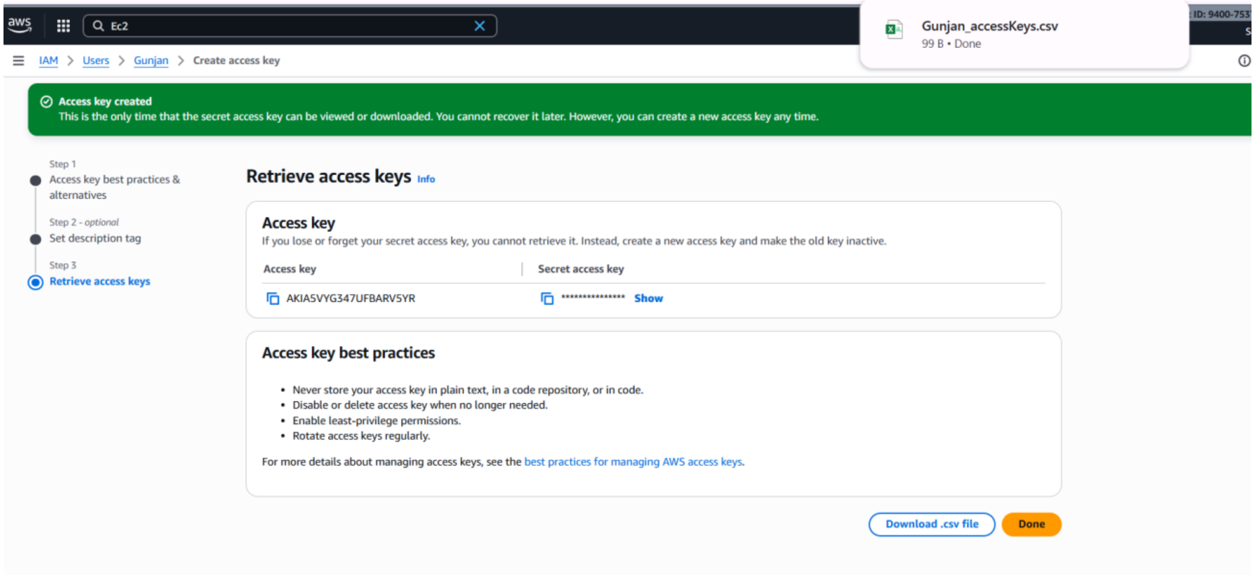
○ Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.

○ 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.



✓ **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.

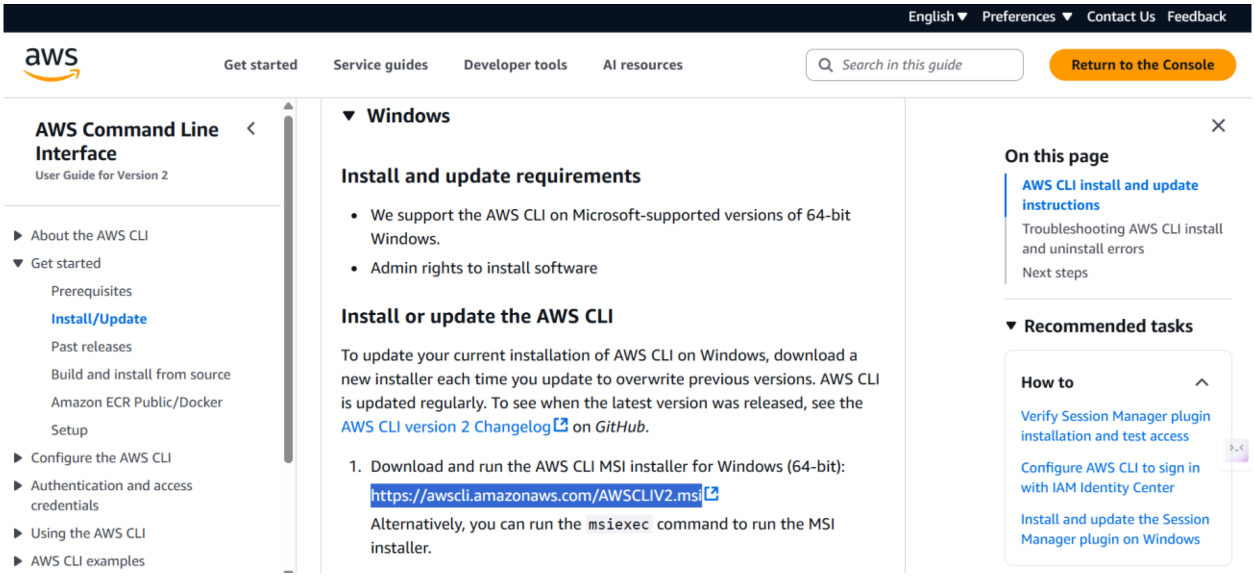
Access key	Secret access key
AKIA5VYG347UFBARV5YR	*************** Show

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.

Install Aws CLI on windows OS



AWS Command Line Interface
User Guide for Version 2

▼ Windows

Install and update requirements

- We support the AWS CLI on Microsoft-supported versions of 64-bit Windows.
- Admin rights to install software

Install or update the AWS CLI

To update your current installation of AWS CLI on Windows, download a new installer each time you update to overwrite previous versions. AWS CLI is updated regularly. To see when the latest version was released, see the AWS CLI version 2 Changelog on *GitHub*.

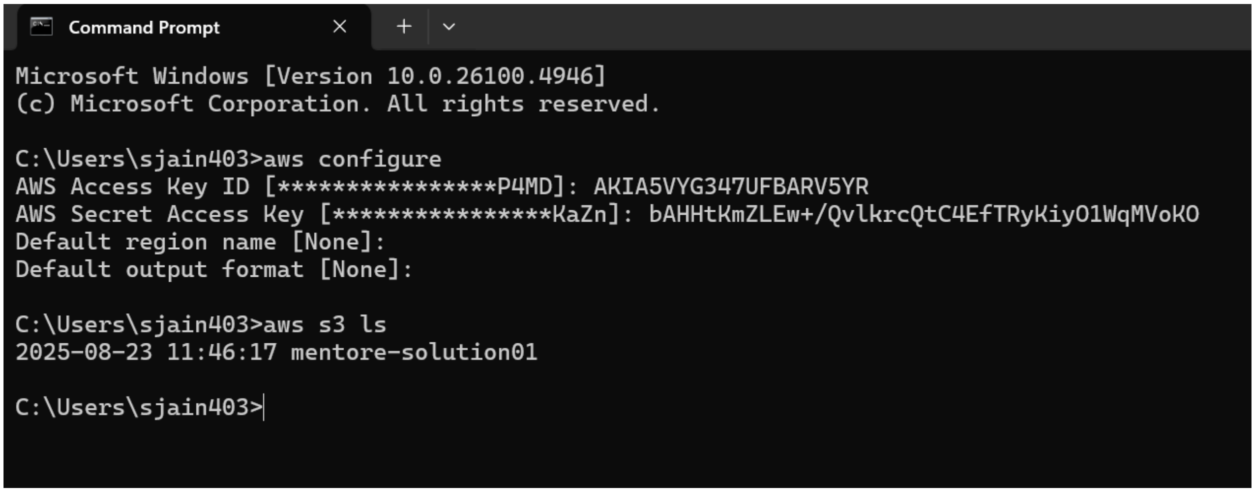
1. Download and run the AWS CLI MSI installer for Windows (64-bit):

   https://awscli.amazonaws.com/AWSCLIV2.msi

   Alternatively, you can run the `msiexec` command to run the MSI installer.

On this page
AWS CLI install and update instructions
Troubleshooting AWS CLI install and uninstall errors
Next steps
▼ Recommended tasks
How to
Verify Session Manager plugin installation and test access
Configure AWS CLI to sign in with IAM Identity Center
Install and update the Session Manager plugin on Windows

You can access user through CLI



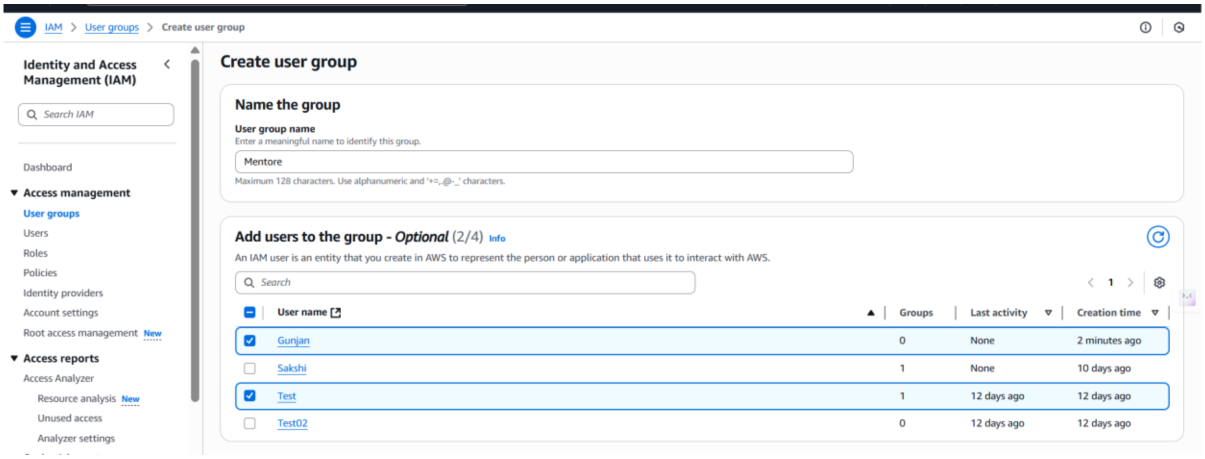
```
Microsoft Windows [Version 10.0.26100.4946]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sjain403>aws configure
AWS Access Key ID [****************P4MD]: AKIA5VYG347UFBARV5YR
AWS Secret Access Key [****************KaZn]: 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
Default region name [None]:
Default output format [None]:

C:\Users\sjain403>aws s3 ls
2025-08-23 11:46:17 mentore-solution01

C:\Users\sjain403>
```

2. **User Groups:** A collection of users.



Create user group

Name the group

User group name

Enter a meaningful name to identify this group.

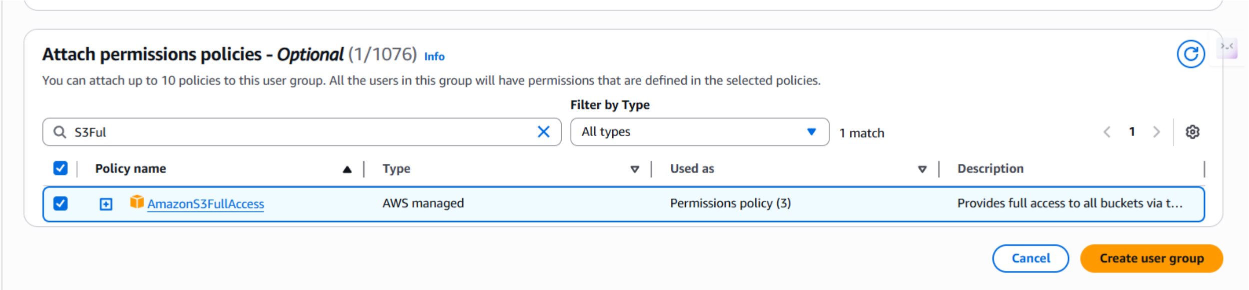
Mentore

Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

Add users to the group - *Optional* (2/4) 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
☑	Gunjan	0	None	2 minutes ago
□	Sakshi	1	None	10 days ago
☑	Test	1	12 days ago	12 days ago
□	Test02	0	12 days ago	12 days ago



Attach permissions policies - *Optional* (1/1076) 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.

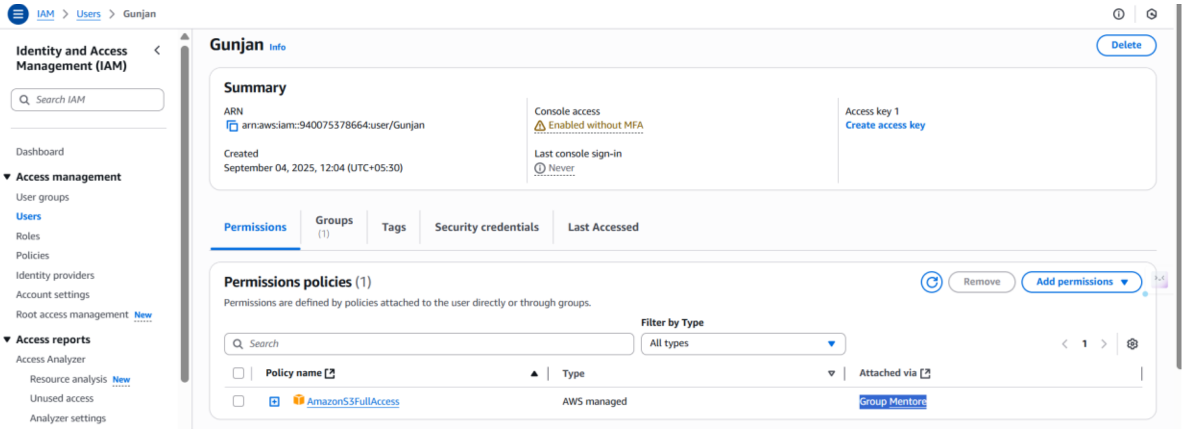
Search: S3Ful

Filter by Type: All types

1 match

☑	Policy name	Type	Used as	Description
☑	AmazonS3FullAccess	AWS managed	Permissions policy (3)	Provides full access to all buckets via t…

You can see user is added in group and having group permissions



IAM > Users > Gunjan

Identity and Access Management (IAM)

Search IAM

Dashboard

▼ Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

Root access management New

▼ Access reports

Access Analyzer

Resource analysis New

Unused access

Analyzer settings

Gunjan Info

Delete

Summary

ARN: arn:aws:iam::940075378664:user/Gunjan

Console access: ⚠ Enabled without MFA

Access key 1: Create access key

Created: September 04, 2025, 12:04 (UTC+05:30)

Last console sign-in: ⓘ Never

Permissions | Groups (1) | Tags | Security credentials | Last Accessed

Permissions policies (1)

Remove | Add permissions ▼

Permissions are defined by policies attached to the user directly or through groups.

Filter by Type: All types

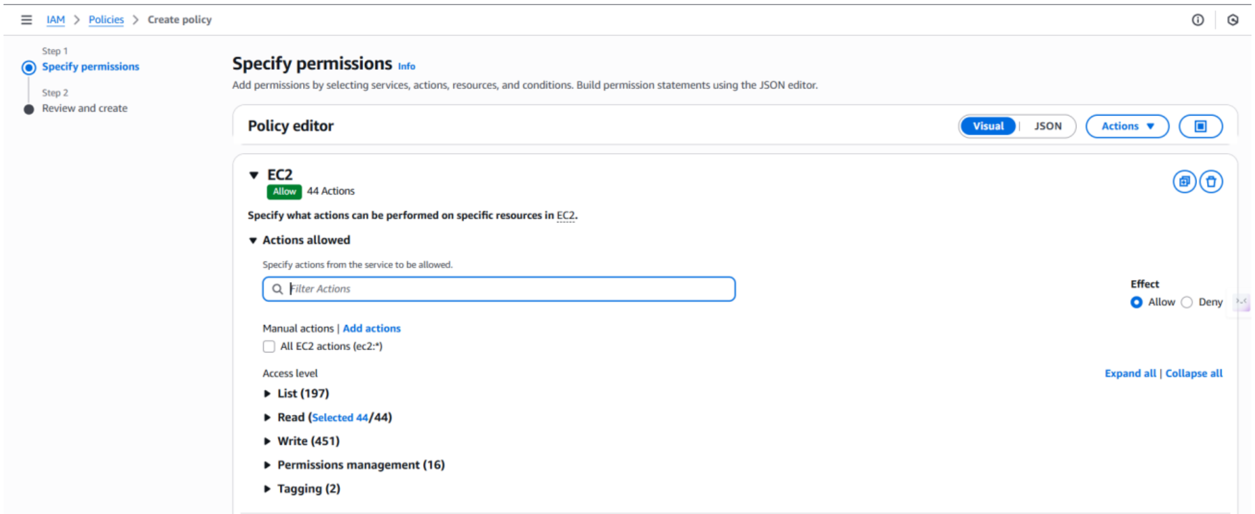
Search

□	Policy name	Type	Attached via
□	AmazonS3FullAccess	AWS managed	Group Mentore

3. **Policies:** Rules that define what a user, group, or role can do.

You can create policy for any service

Here is 2 ways to create policy -1. Visual 2. JSON



IAM > Policies > Create policy

Step 1
Specify permissions

Step 2
Review and create

Specify permissions Info

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

Policy editor

Visual | JSON | Actions ▼

▼ EC2

Allow 44 Actions

Specify what actions can be performed on specific resources in EC2.

▼ Actions allowed

Specify actions from the service to be allowed.

Filter Actions

Effect: ● Allow ○ Deny

Manual actions | Add actions

□ All EC2 actions (ec2:*)

Access level

Expand all | Collapse all

▶ List (197)

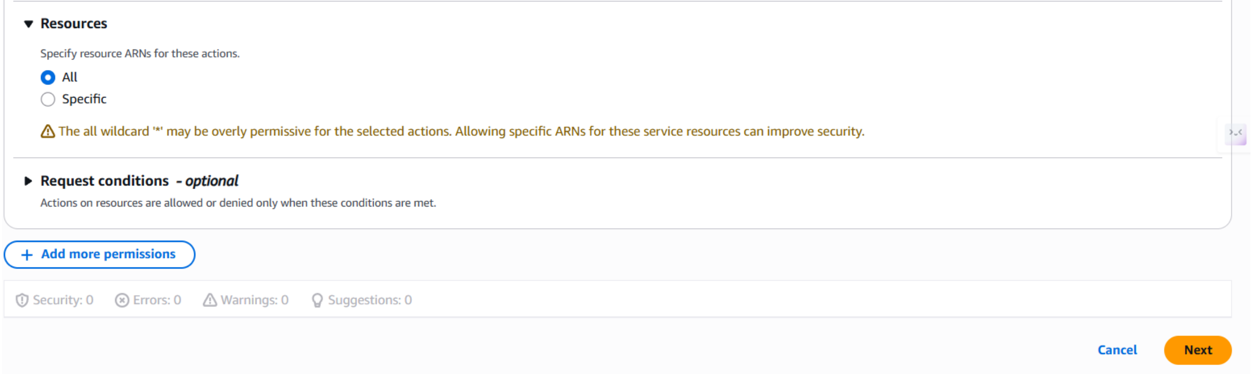
▶ Read (Selected 44/44)

▶ Write (451)

▶ Permissions management (16)

▶ Tagging (2)

You can create policy for all resource or specific resource.



▼ Resources

Specify resource ARNs for these actions.

● All

○ Specific

⚠ The all wildcard '*' may be overly permissive for the selected actions. Allowing specific ARNs for these service resources can improve security.

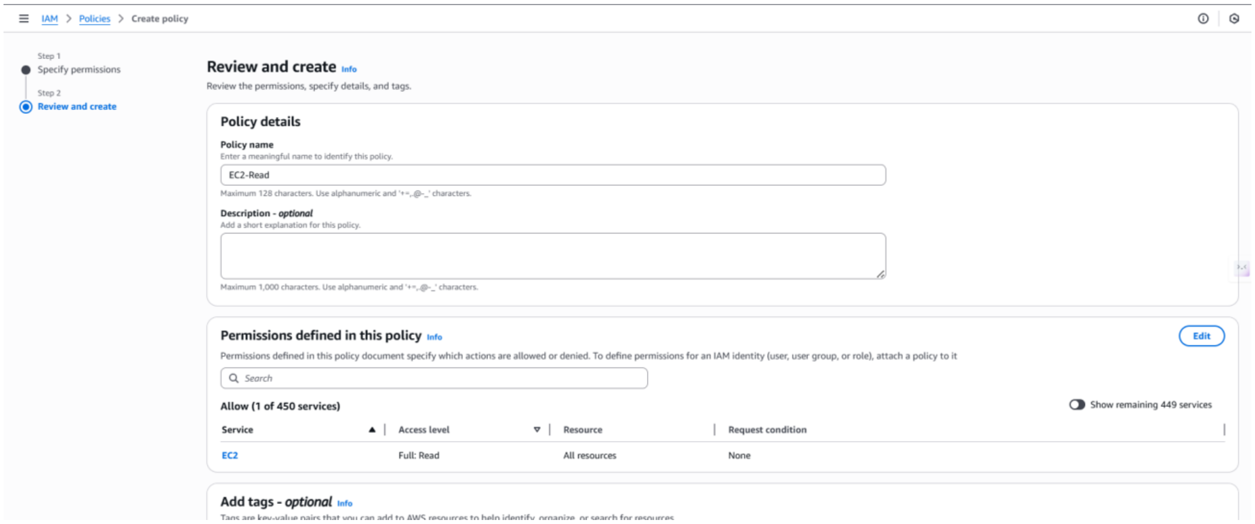
▶ Request conditions - *optional*

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

+ Add more permissions

Security: 0 | Errors: 0 | Warnings: 0 | Suggestions: 0

Cancel | Next



IAM > Policies > Create policy

Step 1
Specify permissions

Step 2
Review and create

Review and create Info

Review the permissions, specify details, and tags.

Policy details

Policy name
Enter a meaningful name to identify this policy.

EC2-Read

Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

Description - *optional*
Add a short explanation for this policy.

Maximum 1,000 characters. Use alphanumeric and '+=,.@-_' characters.

Permissions defined in this policy Info

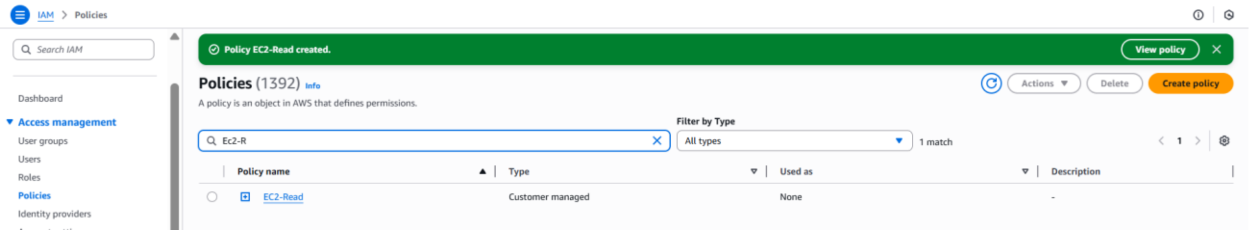
Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it

Allow (1 of 450 services)

Service	Access level	Resource	Request condition
EC2	Full: Read	All resources	None

Add tags - *optional* Info

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



IAM > Policies

Policy EC2-Read created.

Policies (1392) Info

A policy is an object in AWS that defines permissions.

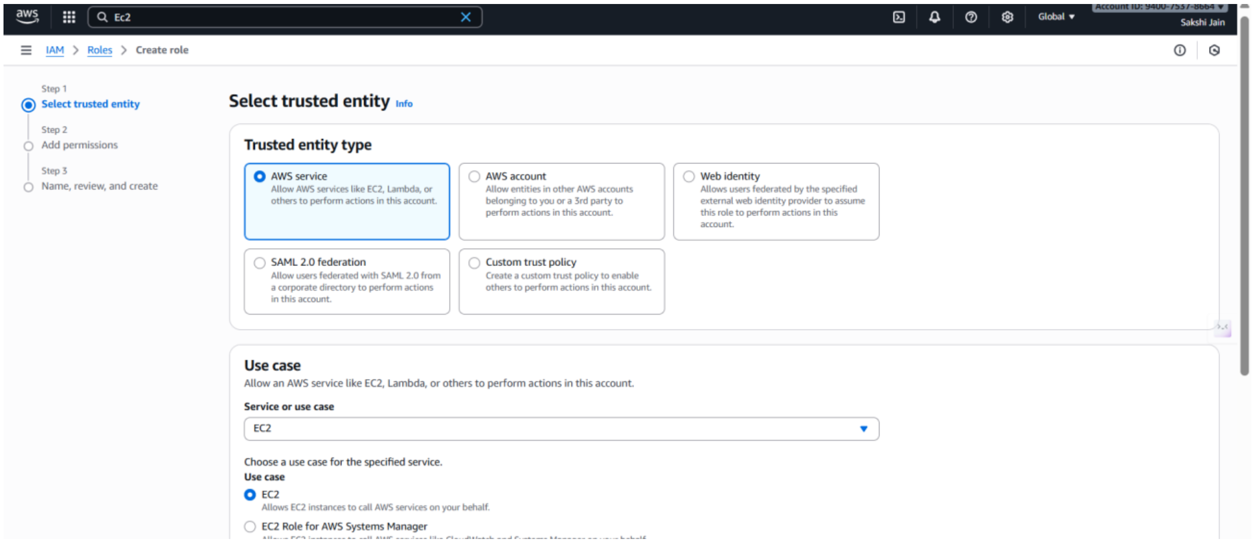
Ec2-R

Filter by Type
All types

1 match

	Policy name	Type	Used as	Description
○	EC2-Read	Customer managed	None	-

4. **Roles:** Temporary identities with permissions, usually for applications or services.



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

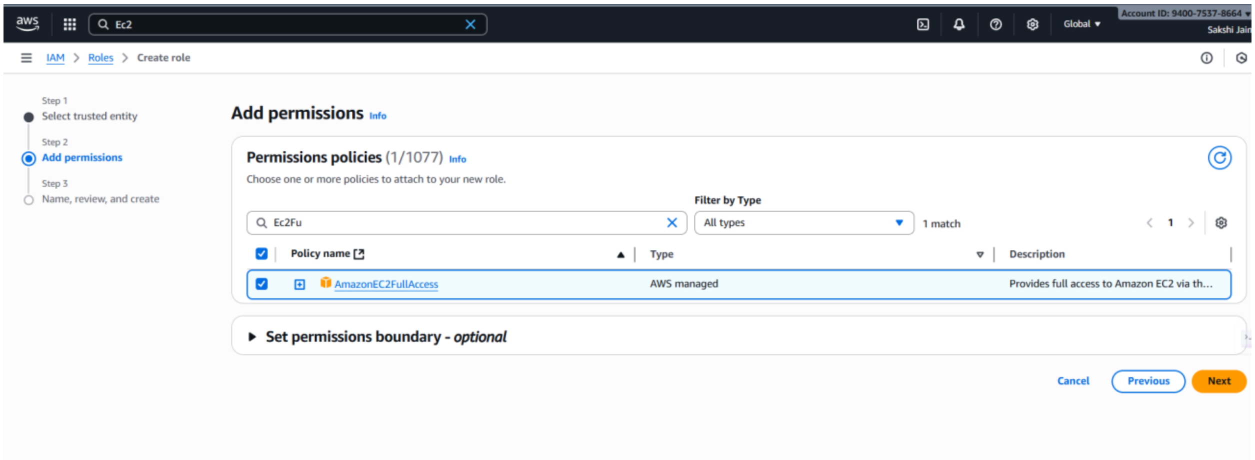
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

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



Add permissions

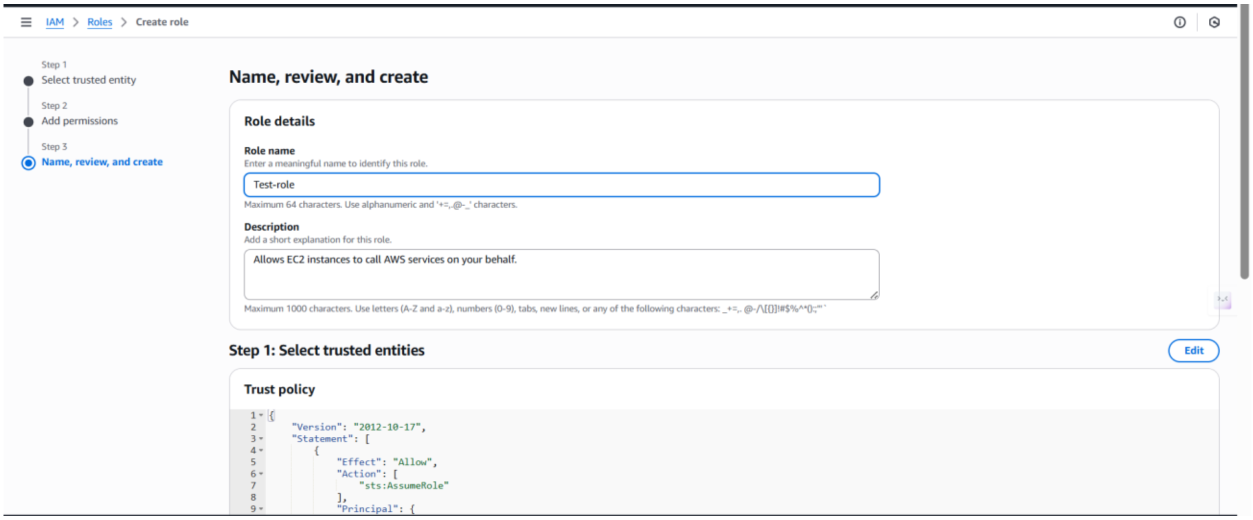
Permissions policies (1/1077)

Choose one or more policies to attach to your new role.

Search: Ec2Fu — Filter by Type: All types — 1 match

☑	Policy name	Type	Description
☑	AmazonEC2FullAccess	AWS managed	Provides full access to Amazon EC2 via th...

▶ Set permissions boundary - *optional*



Name, review, and create

Role details

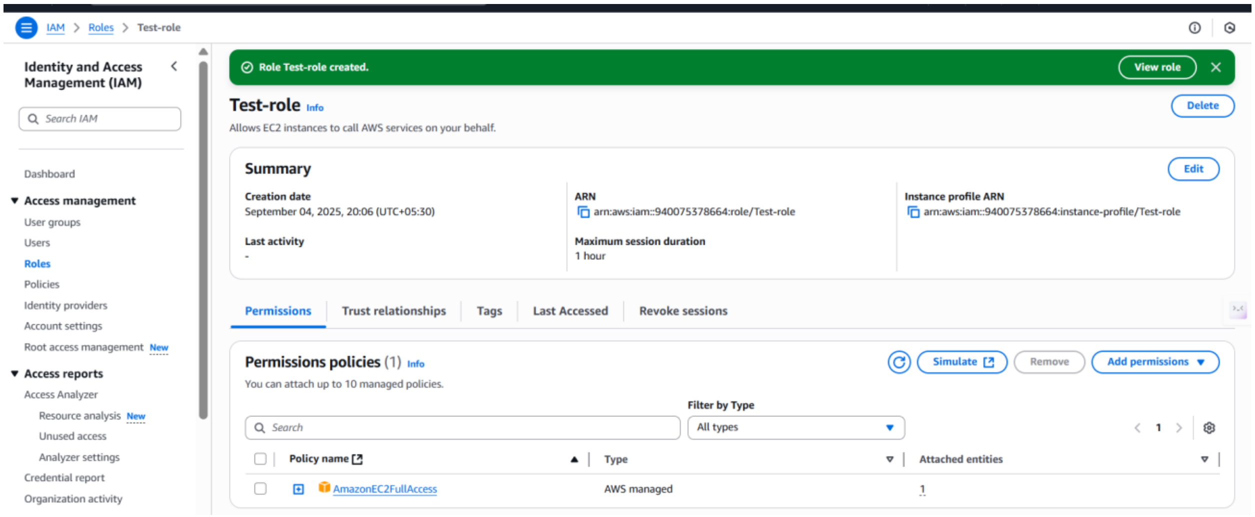
Role name
Enter a meaningful name to identify this role.
Test-role
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description
Add a short explanation for this role.
Allows EC2 instances to call AWS services on your behalf.
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/\[{}]!#$%^*();:"'`

Step 1: Select trusted entities

Trust policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Principal": {
```



Role Test-role created.

Test-role

Allows EC2 instances to call AWS services on your behalf.

Summary

Creation date
September 04, 2025, 20:06 (UTC+05:30)

Last activity
-

ARN
arn:aws:iam::940075378664:role/Test-role

Maximum session duration
1 hour

Instance profile ARN
arn:aws:iam::940075378664:instance-profile/Test-role

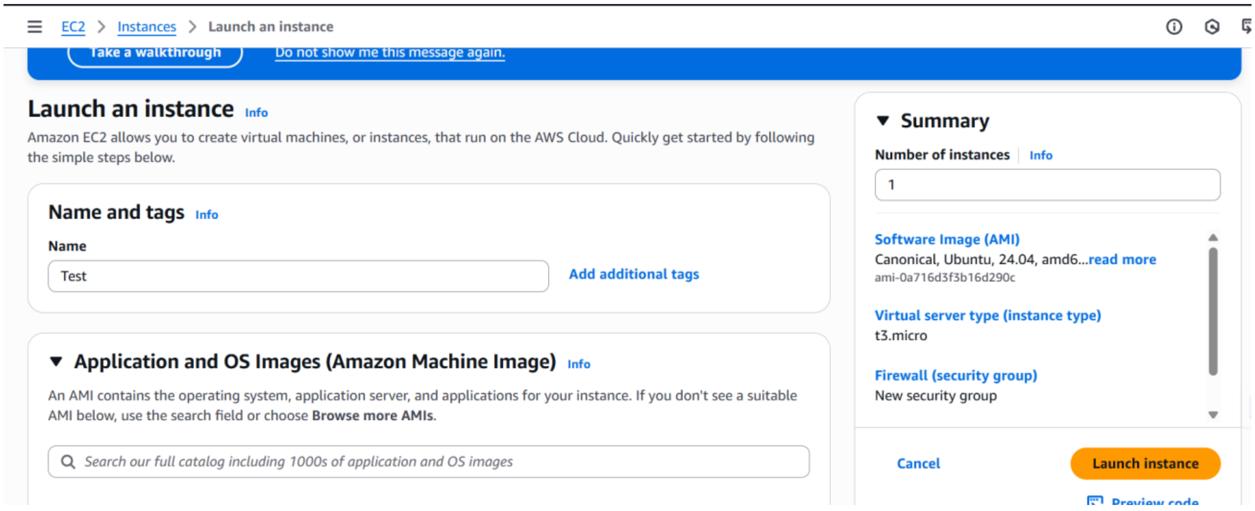
Permissions policies (1)

You can attach up to 10 managed policies.

Search — Filter by Type: All types

□	Policy name	Type	Attached entities
□	AmazonEC2FullAccess	AWS managed	1

Create EC2 Instance


EC2 > Instances > Launch an instance
Take a walkthrough
Do not show me this message again.

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: Test

▼ Application and OS Images (Amazon Machine Image) Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose **Browse more AMIs**.

Search our full catalog including 1000s of application and OS images

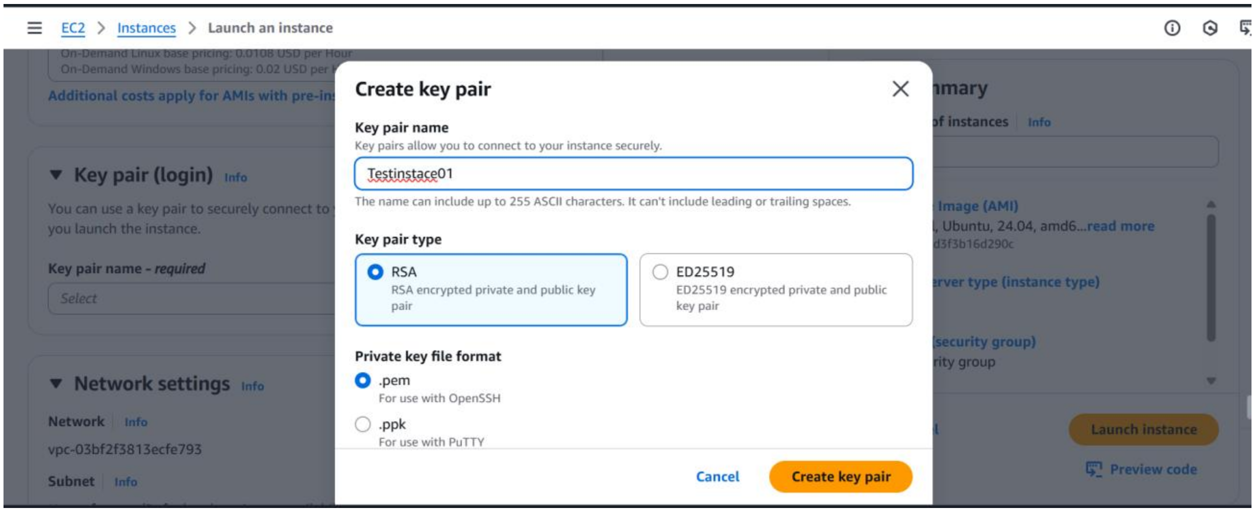
▼ Summary

Number of instances Info: 1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-0a716d3f3b16d290c

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group



Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.

Testinstace01

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

● RSA
RSA encrypted private and public key pair

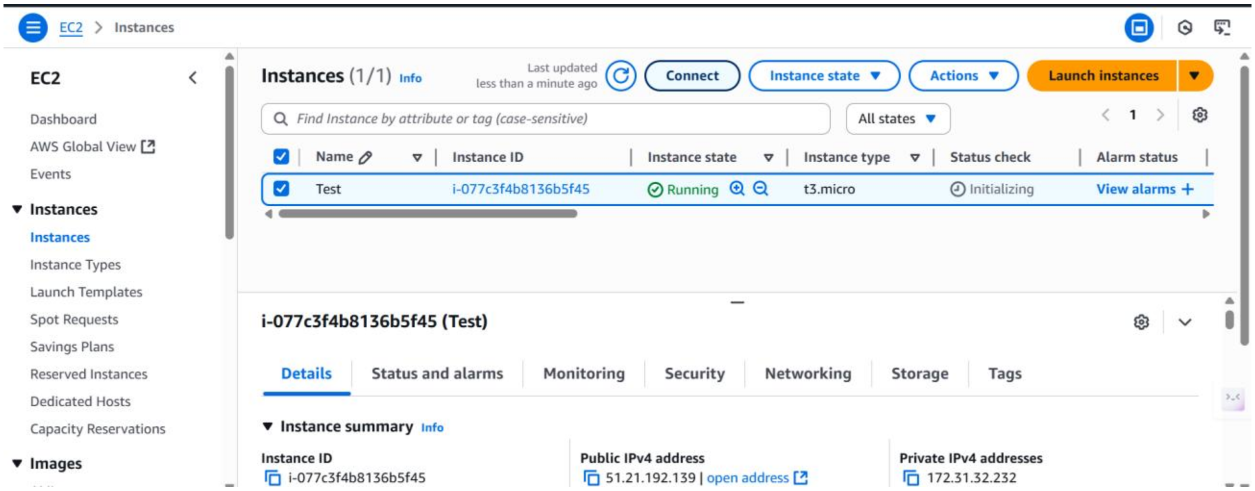
○ ED25519
ED25519 encrypted private and public key pair

Private key file format

● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY

Click on connect



Instances (1/1) Info — Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive)

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
☑	Test	i-077c3f4b8136b5f45	Running	t3.micro	Initializing	View alarms +

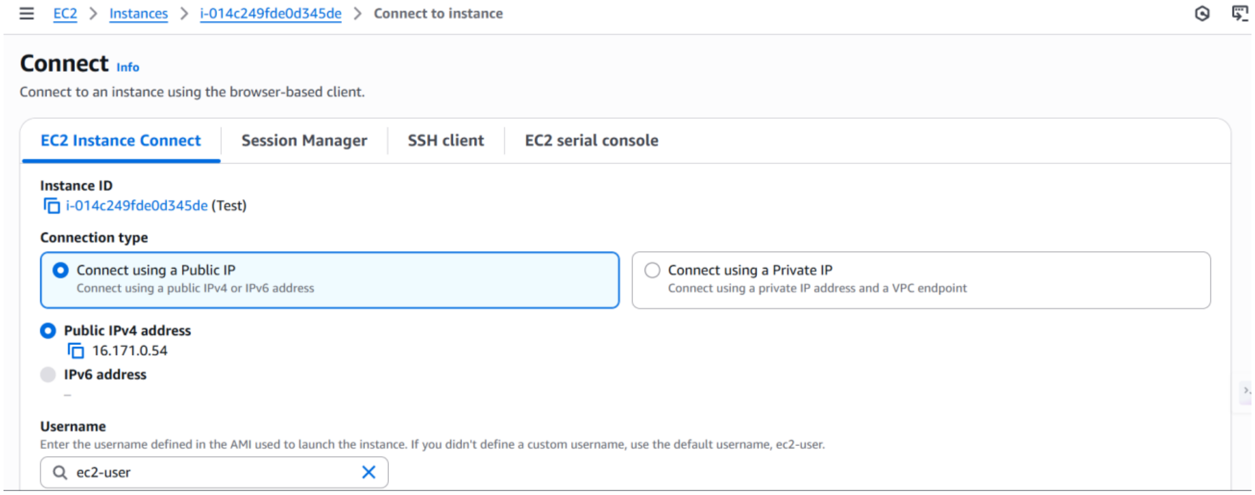
i-077c3f4b8136b5f45 (Test)

▼ Instance summary Info

Instance ID: i-077c3f4b8136b5f45

Public IPv4 address: 51.21.192.139 | open address

Private IPv4 addresses: 172.31.32.232



EC2 > Instances > i-014c249fde0d345de > Connect to instance

Connect Info

Connect to an instance using the browser-based client.

Instance ID

i-014c249fde0d345de (Test)

Connection type

● Connect using a Public IP
Connect using a public IPv4 or IPv6 address

○ Connect using a Private IP
Connect using a private IP address and a VPC endpoint

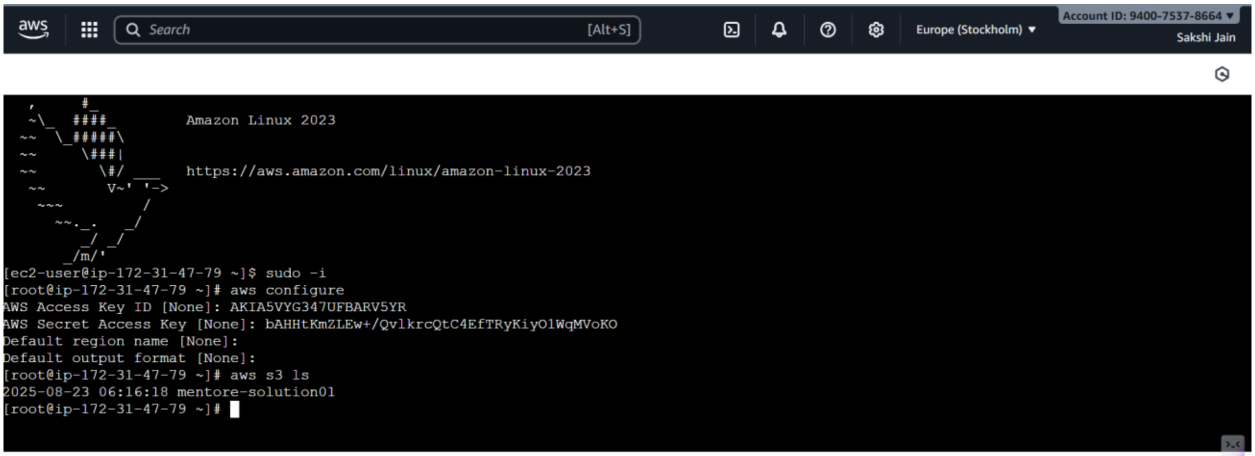
● Public IPv4 address
16.171.0.54

○ IPv6 address
–

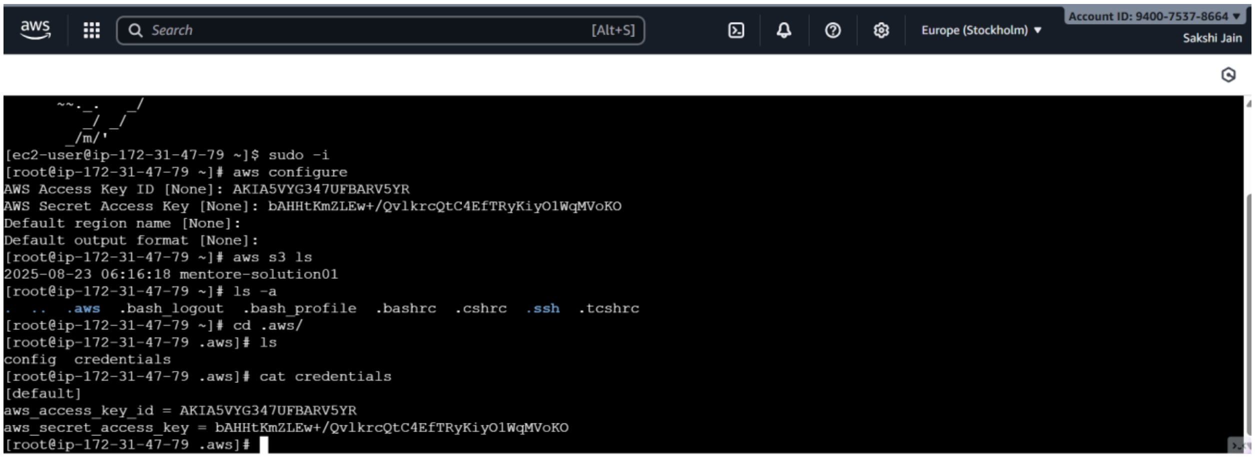
Username

Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.

ec2-user



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-47-79 ~]$ sudo -i
[root@ip-172-31-47-79 ~]# aws configure
AWS Access Key ID [None]: AKIA5VYG347UFBARV5YR
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
[root@ip-172-31-47-79 ~]# aws s3 ls
2025-08-23 06:16:18 mentore-solution01
[root@ip-172-31-47-79 ~]#
```



```
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-47-79 ~]$ sudo -i
[root@ip-172-31-47-79 ~]# aws configure
AWS Access Key ID [None]: AKIA5VYG347UFBARV5YR
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
[root@ip-172-31-47-79 ~]# aws s3 ls
2025-08-23 06:16:18 mentore-solution01
[root@ip-172-31-47-79 ~]# ls -a
.  ..  .aws  .bash_logout  .bash_profile  .bashrc  .cshrc  .ssh  .tcshrc
[root@ip-172-31-47-79 ~]# cd .aws/
[root@ip-172-31-47-79 .aws]# ls
config  credentials
[root@ip-172-31-47-79 .aws]# cat credentials
[default]
aws_access_key_id = AKIA5VYG347UFBARV5YR
aws_secret_access_key = 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
[root@ip-172-31-47-79 .aws]#
```