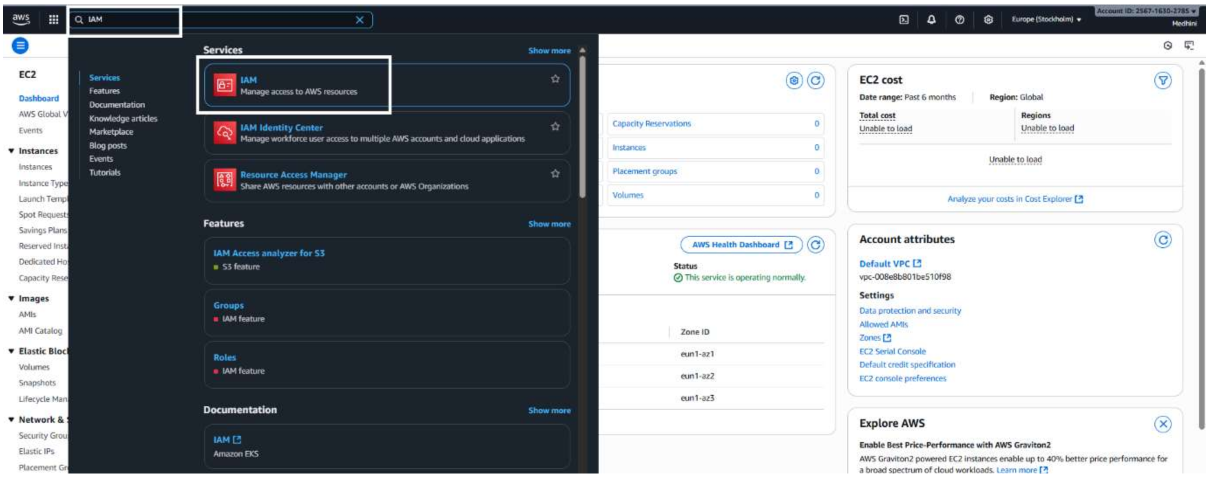
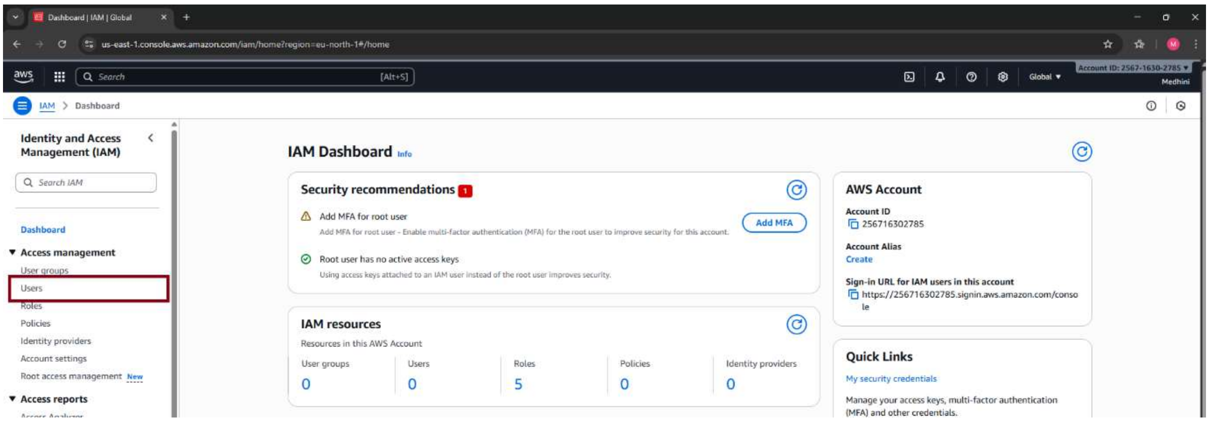
Create 4 IAM users named "Dev1", "Dev2", "Test1" and "Test2"

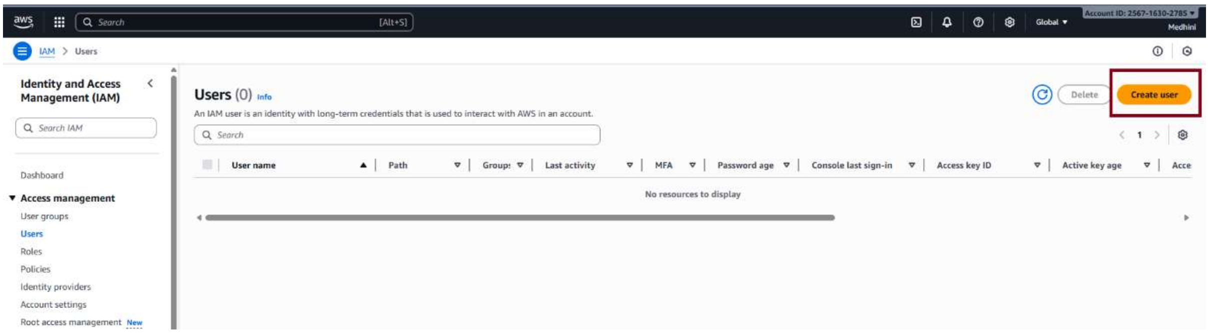
Step 1: Login to your account and search for **IAM** and click on **IAM**



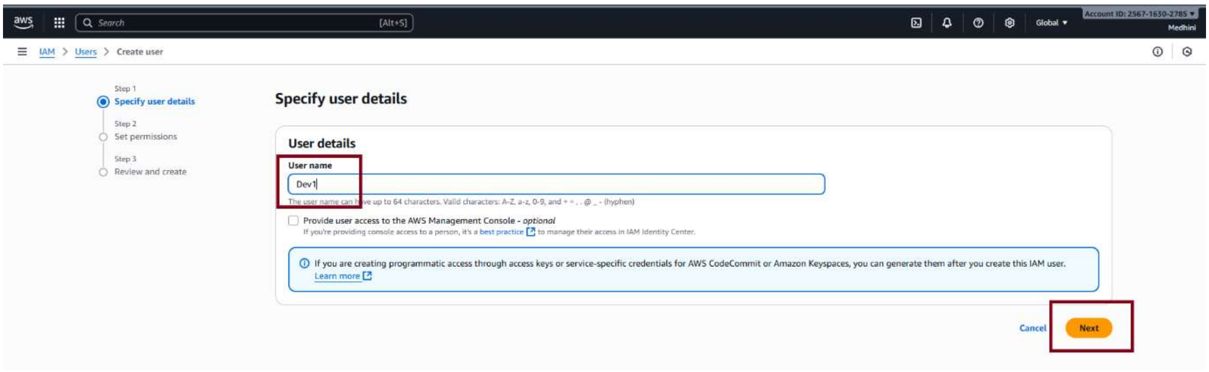
Step 2: You will be redirected to **IAM Dashboard** and go to "**Users**" and click on it


Dashboard | IAM | Global
us-east-1.console.aws.amazon.com/iam/home?region=eu-north-1#/home
Search
[Alt+S]
Global
Account ID: 2567-1630-2785
Medhini
IAM > Dashboard
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
IAM Dashboard Info
Security recommendations 1
Add MFA for root user
Add MFA for root user - Enable multi-factor authentication (MFA) for the root user to improve security for this account.
Add MFA
Root user has no active access keys
Using access keys attached to an IAM user instead of the root user improves security.
IAM resources
Resources in this AWS Account
User groups 0
Users 0
Roles 5
Policies 0
Identity providers 0
AWS Account
Account ID
256716302785
Account Alias
Create
Sign-in URL for IAM users in this account
https://256716302785.signin.aws.amazon.com/console
Quick Links
My security credentials
Manage your access keys, multi-factor authentication (MFA) and other credentials.

Step 3: Click on "**Create User**" as shown in below picture



Step 4: Give a user name as mentioned **Dev1** and click on "**Next**"



Specify user details

User details

User name

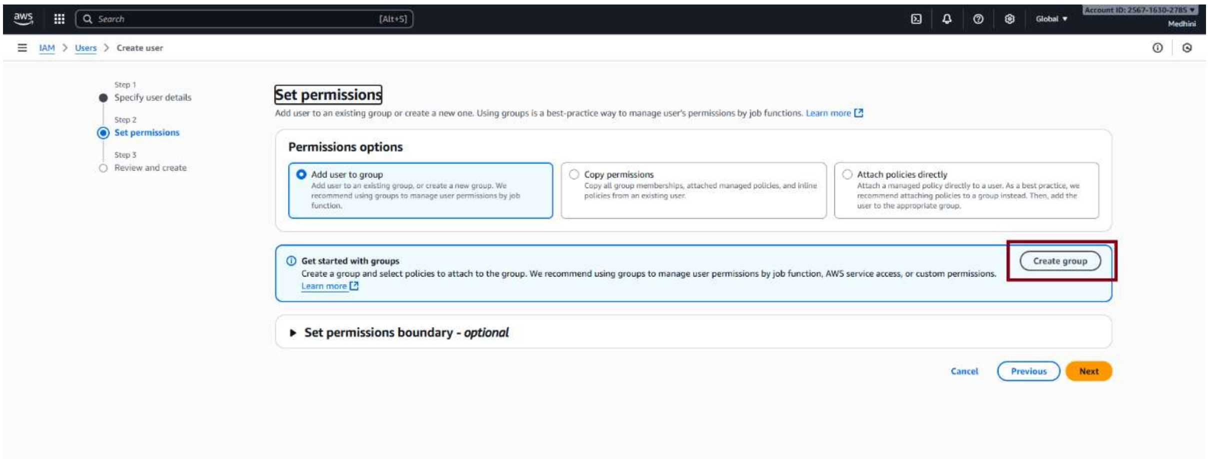
Dev1

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.

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

Step 5: Click on "**Create group**" since there are no groups yet.



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.

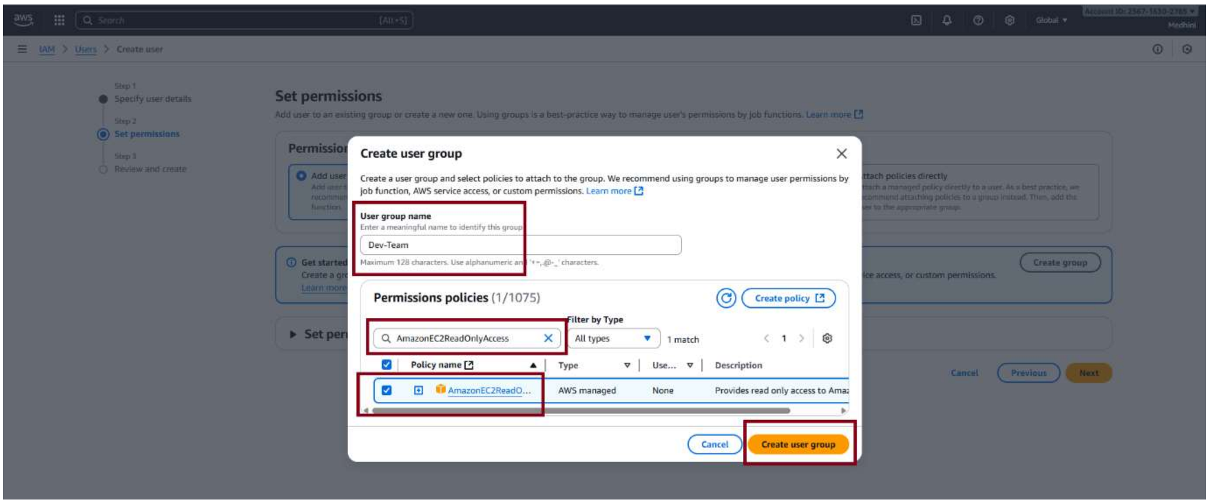
○ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

○ 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

▸ Set permissions boundary - *optional*

Step 6: Give group name as "**Dev Team**" and select policy and click on "**Create Group**" select policy according to requirement



Create user group

Create a user 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

User group name
Enter a meaningful name to identify this group

Dev-Team

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

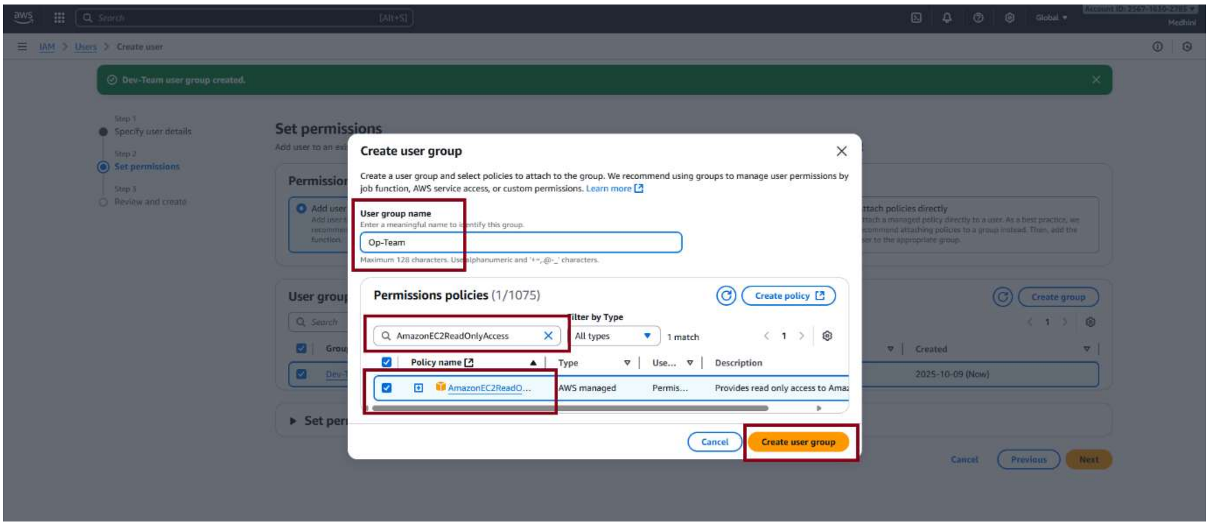
Permissions policies (1/1075)

AmazonEC2ReadOnlyAccess

All types

☑	Policy name	Type	Use...	Description
☑	AmazonEC2ReadO...	AWS managed	None	Provides read only access to Amaz

Step 7: Create one more group as "**Op-Team**" as shown in below picture



Dev-Team user group created.

Create user group

Create a user 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

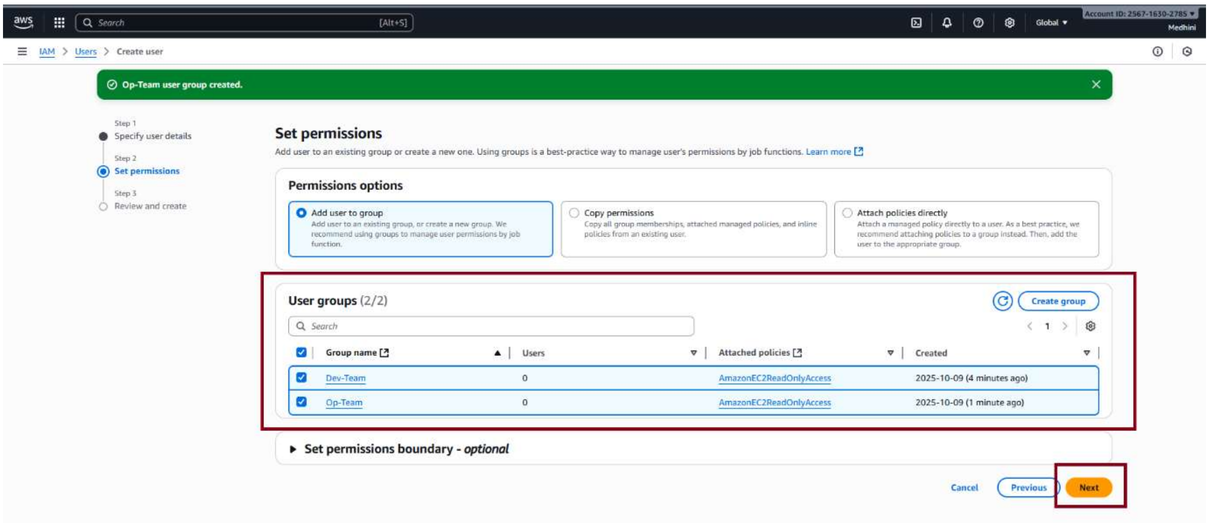
User group name
Enter a meaningful name to identify this group.
Op-Team
Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

Permissions policies (1/1075)

Filter by Type: AmazonEC2ReadOnlyAccess | All types | 1 match

	Policy name	Type	Use...	Description
☑	AmazonEC2ReadO...	AWS managed	Permis...	Provides read only access to Amaz

Step 8: Now "**Dev1**" user is added for both "**Dev Team**" and "**Op Team**" and then click on "**Next**"



Op-Team user group created.

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.

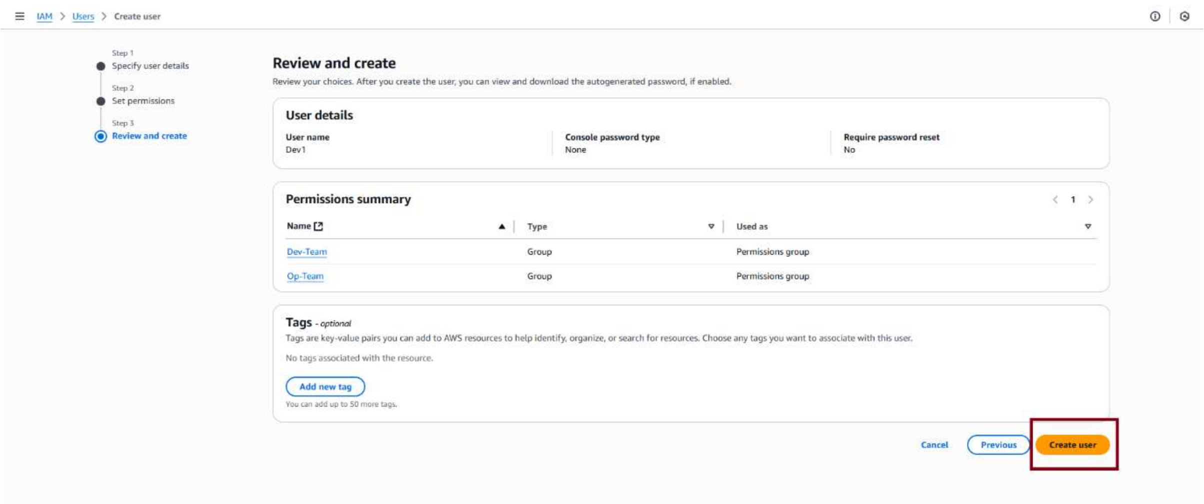
○ Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

User groups (2/2)

☑	Group name	Users	Attached policies	Created
☑	Dev-Team	0	AmazonEC2ReadOnlyAccess	2025-10-09 (4 minutes ago)
☑	Op-Team	0	AmazonEC2ReadOnlyAccess	2025-10-09 (1 minute ago)

▸ Set permissions boundary - *optional*

Step 9: Review it and click on "**Create User**"



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
Dev1	None	No

Permissions summary

Name	Type	Used as
Dev-Team	Group	Permissions group
Op-Team	Group	Permissions group

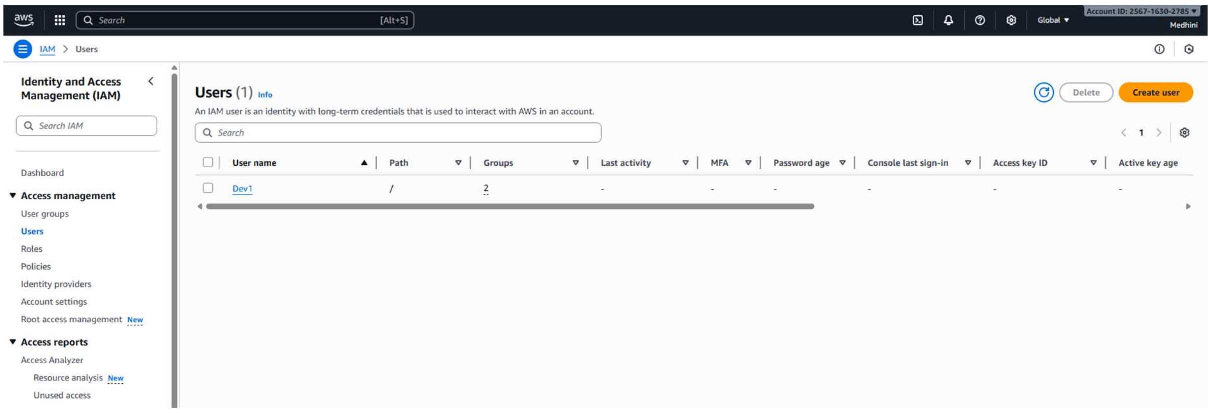
Tags - *optional*

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.

You can add up to 50 more tags.

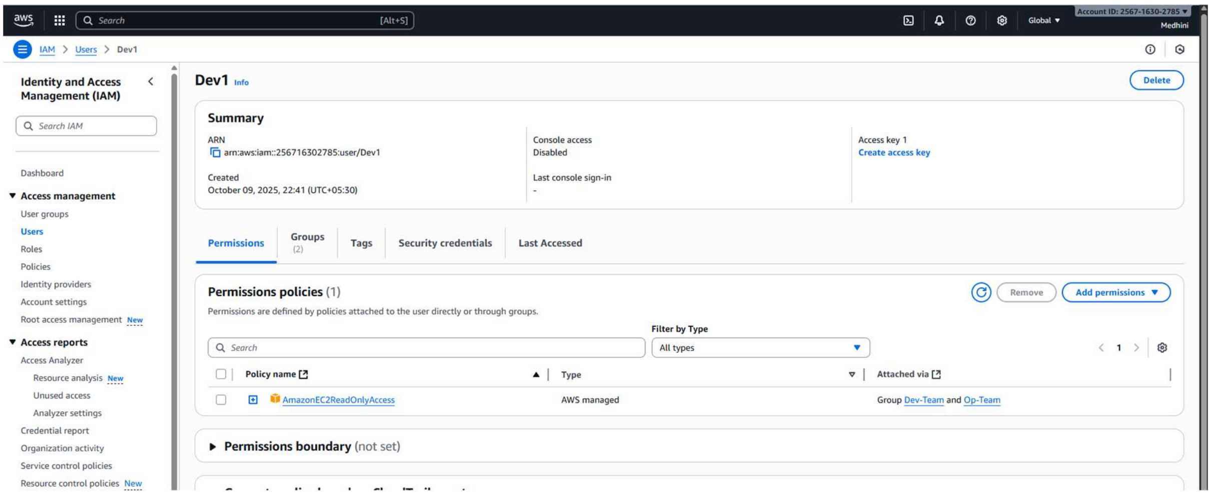
Step 10: "**Dev1**" is added to both the Groups "**Dev Team**" and "**Op Team**" as shown in below figure



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	Access key ID	Active key age
Dev1	/	2	-	-	-	-	-	-



Dev1 Info

Summary

ARN: arn:aws:iam::256716302785:user/Dev1

Console access: Disabled

Access key 1: Create access key

Created: October 09, 2025, 22:41 (UTC+05:30)

Last console sign-in: -

Permissions policies (1)

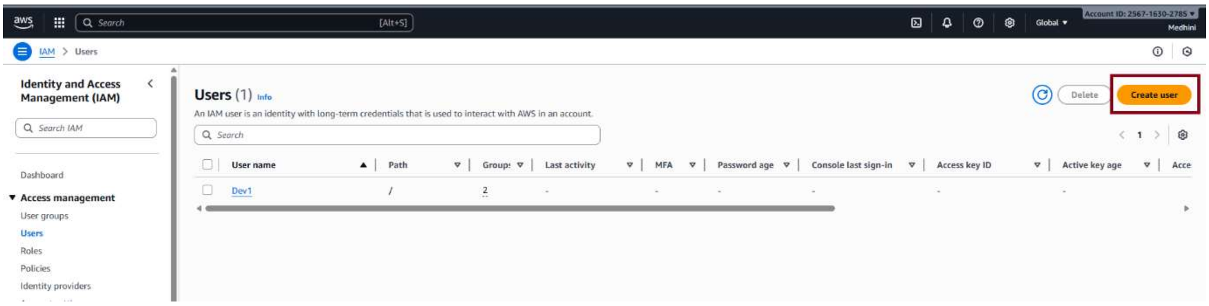
Permissions are defined by policies attached to the user directly or through groups.

Filter by Type: All types

Policy name	Type	Attached via
AmazonEC2ReadOnlyAccess	AWS managed	Group Dev-Team and Op-Team

Create Dev2 and Add to "Dev Team" Group

Step 1: Go to user and click on "**Create users**"

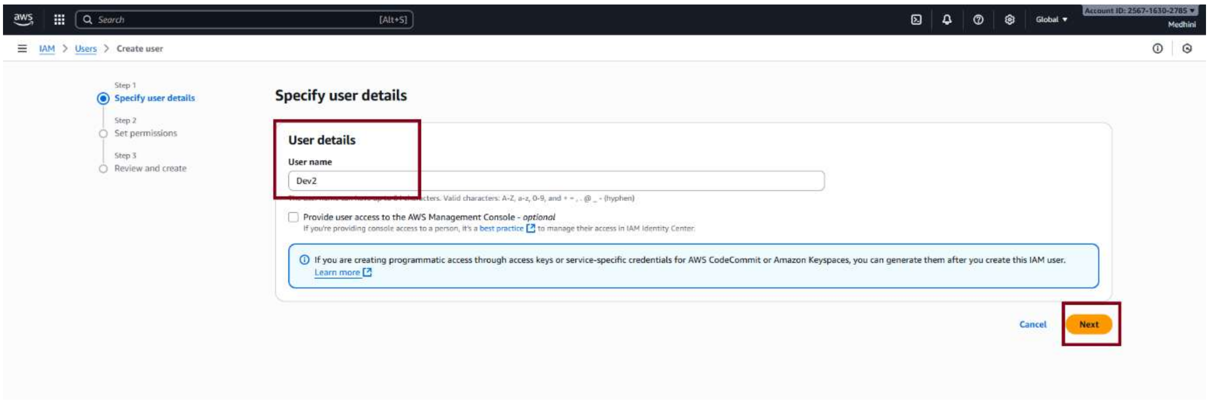


Users (1) Info

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	Access key ID	Active key age	Acce
Dev1	/	2	-	-	-	-	-	-	

Step 2: Give user name as "**Dev2**" and click on "**Next**" as shown in below picture



Step 1
● Specify user details
Step 2
○ Set permissions
Step 3
○ Review and create

Specify user details

User details

User name: Dev2

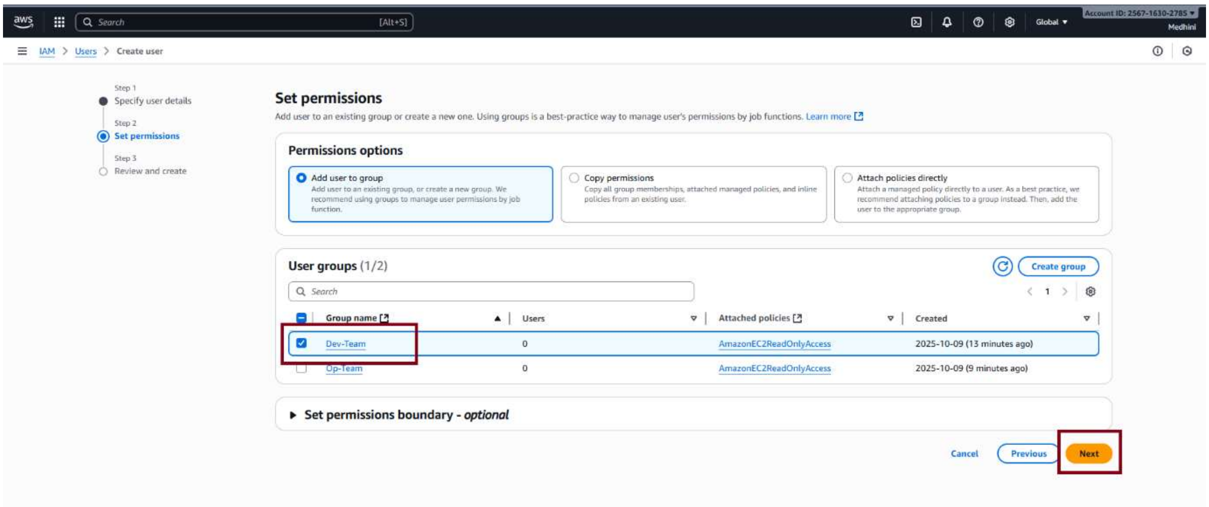
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☐ 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.

ⓘ 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

Cancel | Next

Step 3: Select “**Dev-Team**” group and click on “**Next**”



Step 1
● Specify user details
Step 2
● Set permissions
Step 3
○ Review and create

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.

User groups (1/2)

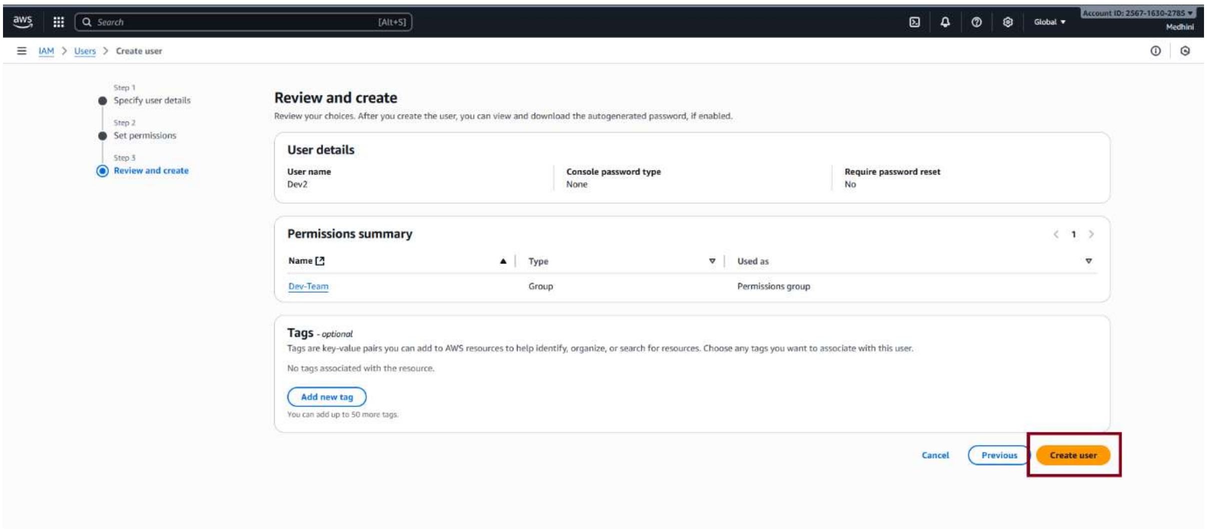
Create group

	Group name	Users	Attached policies	Created
☑	Dev-Team	0	AmazonEC2ReadOnlyAccess	2025-10-09 (13 minutes ago)
☐	Op-Team	0	AmazonEC2ReadOnlyAccess	2025-10-09 (9 minutes ago)

▸ **Set permissions boundary** - *optional*

Cancel | Previous | Next

Step 4: Click on “**Create user**”



Step 1
● Specify user details
Step 2
● Set permissions
Step 3
● Review and create

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
Dev2	None	No

Permissions summary

Name	Type	Used as
Dev-Team	Group	Permissions group

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.

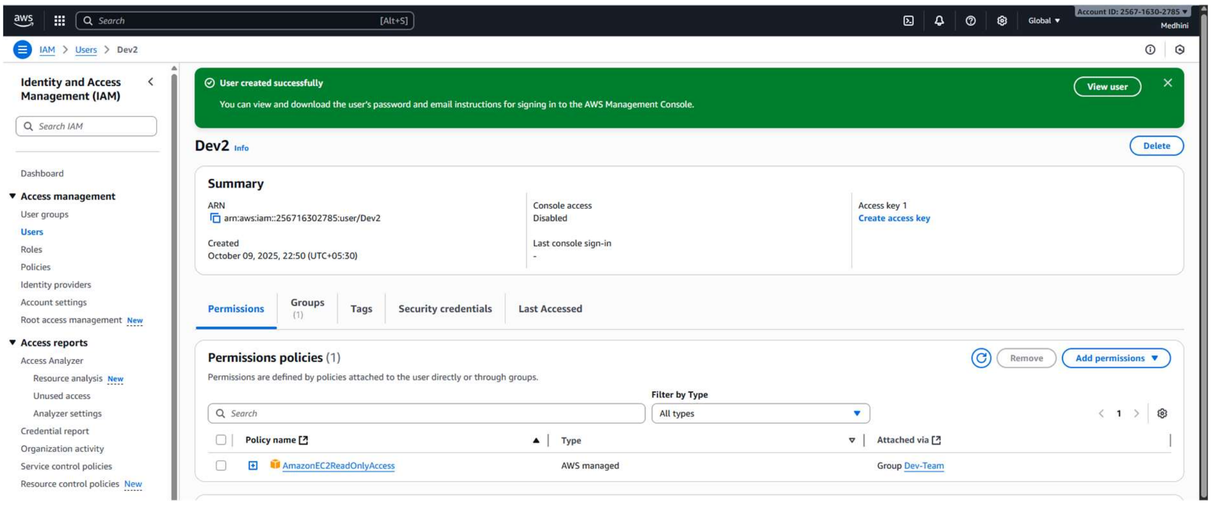
No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel | Previous | Create user

Step 5: **Dev2** is added to "**Dev-Team**" as shown in below picture



User created successfully

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

Dev2 Info

Summary

ARN arn:aws:iam::256716302785:user/Dev2	Console access Disabled	Access key 1 Create access key
Created October 09, 2025, 22:50 (UTC+05:30)	Last console sign-in -	

Permissions policies (1)

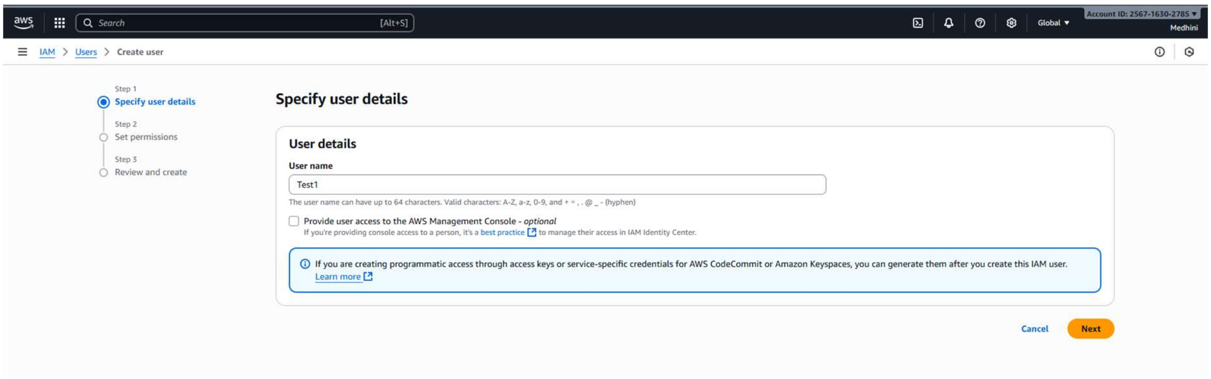
Permissions are defined by policies attached to the user directly or through groups.

Filter by Type: All types

Policy name	Type	Attached via
AmazonEC2ReadOnlyAccess	AWS managed	Group Dev-Team

Create Test 1 and Test 2 user

Step 1: Follow above steps initially and give a name and click "**Next**"



Specify user details

User details

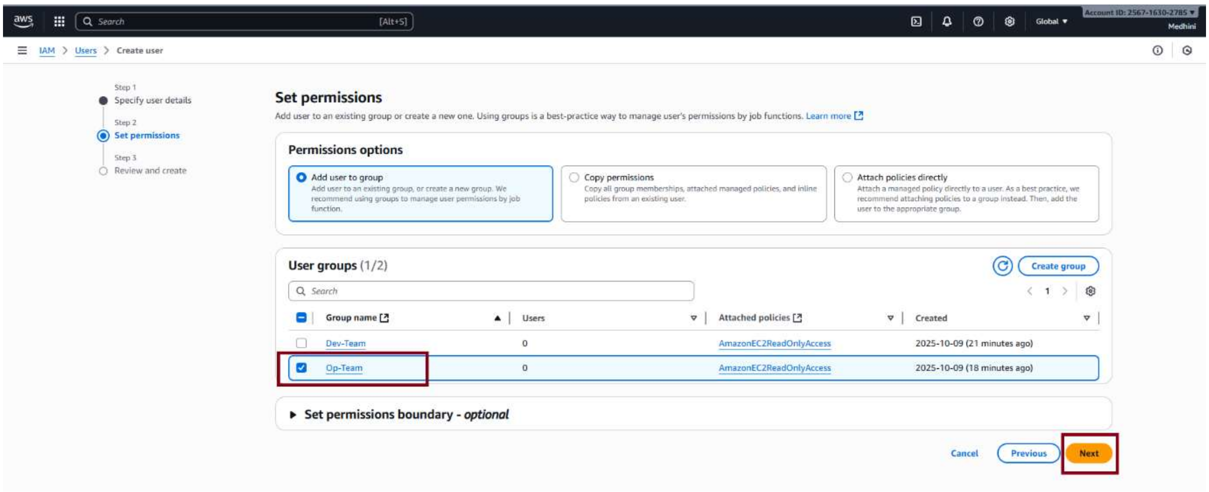
User name: Test1

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.

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

Step 2: Select "**Op-Team**" and click on "**Next**"



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.

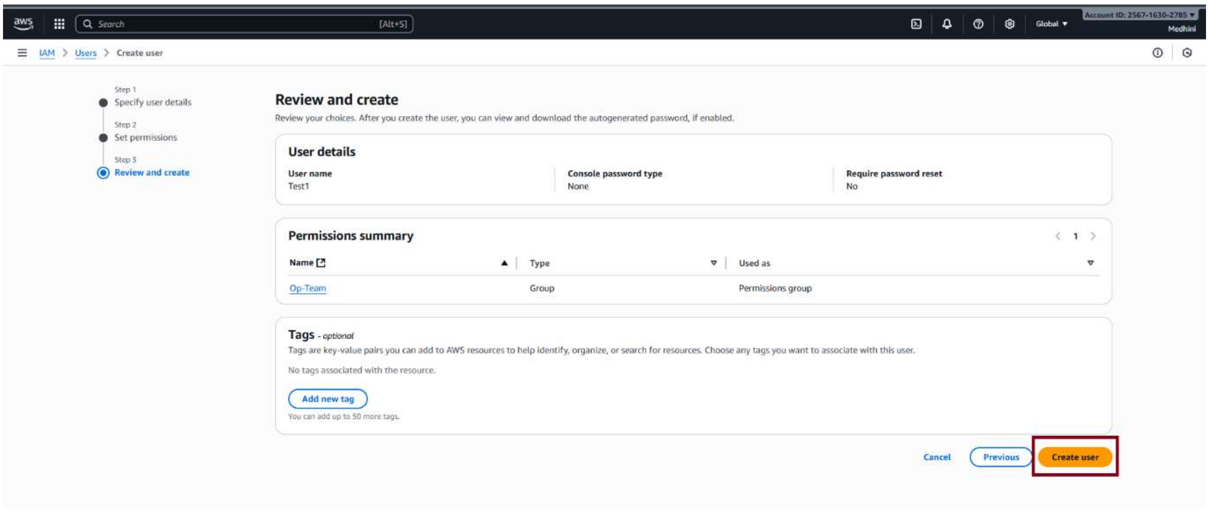
○ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

○ 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.

User groups (1/2)

	Group name	Users	Attached policies	Created
□	Dev-Team	0	AmazonEC2ReadOnlyAccess	2025-10-09 (21 minutes ago)
☑	Op-Team	0	AmazonEC2ReadOnlyAccess	2025-10-09 (18 minutes ago)

Step 3: Review and click on "**Next**" as shown in below picture



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
Test1	None	No

Permissions summary

Name	Type	Used as
Op-Team	Group	Permissions group

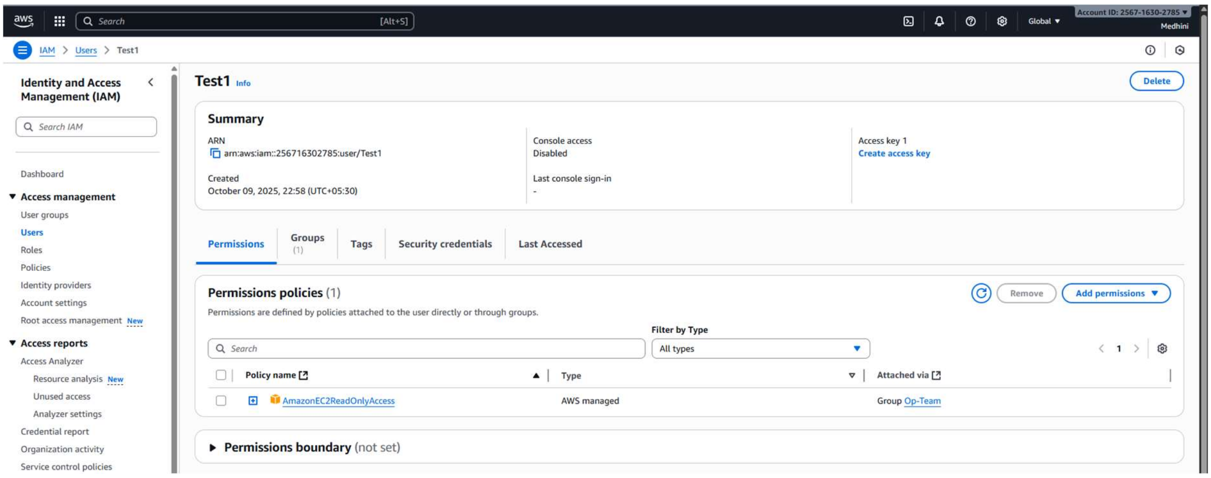
Tags - *optional*

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.

You can add up to 50 more tags.

Step 4: The user "**Test1**" has been added to "**Op-Team**"



Test1 Info

Summary

ARN
arn:aws:iam::256716302785:user/Test1

Created
October 09, 2025, 22:58 (UTC+05:30)

Console access
Disabled

Last console sign-in
-

Access key 1
Create access key

Permissions policies (1)

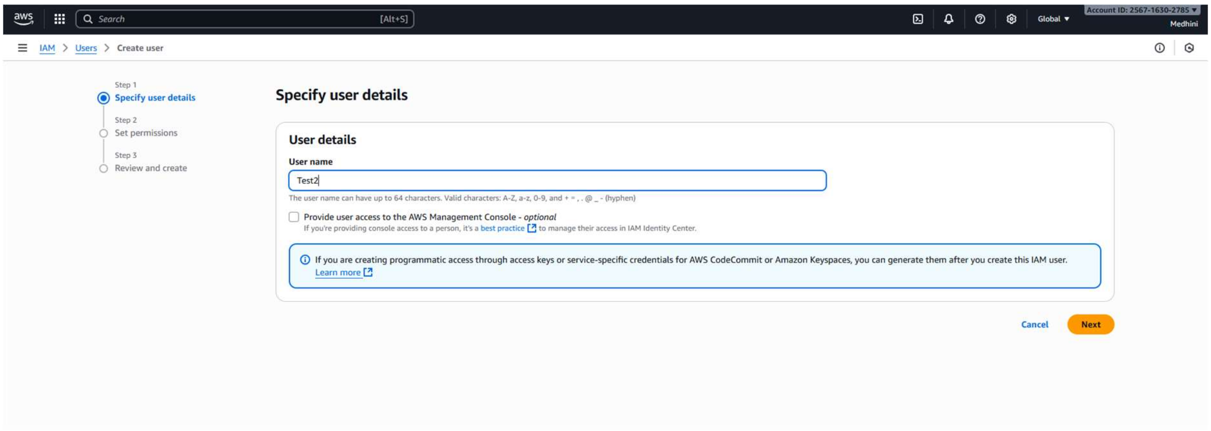
Permissions are defined by policies attached to the user directly or through groups.

Filter by Type: All types

Policy name	Type	Attached via
AmazonEC2ReadOnlyAccess	AWS managed	Group Op-Team

▸ **Permissions boundary** (not set)

Step 5: Give a name as "**Test2**"



Specify user details

User details

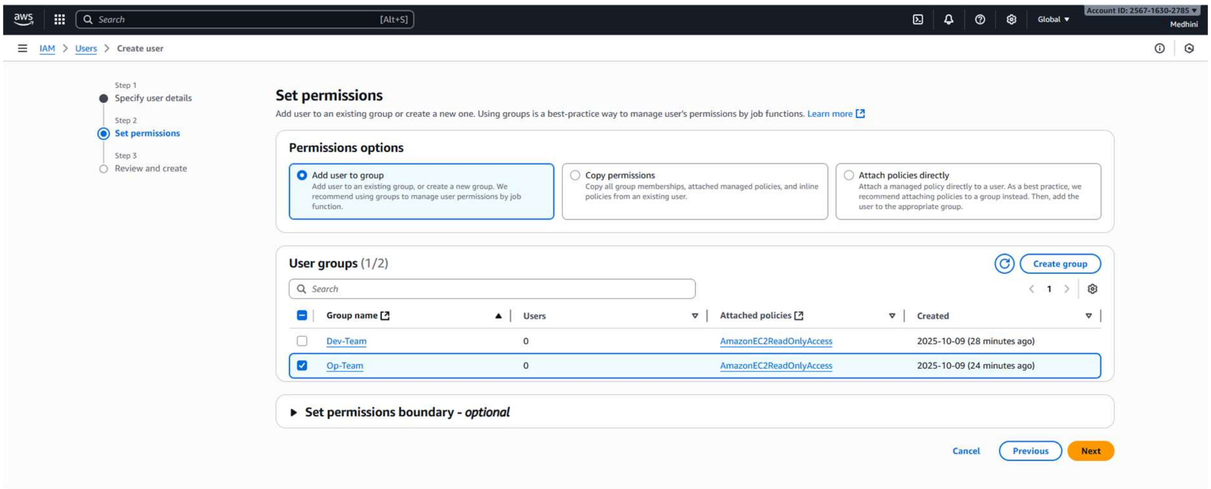
User name: Test2

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.

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

Step 6: Select “**Op-Team**” as a group and click on “**Next**”



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.

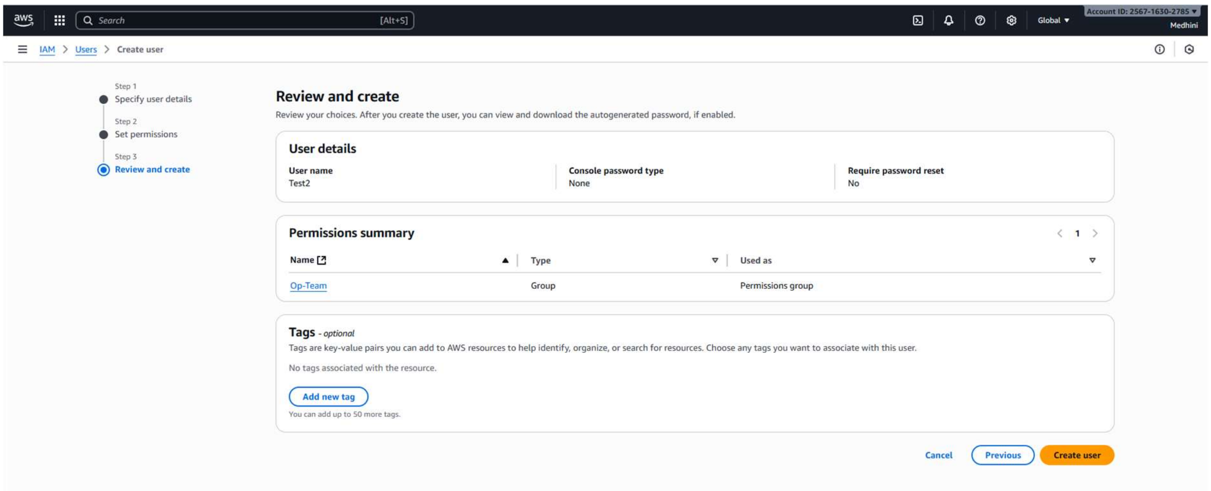
○ Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

User groups (1/2)

	Group name	Users	Attached policies	Created
□	Dev-Team	0	AmazonEC2ReadOnlyAccess	2025-10-09 (28 minutes ago)
☑	Op-Team	0	AmazonEC2ReadOnlyAccess	2025-10-09 (24 minutes ago)

▸ Set permissions boundary - *optional*

Step 7: Click on “**Create user**”



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
Test2	None	No

Permissions summary

Name	Type	Used as
Op-Team	Group	Permissions group

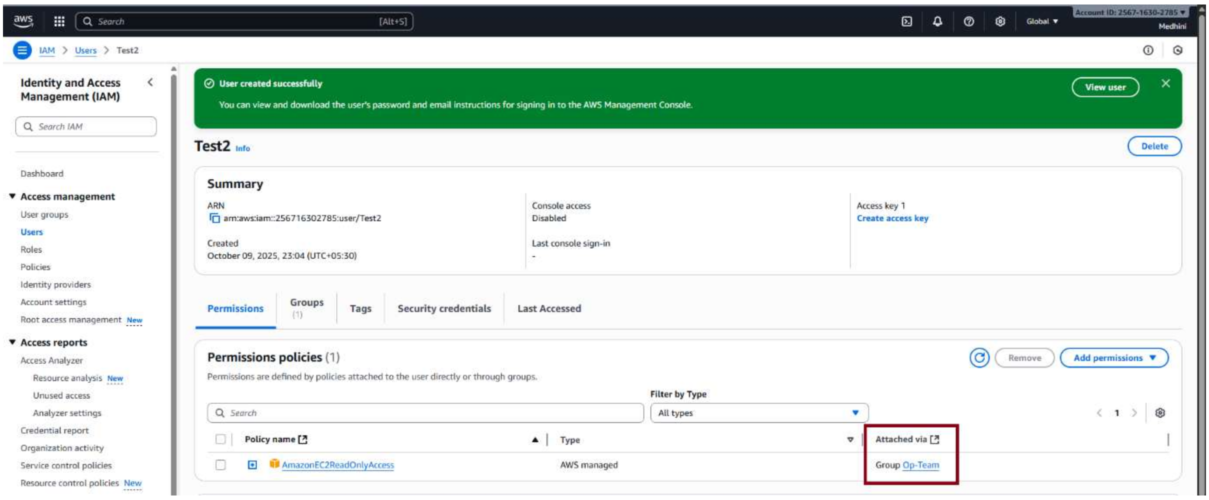
Tags - *optional*

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.

You can add up to 50 more tags.

Step 8: “**Test2**” is added to group “**Op-Team**”



User created successfully
You can view and download the user's password and email instructions for signing in to the AWS Management Console.

Summary

ARN	Console access	Access key 1
arn:aws:iam::256716302785:user/Test2	Disabled	Create access key

Created
October 09, 2025, 23:04 (UTC+05:30)

Last console sign-in
-

Permissions policies (1)

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

Filter by Type: All types

	Policy name	Type	Attached via
□	AmazonEC2ReadOnlyAccess	AWS managed	Group Op-Team