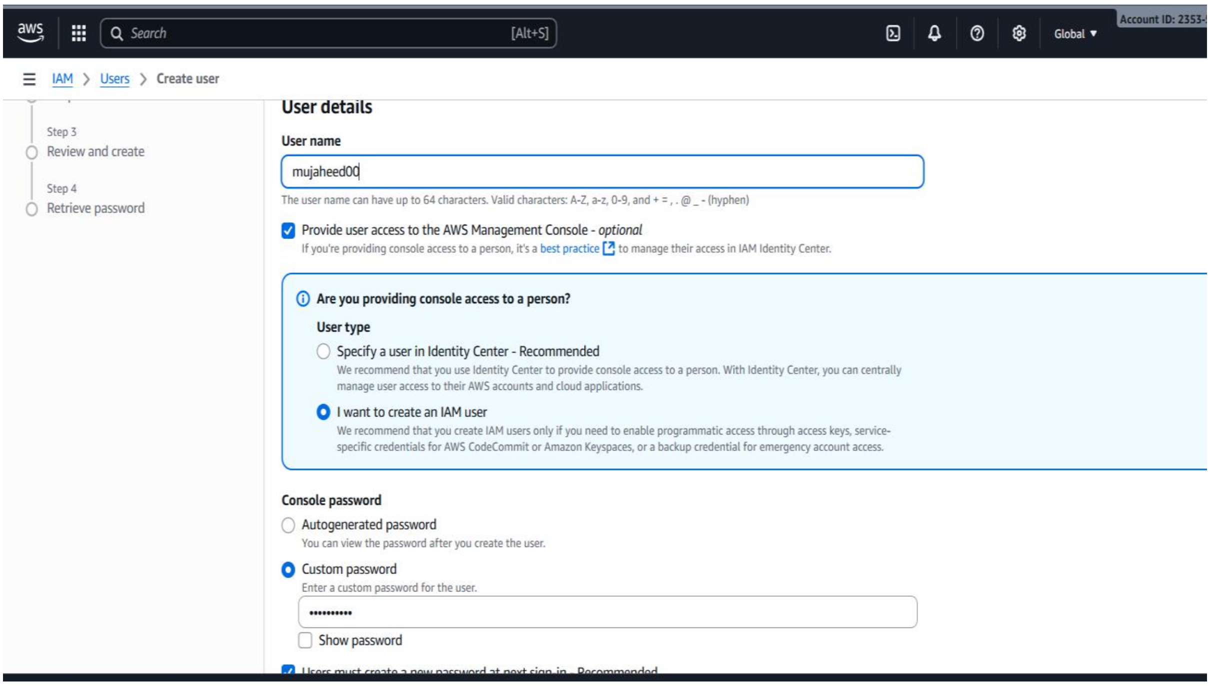
1. Create one IAM user and assign EC2 and S3 full access roles.

Go to IAM and select user, create user.



User details

User name

mujaheed00

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.

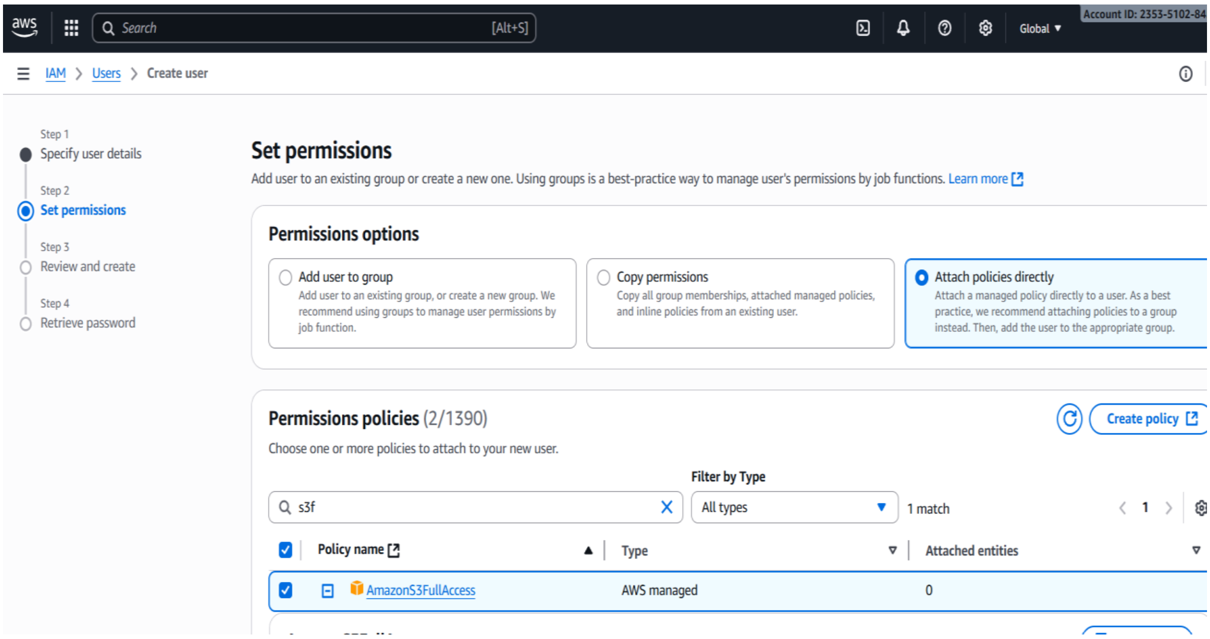
● Custom password
Enter a custom password for the user.

••••••••••

□ Show password

☑ Users must create a new password at next sign-in - Recommended

Select ec2fullaccess,s3fullacess.



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.

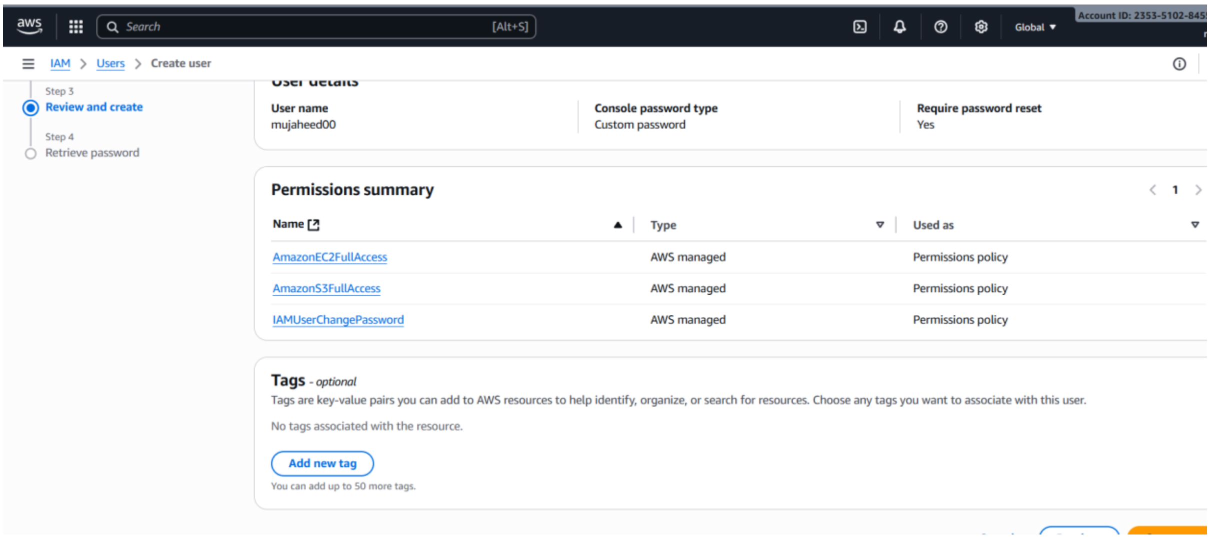
Permissions policies (2/1390)

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

Filter by Type

s3f | All types | 1 match

☑	Policy name	Type	Attached entities
☑	AmazonS3FullAccess	AWS managed	0



User name	Console password type	Require password reset
mujaheed00	Custom password	Yes

Permissions summary

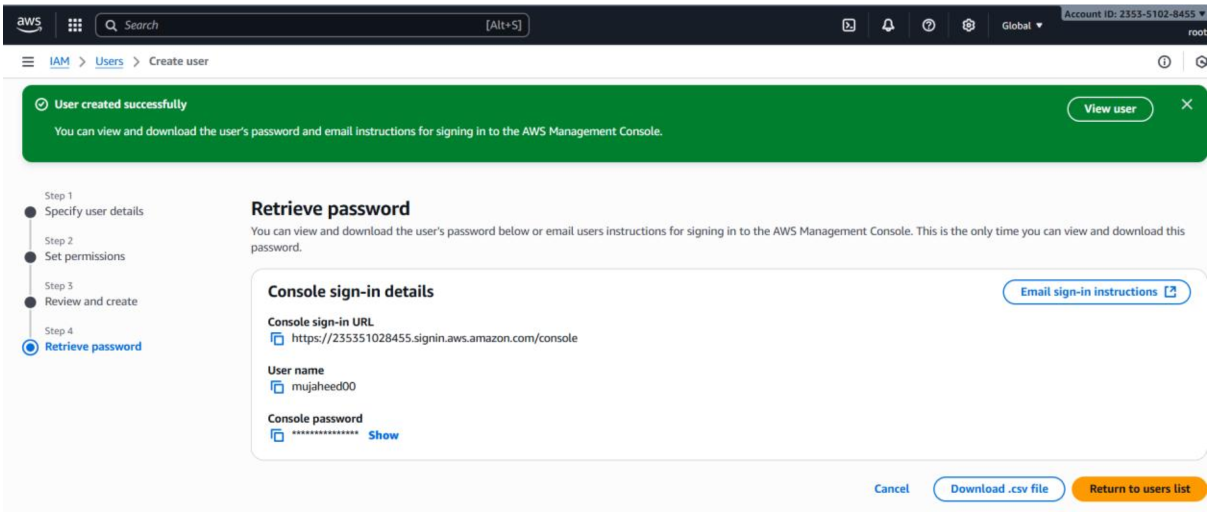
Name	Type	Used as
AmazonEC2FullAccess	AWS managed	Permissions policy
AmazonS3FullAccess	AWS managed	Permissions policy
IAMUserChangePassword	AWS managed	Permissions policy

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.

You can add up to 50 more tags.



User created successfully

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

Retrieve password

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details

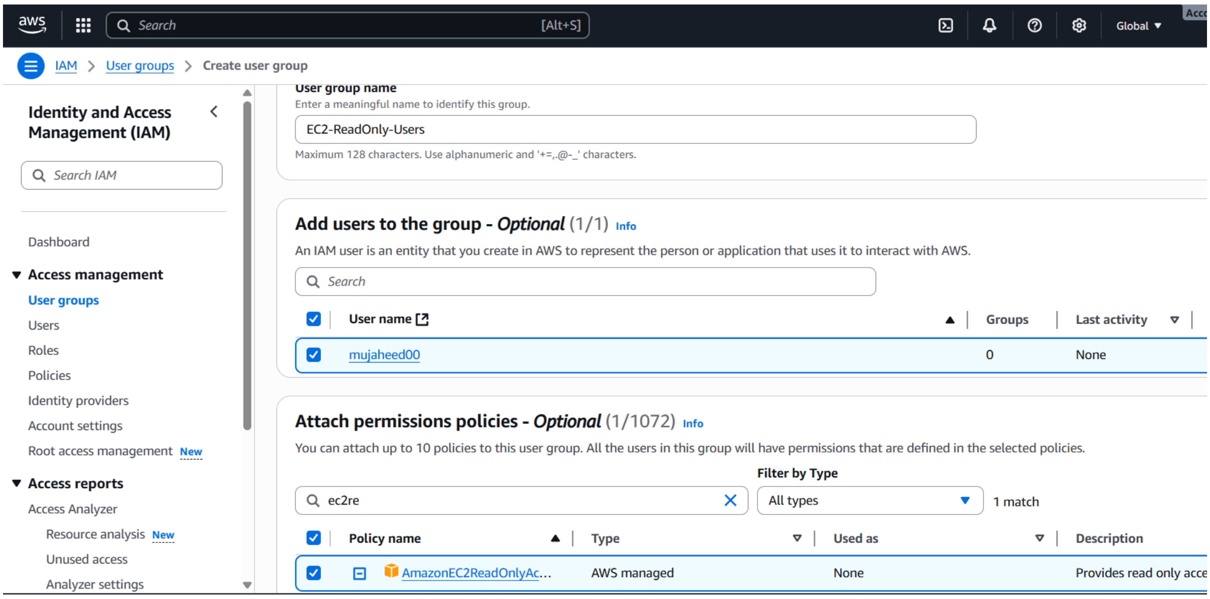
Console sign-in URL
https://235351028455.signin.aws.amazon.com/console

User name
mujaheed00

Console password
****

2. Create one group in IAM and assign read access for EC2.

Click on user groups create an usergroup give the name and add the user . attach permissions for only ec2readonlyaccess and create.



User group name

Enter a meaningful name to identify this group.

EC2-ReadOnly-Users

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

Add users to the group - *Optional* (1/1) 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
☑	mujaheed00	0	None

Attach permissions policies - *Optional* (1/1072) Info

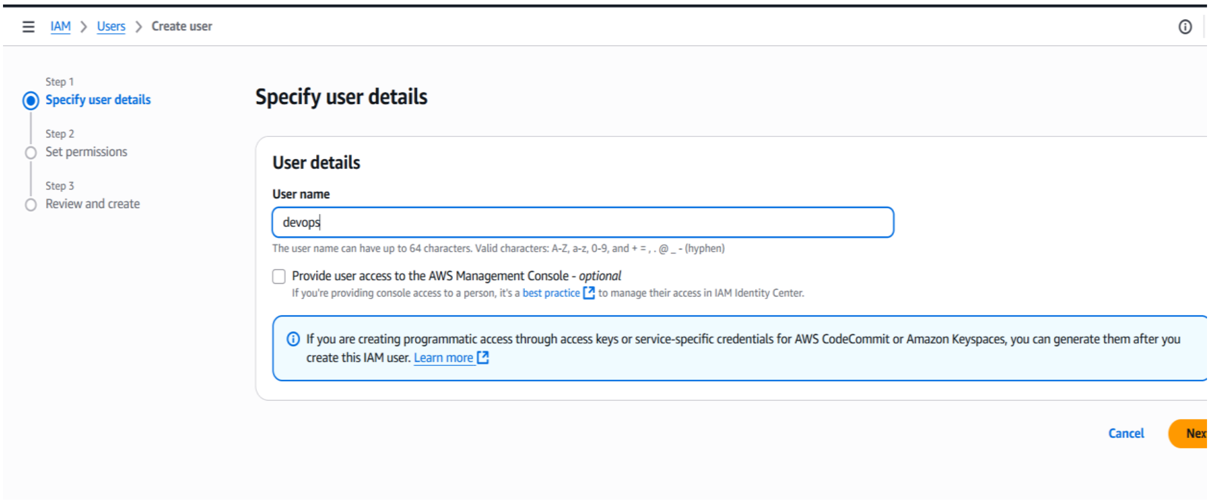
You can attach up to 10 policies to this user group. All the users in this group will have permissions that are defined in the selected policies.

ec2re — Filter by Type: All types — 1 match

	Policy name	Type	Used as	Description
☑	AmazonEC2ReadOnlyAc...	AWS managed	None	Provides read only acce

3. Create a new user named "Devops" and add to the group created in task 2.

Click on create user and give name as Devops.



Specify user details

User details

User name

devops

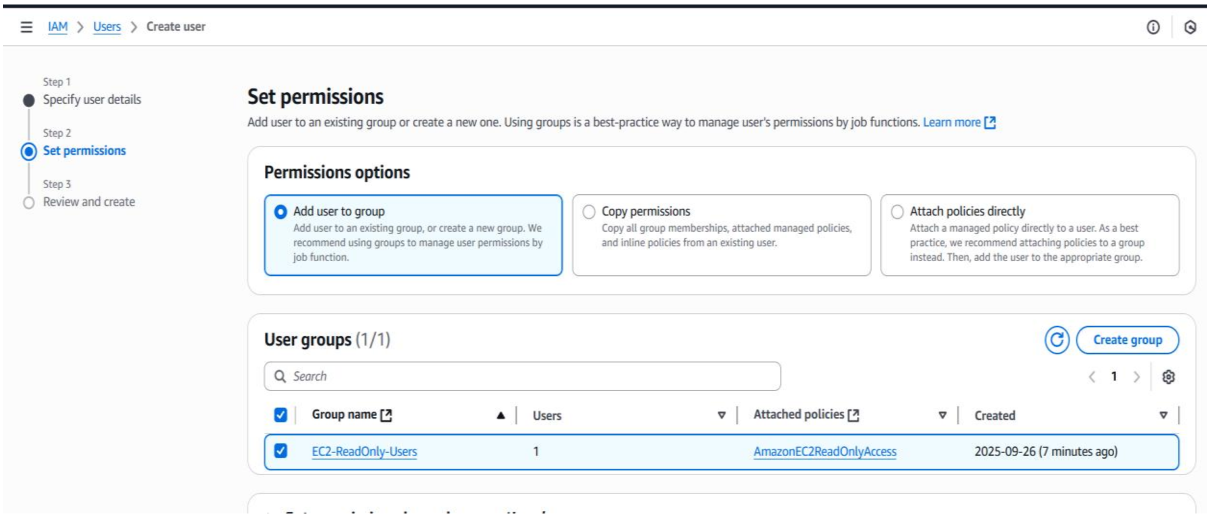
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

□ 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

In permissions give add user to the group and select the group.



IAM > Users > Create user

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 group. 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.

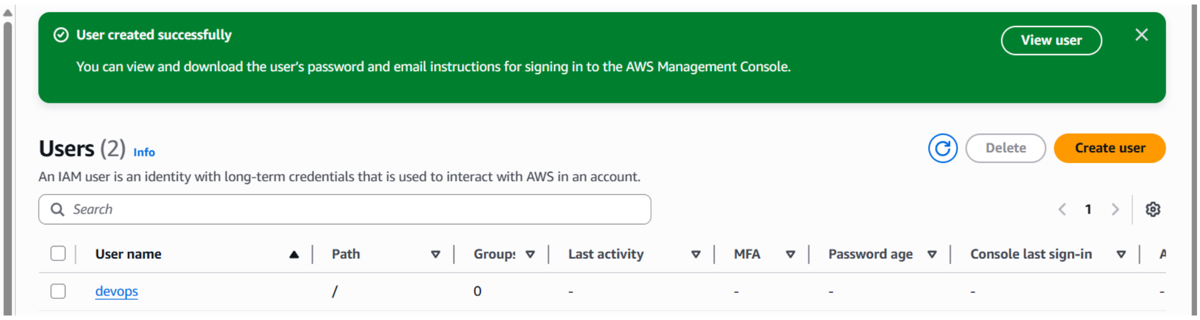
○ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

○ 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 (1/1)

Create group

☑	Group name	Users	Attached policies	Created
☑	EC2-ReadOnly-Users	1	AmazonEC2ReadOnlyAccess	2025-09-26 (7 minutes ago)



User created successfully

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

View user

Users (2) Info

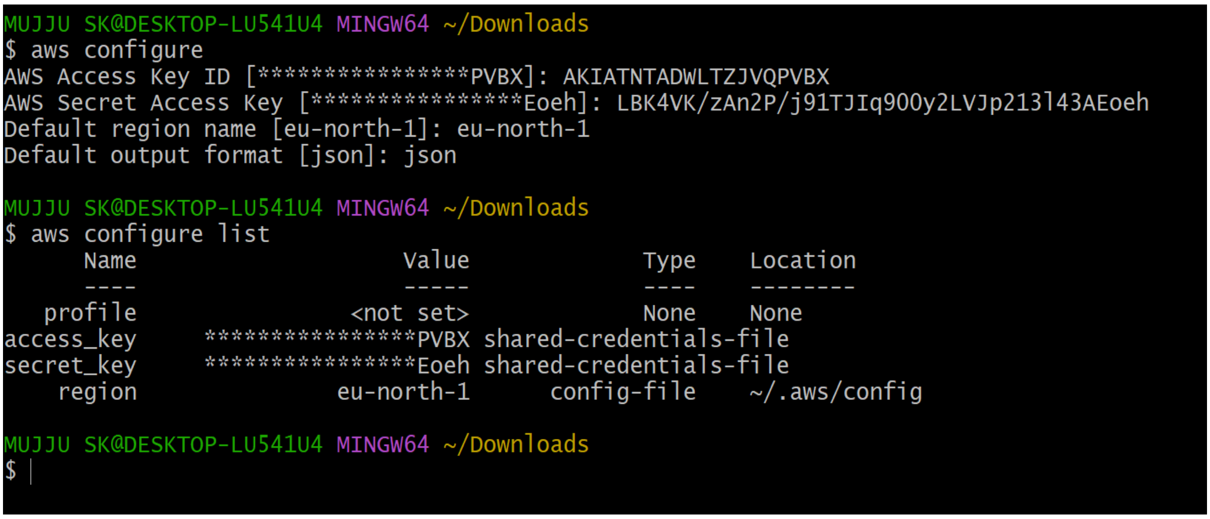
Delete | Create user

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	A
□	devops	/	0	-	-	-	-	-

4. Write a bash script to create an IAM user with VPC full access.

Configure AWS in CLI.

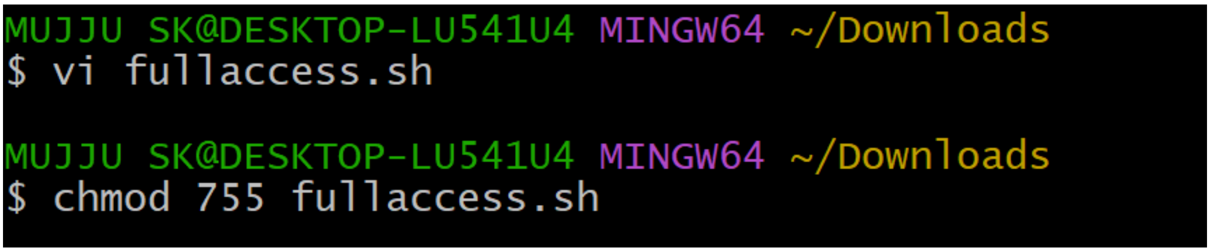


```
MUJJU SK@DESKTOP-LU541U4 MINGW64 ~/Downloads
$ aws configure
AWS Access Key ID [****************PVBX]: AKIATNTADWLTZJVQPVBX
AWS Secret Access Key [****************Eoeh]: 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
Default region name [eu-north-1]: eu-north-1
Default output format [json]: json

MUJJU SK@DESKTOP-LU541U4 MINGW64 ~/Downloads
$ aws configure list
      Name                    Value             Type    Location
      ----                    -----             ----    --------
   profile                <not set>             None    None
access_key     ****************PVBX shared-credentials-file
secret_key     ****************Eoeh shared-credentials-file
    region               eu-north-1      config-file    ~/.aws/config

MUJJU SK@DESKTOP-LU541U4 MINGW64 ~/Downloads
$ |
```

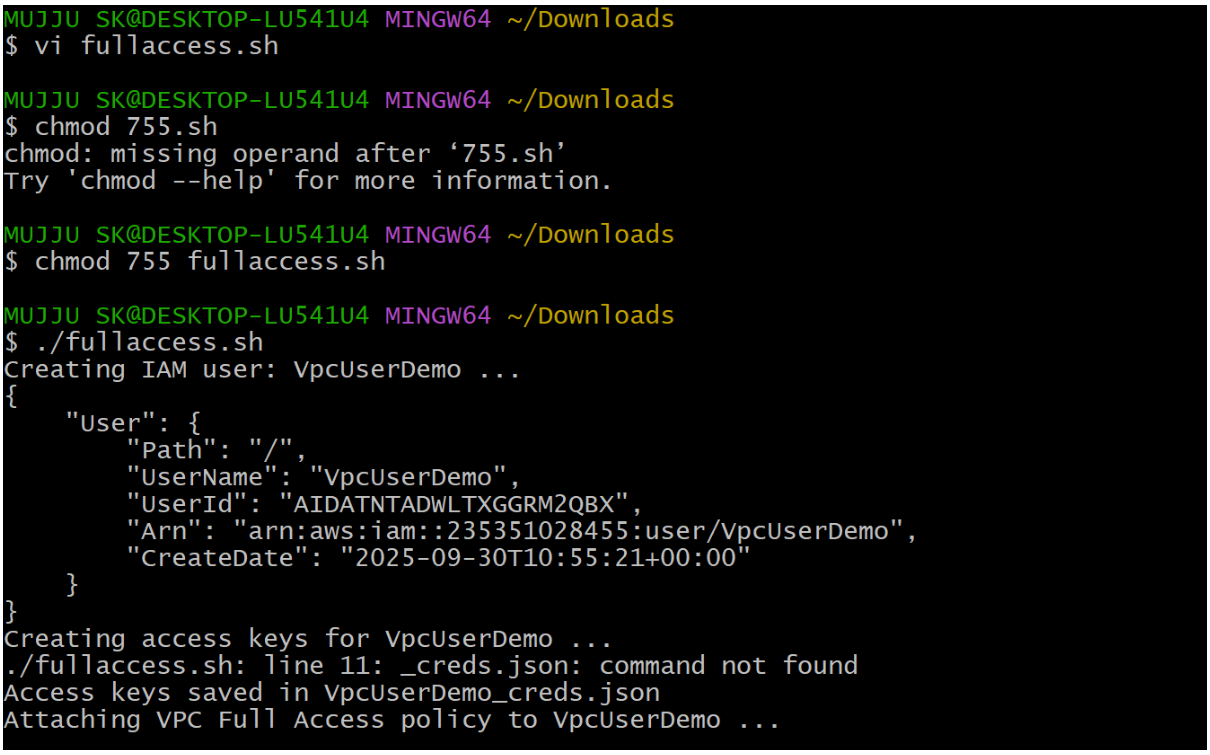
Enter the bash script.

```
##!/bin/bash
# ============ VARIABLES ============
IAM_USER="VpcUserDemo"
POLICY_ARN="arn:aws:iam::aws:policy/AmazonVPCFullAcce"
# ============ CREATE IAM USER ============
echo "Creating IAM user: $IAM_USER ..."
aws iam create-user --user-name $IAM_USER
# ============ CREATE ACCESS KEYS ============
echo "Creating access keys for $IAM_USER ..."
aws iam create-access-key --user-name $IAM_USER >
${IAM_USER}_creds.json
echo "Access keys saved in ${IAM_USER}_creds.json"
# ============ ATTACH POLICY ============
echo "Attaching VPC Full Access policy to $IAM_USER ..."
aws iam attach-user-policy --user-name $IAM_USER policyarn
$POLICY_ARN
echo "User $IAM_USER created successfully with VPC Full
Access."
```

Write script and give the permissions as chmod 755.



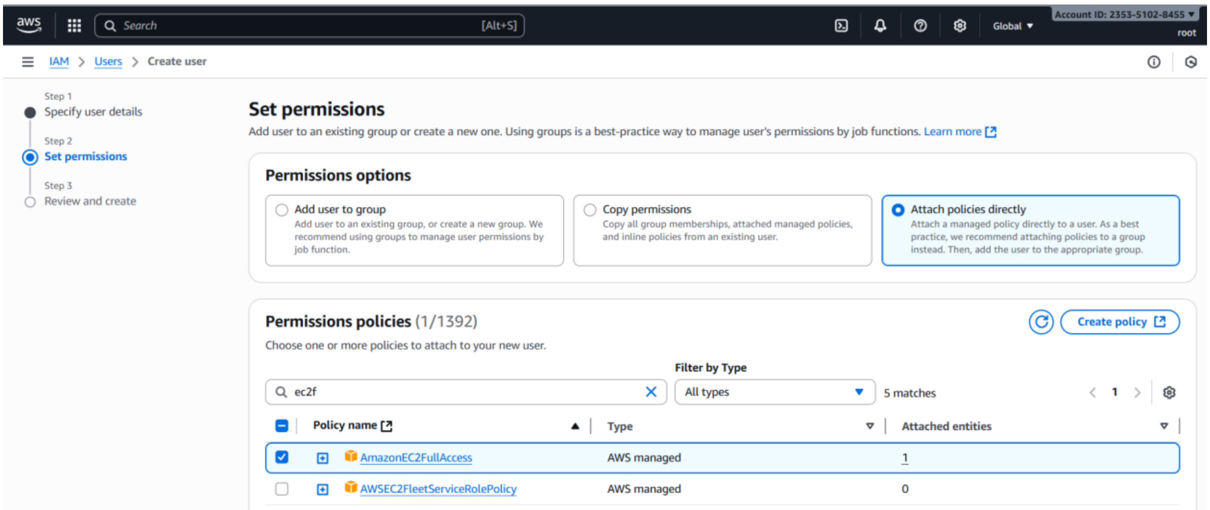
```
MUJJU SK@DESKTOP-LU541U4 MINGW64 ~/Downloads
$ vi fullaccess.sh

MUJJU SK@DESKTOP-LU541U4 MINGW64 ~/Downloads
$ chmod 755 fullaccess.sh
```



```
MUJJU SK@DESKTOP-LU541U4 MINGW64 ~/Downloads
$ vi fullaccess.sh

MUJJU SK@DESKTOP-LU541U4 MINGW64 ~/Downloads
$ chmod 755.sh
chmod: missing operand after ‘755.sh’
Try 'chmod --help' for more information.

MUJJU SK@DESKTOP-LU541U4 MINGW64 ~/Downloads
$ chmod 755 fullaccess.sh

MUJJU SK@DESKTOP-LU541U4 MINGW64 ~/Downloads
$ ./fullaccess.sh
Creating IAM user: VpcUserDemo ...
{
    "User": {
        "Path": "/",
        "UserName": "VpcUserDemo",
        "UserId": "AIDATNTADWLTXGGRM2QBX",
        "Arn": "arn:aws:iam::235351028455:user/VpcUserDemo",
        "CreateDate": "2025-09-30T10:55:21+00:00"
    }
}
Creating access keys for VpcUserDemo ...
./fullaccess.sh: line 11: _creds.json: command not found
Access keys saved in VpcUserDemo_creds.json
Attaching VPC Full Access policy to VpcUserDemo ...
```

5. Create an IAM policy to allow EC2 access for a specific user in specific regions only.

I created an IAM user name, EC2-Region-Restrict-Policy and give amazon ec2fullacess for specific user for specific region.



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.

Permissions policies (1/1392)

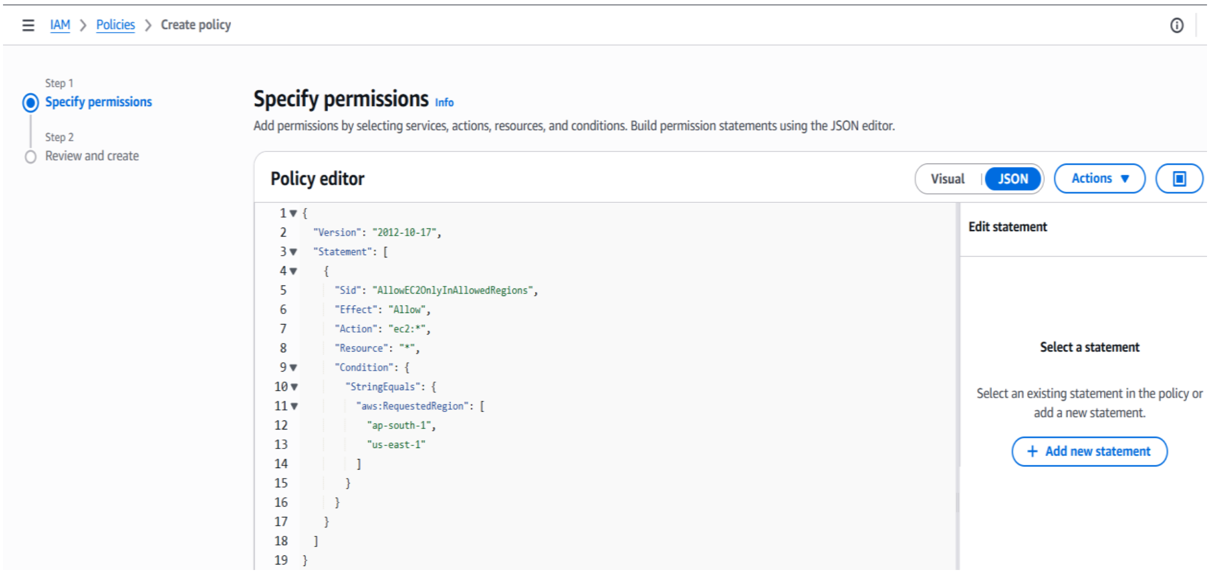
Choose one or more policies to attach to your new user.

Filter by Type

ec2f — All types ▼ — 5 matches — < 1 >

☐	Policy name	Type	Attached entities
☑	⊞ AmazonEC2FullAccess	AWS managed	1
☐	⊞ AWSEC2FleetServiceRolePolicy	AWS managed	0

Create permissions to allow only specific user to specific region.



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

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowEC2OnlyInAllowedRegions",
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": [
            "ap-south-1",
            "us-east-1"
          ]
        }
      }
    }
  ]
}
```

Edit statement

Select a statement

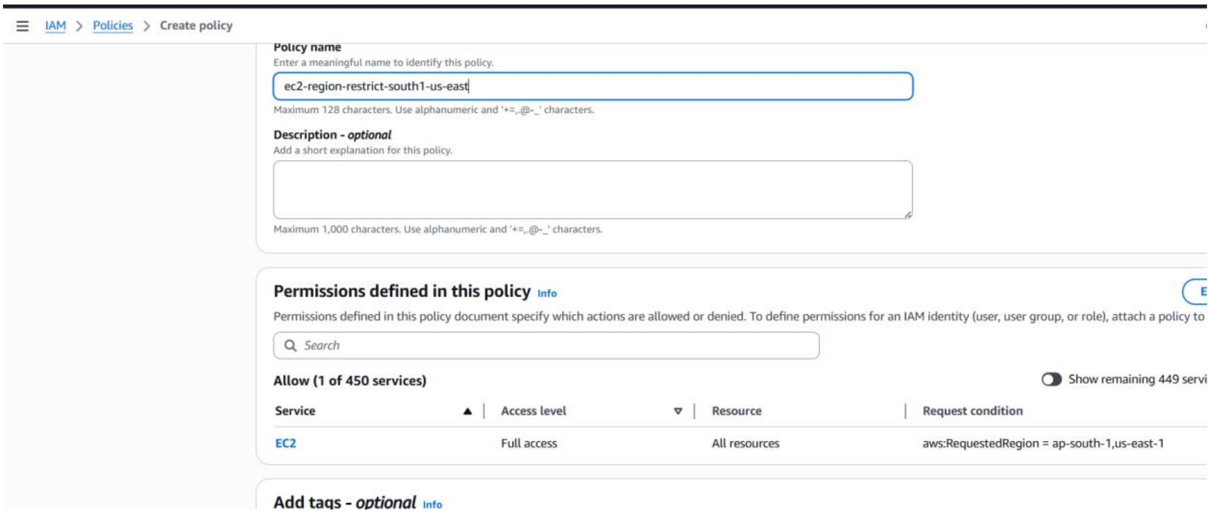
Select an existing statement in the policy or add a new statement.

As this a custom policy, we will be using JSON to write it from scratch.

1. Select the JSON tab to write the policy manually.This policy above will prevent users Launching EC2 instances in any region other than us-east-1.

2. Creating S3 buckets in any region other than us-east-1.

3.Give the Policy a name and click "create policy"



IAM > Policies > Create policy

Policy name

Enter a meaningful name to identify this policy.

ec2-region-restrict-south1-us-east

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

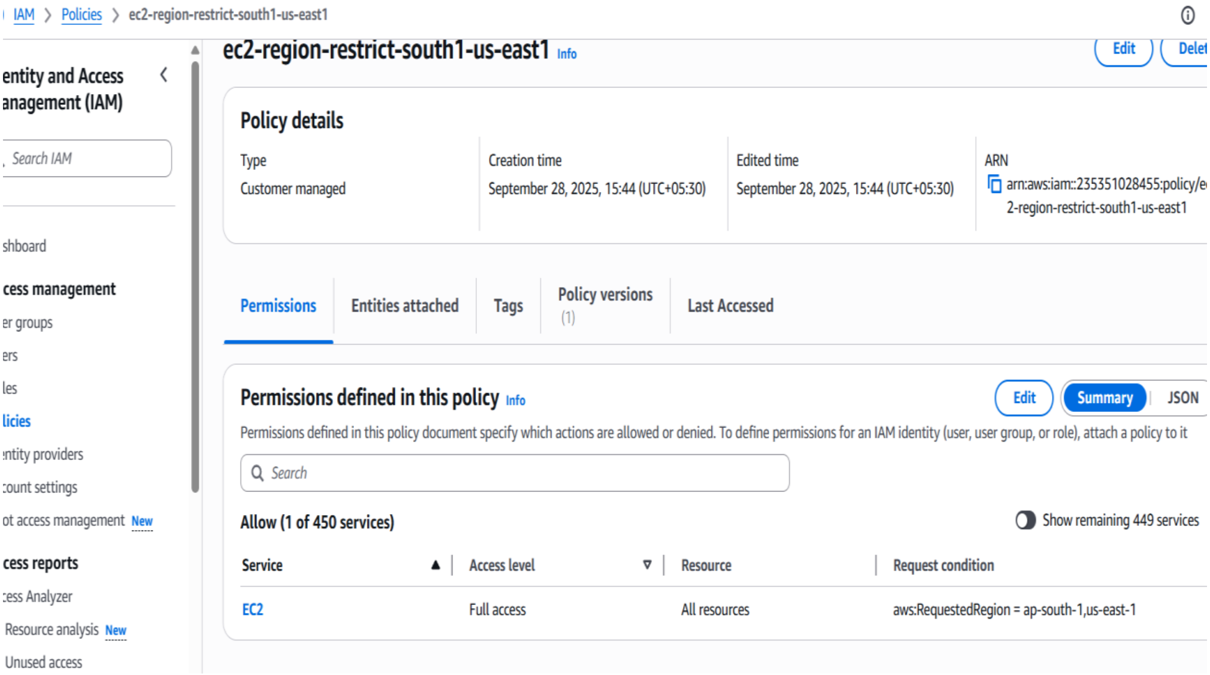
Allow (1 of 450 services)

Show remaining 449 servi

Service	Access level	Resource	Request condition
EC2	Full access	All resources	aws:RequestedRegion = ap-south-1,us-east-1

Add tags - *optional* Info

We have created the policy .



IAM > Policies > ec2-region-restrict-south1-us-east1

ec2-region-restrict-south1-us-east1 Info

Policy details

Type	Creation time	Edited time	ARN
Customer managed	September 28, 2025, 15:44 (UTC+05:30)	September 28, 2025, 15:44 (UTC+05:30)	arn:aws:iam::235351028455:policy/e 2-region-restrict-south1-us-east1

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

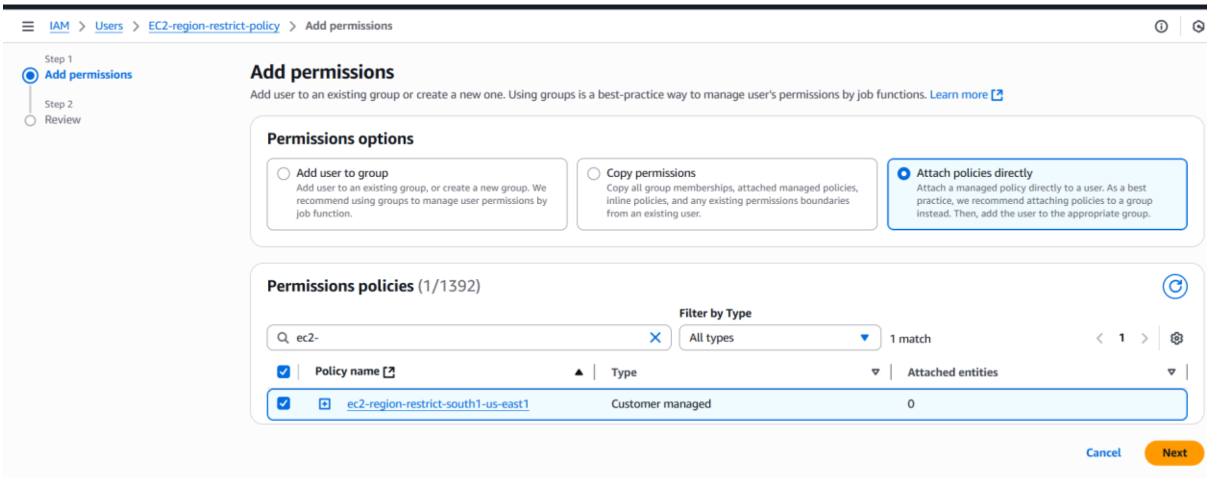
Allow (1 of 450 services)

Show remaining 449 services

Service	Access level	Resource	Request condition
EC2	Full access	All resources	aws:RequestedRegion = ap-south-1,us-east-1

Go to users and select the user what you need to add the specific policy that you are created.

Go to IAM → Users, click the target user, Permissions → Add permissions → Attach existing policies



IAM > Users > EC2-region-restrict-policy > Add permissions

Step 1
● Add permissions

Step 2
○ Review

Add 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, inline policies, and any existing permissions boundaries from an existing user.

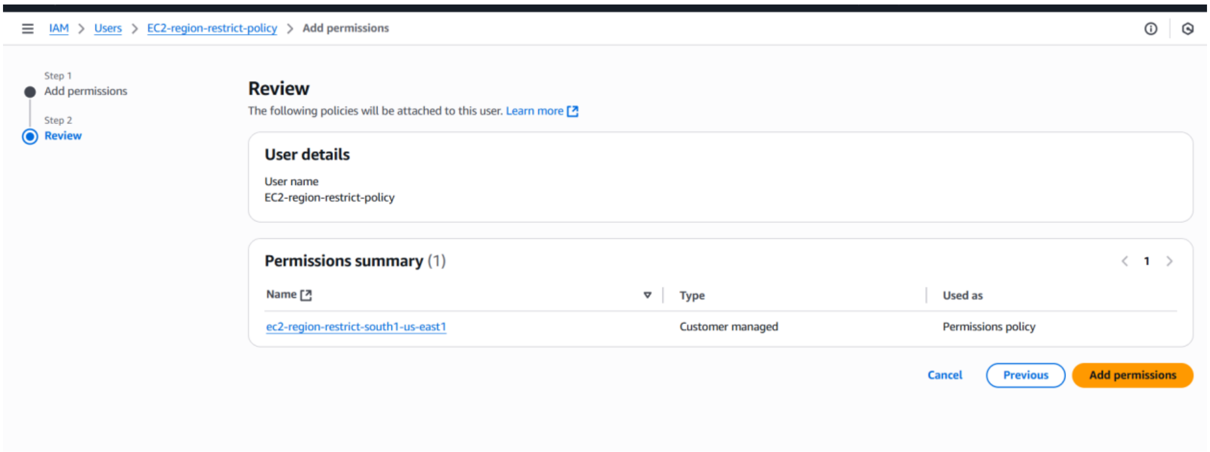
● Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1392)

Search: ec2-
Filter by Type: All types
1 match

☑	Policy name	Type	Attached entities
☑	ec2-region-restrict-south1-us-east1	Customer managed	0

Click next.



IAM > Users > EC2-region-restrict-policy > Add permissions

Step 1
● Add permissions

Step 2
● Review

Review

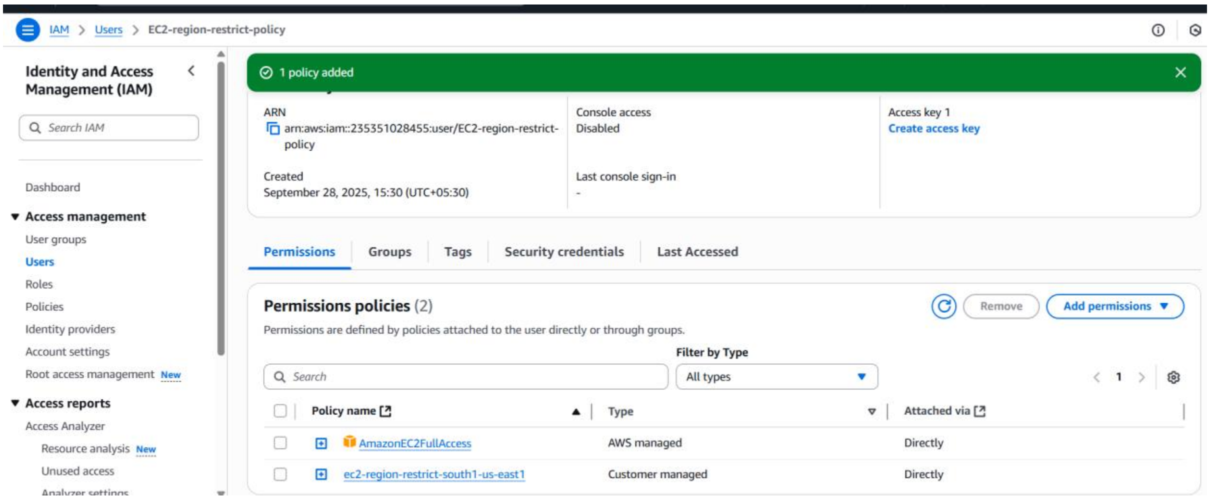
The following policies will be attached to this user. Learn more

User details

User name
EC2-region-restrict-policy

Permissions summary (1)

Name	Type	Used as
ec2-region-restrict-south1-us-east1	Customer managed	Permissions policy



IAM > Users > EC2-region-restrict-policy

1 policy added

ARN
arn:aws:iam::235351028455:user/EC2-region-restrict-policy

Console access
Disabled

Access key 1
Create access key

Created
September 28, 2025, 15:30 (UTC+05:30)

Last console sign-in
-

Permissions policies (2)

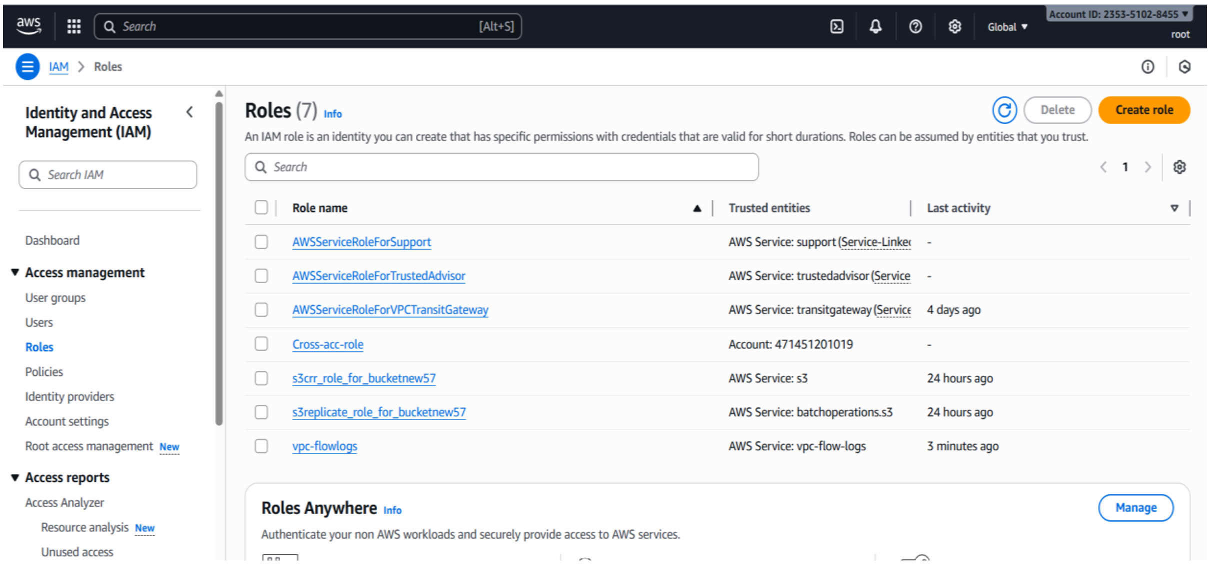
Permissions are defined by policies attached to the user directly or through groups.

Search
Filter by Type: All types

□	Policy name	Type	Attached via
□	AmazonEC2FullAccess	AWS managed	Directly
□	ec2-region-restrict-south1-us-east1	Customer managed	Directly

6. We have two accounts: Account A and Account B. Account A user should access an S3 bucket in Account B.

Create an IAM role in account A.



Roles (7) 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.

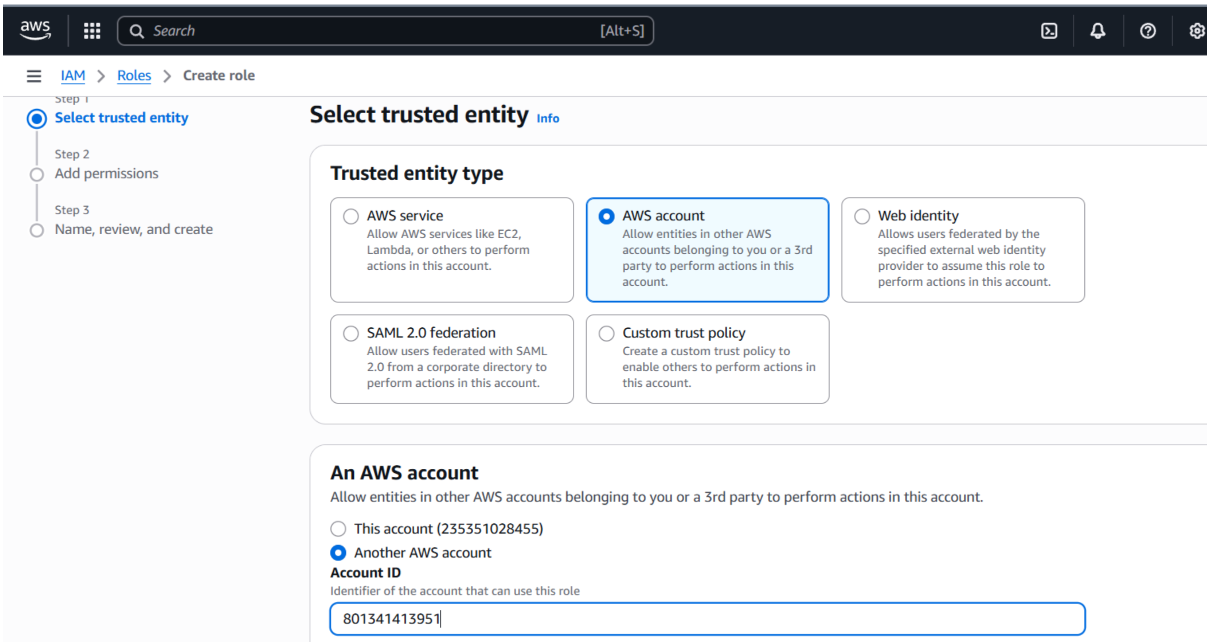
	Role name	Trusted entities	Last activity
□	AWSServiceRoleForSupport	AWS Service: support (Service-Linke	-
□	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-
□	AWSServiceRoleForVPCTransitGateway	AWS Service: transitgateway (Service	4 days ago
□	Cross-acc-role	Account: 471451201019	-
□	s3crr_role_for_bucketnew57	AWS Service: s3	24 hours ago
□	s3replicate_role_for_bucketnew57	AWS Service: batchoperations.s3	24 hours ago
□	vpc-flowlogs	AWS Service: vpc-flow-logs	3 minutes ago

Roles Anywhere Info

Manage

Authenticate your non AWS workloads and securely provide access to AWS services.

Give accout B’s account id in account B.



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.

An AWS account

Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ This account (235351028455)

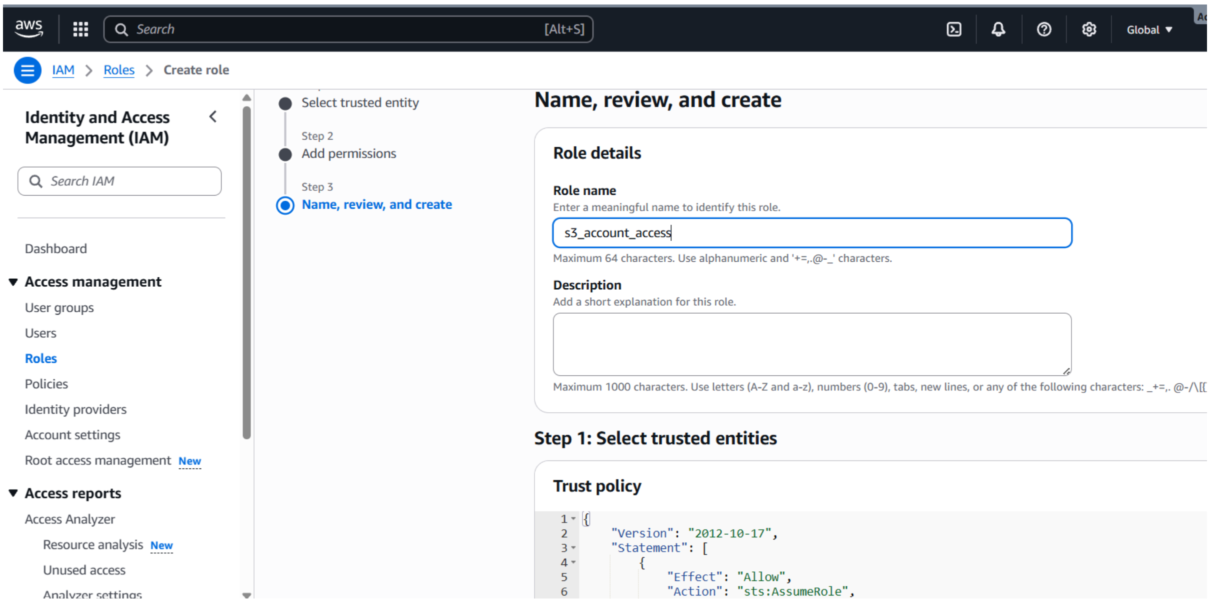
● Another AWS account

Account ID

Identifier of the account that can use this role

801341413951

Give the role name as s3_account_access and save it.



Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

s3_account_access

Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description

Add a short explanation for this role.

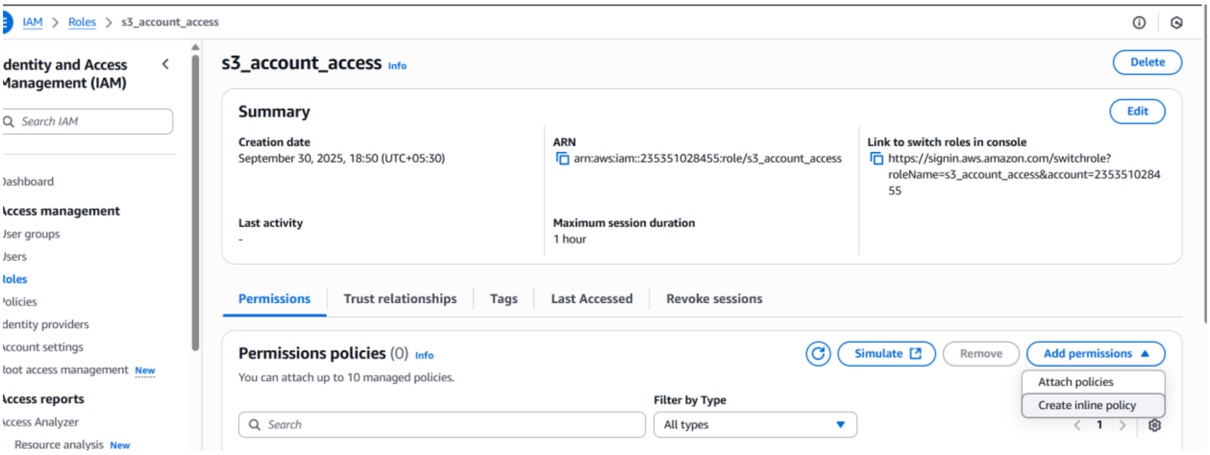
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/\[[

Step 1: Select trusted entities

Trust policy

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

Go to that created role and add permissions create inline policy.



s3_account_access Info

Summary

Creation date
September 30, 2025, 18:50 (UTC+05:30)

ARN
arn:aws:iam::235351028455:role/s3_account_access

Link to switch roles in console
https://signin.aws.amazon.com/switchrole?roleName=s3_account_access&account=235351028455

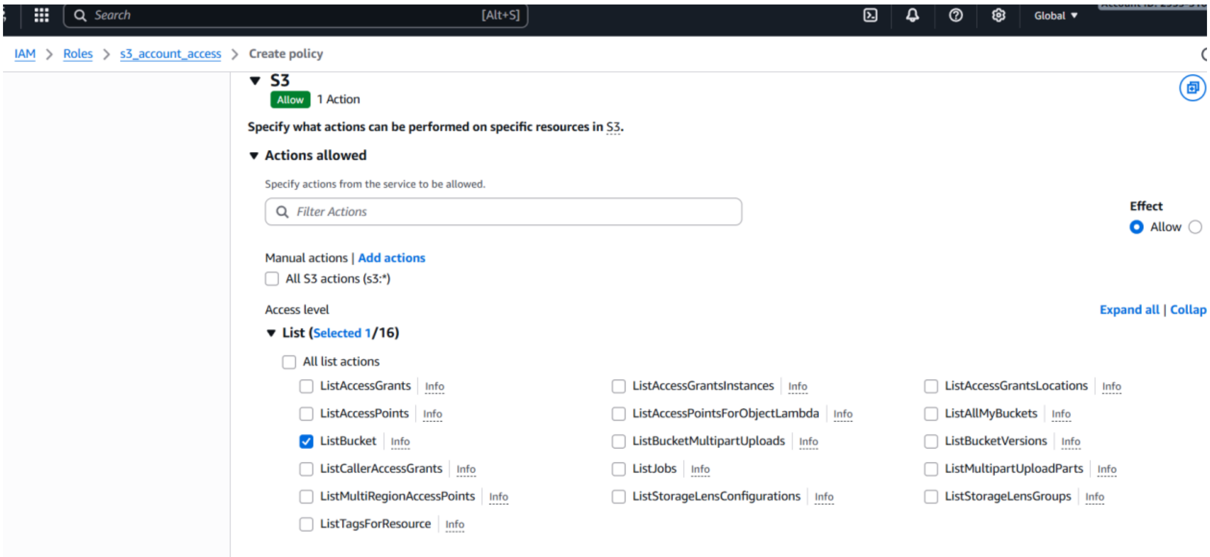
Last activity
-

Maximum session duration
1 hour

Permissions policies (0) Info

You can attach up to 10 managed policies.

Select S3 , select listbucket. In the bucket arn give the your bucket arn and add it.



▼ S3

Allow 1 Action

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

▼ Actions allowed

Specify actions from the service to be allowed.

Filter Actions

Effect
● Allow ○

Manual actions | Add actions

□ All S3 actions (s3:*)

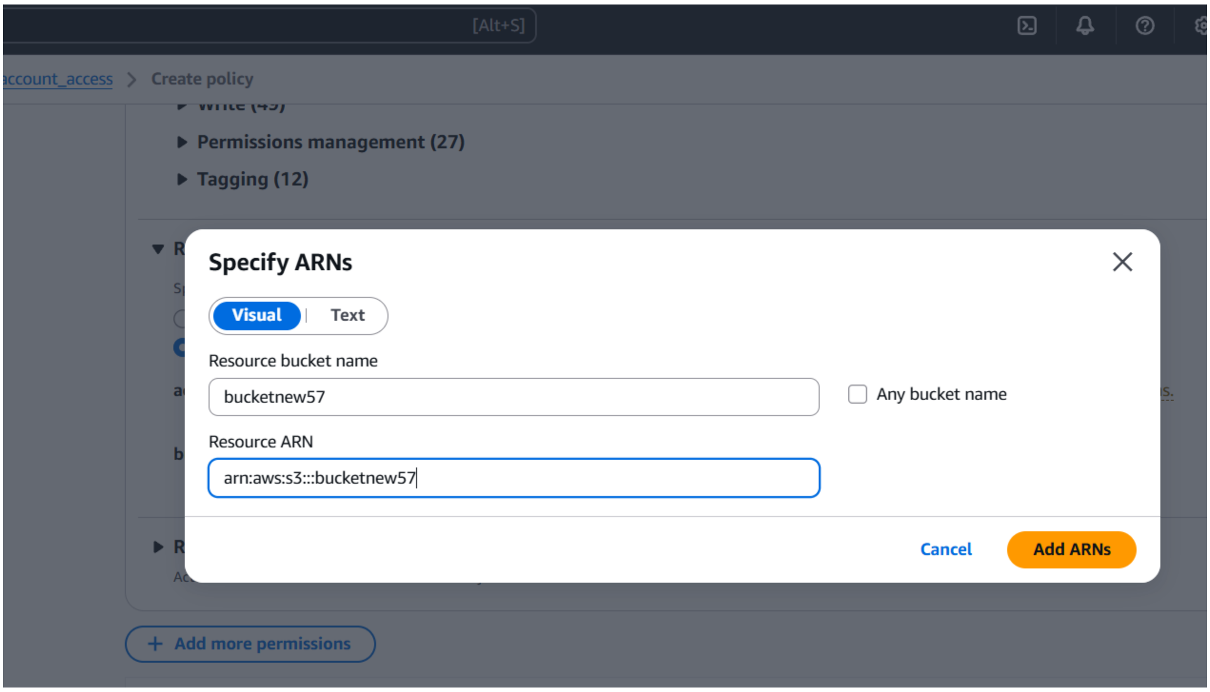
Access level

Expand all | Collap

▼ List (Selected 1/16)

□ All list actions

□ ListAccessGrants Info	□ ListAccessGrantsInstances Info	□ ListAccessGrantsLocations Info
□ ListAccessPoints Info	□ ListAccessPointsForObjectLambda Info	□ ListAllMyBuckets Info
☑ ListBucket Info	□ ListBucketMultipartUploads Info	□ ListBucketVersions Info
□ ListCallerAccessGrants Info	□ ListJobs Info	□ ListMultipartUploadParts Info
□ ListMultiRegionAccessPoints Info	□ ListStorageLensConfigurations Info	□ ListStorageLensGroups Info
□ ListTagsForResource Info		



Specify ARNs

Visual | Text

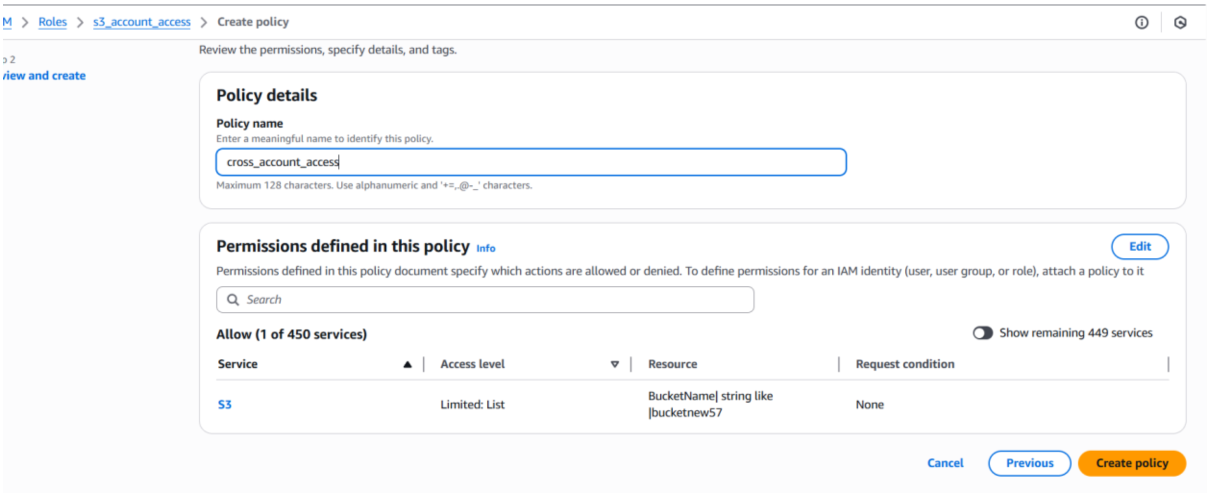
Resource bucket name: bucketnew57

□ Any bucket name

Resource ARN: arn:aws:s3:::bucketnew57

Cancel   Add ARNs

In the policy give the policy name as cross_account_access and create policy



M > Roles > s3_account_access > Create policy

Review the permissions, specify details, and tags.

Policy details

Policy name
Enter a meaningful name to identify this policy.
cross_account_access
Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

Permissions defined in this policy Info

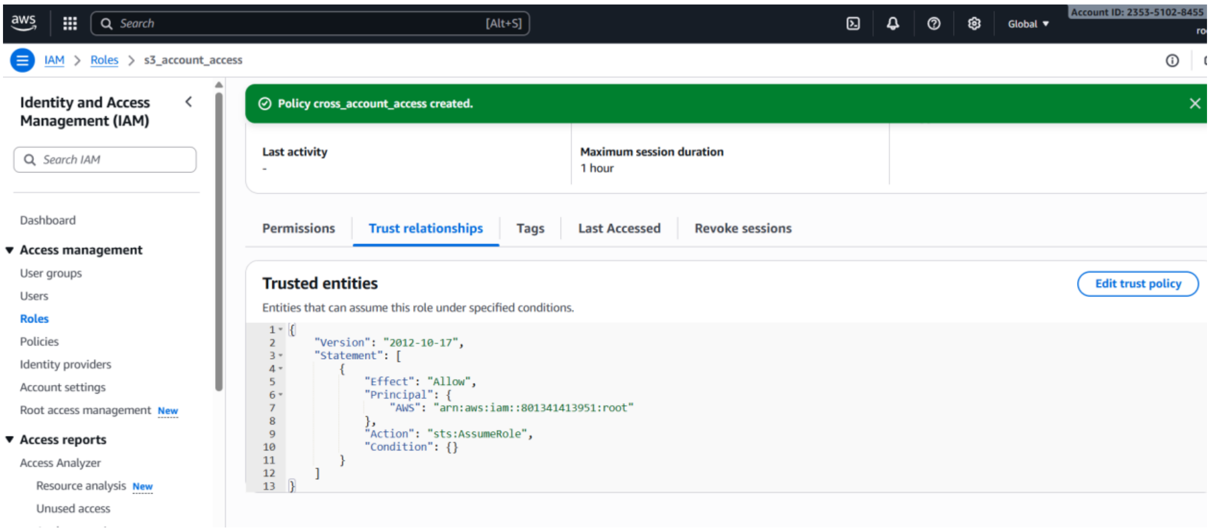
Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it

Allow (1 of 450 services)

Show remaining 449 services

Service	Access level	Resource	Request condition
S3	Limited: List	BucketName| string like |bucketnew57	None

Give the arn id of another account’s that who need to access your bucket.



Account ID: 2353-5102-8455

IAM > Roles > s3_account_access

Policy cross_account_access created.

Last activity
-

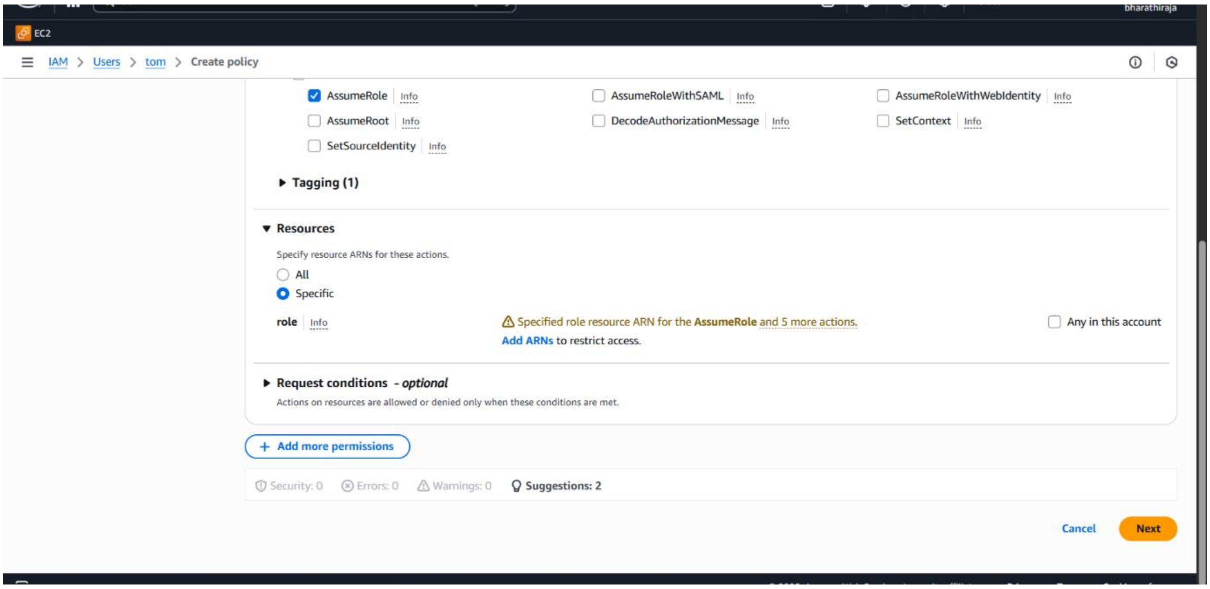
Maximum session duration
1 hour

Trusted entities

Entities that can assume this role under specified conditions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::801341413951:root"
            },
            "Action": "sts:AssumeRole",
            "Condition": {}
        }
    ]
}
```

Go to account b’s bucket and add the policy to assume the role go to permissions and create inline policy and choose STS. In the write section select assume role


IAM > Users > tom > Create policy

☑ AssumeRole Info □ AssumeRoleWithSAML Info □ AssumeRoleWithWebIdentity Info
□ AssumeRoot Info □ DecodeAuthorizationMessage Info □ SetContext Info
□ SetSourceIdentity Info

▸ Tagging (1)

▾ Resources

Specify resource ARNs for these actions.

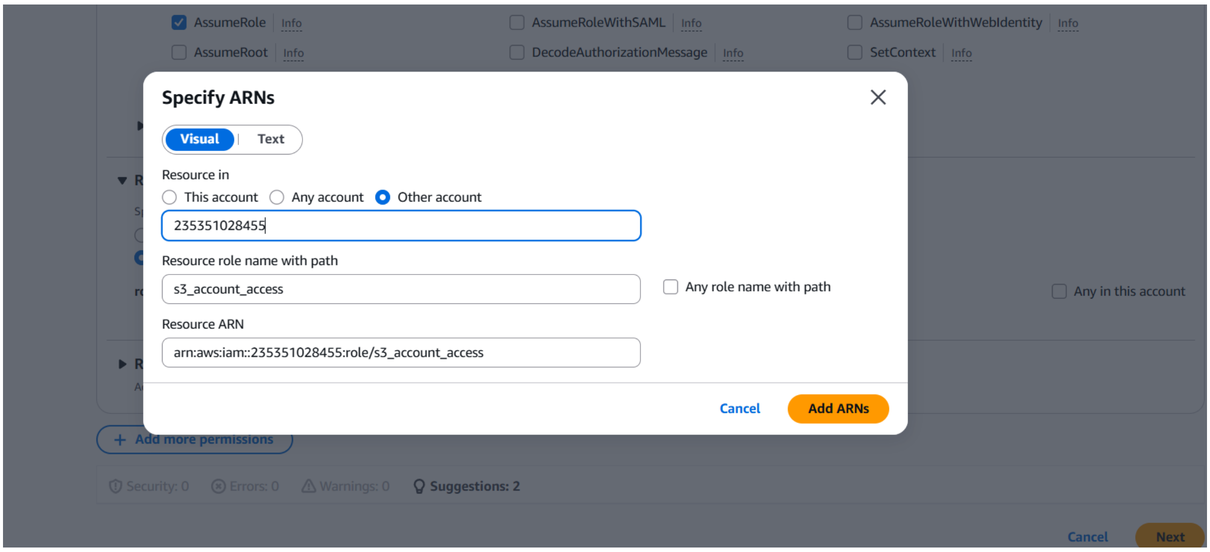
○ All
● Specific

role Info ⚠ Specified role resource ARN for the AssumeRole and 5 more actions. Add ARNs to restrict access. □ Any in this account

▸ Request conditions - *optional*

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

In the role arn give the role arn of A’s account .it will only generate the account id’s of A and role path.


Specify ARNs

Resource in

○ This account ○ Any account ● Other account

235351028455

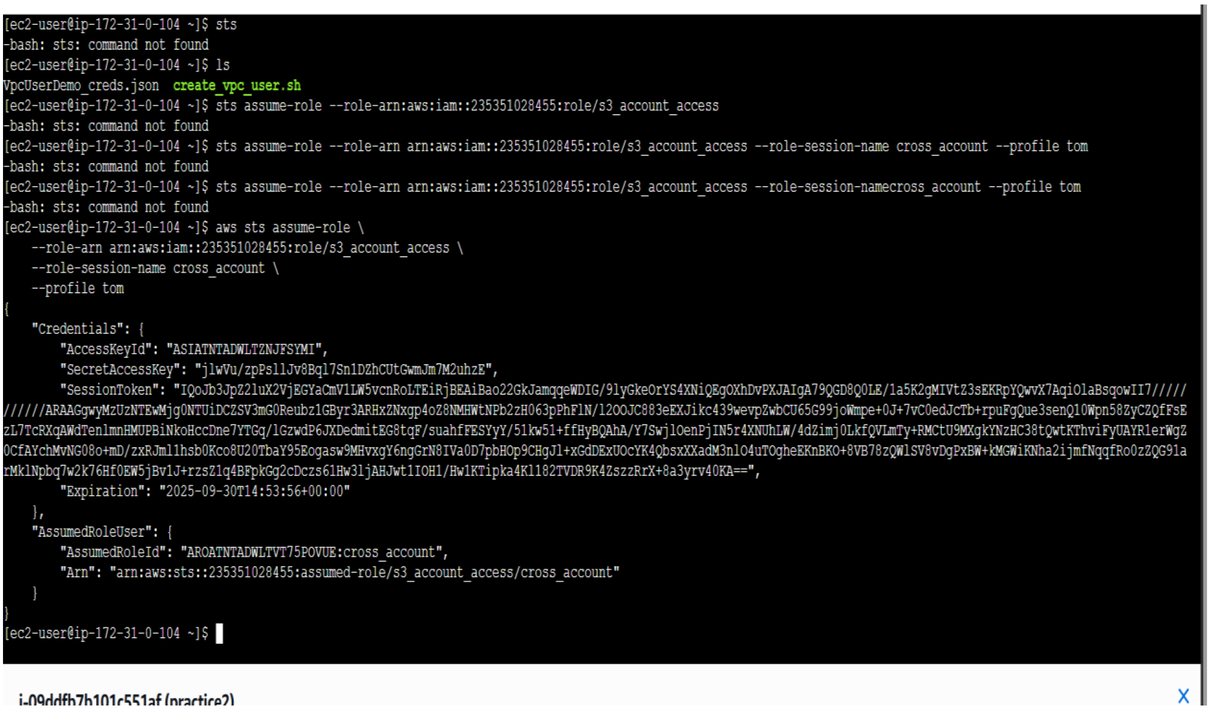
Resource role name with path

s3_account_access □ Any role name with path

Resource ARN

arn:aws:iam::235351028455:role/s3_account_access

Execute with the command: sts assume-role\ --role-arn <role arn of the account B’s policy>s3_account_access –role-session-namecross_account --profile-tom



```
[ec2-user@ip-172-31-0-104 ~]$ sts
-bash: sts: command not found
[ec2-user@ip-172-31-0-104 ~]$ ls
VpcUserDemo_creds.json  create_vpc_user.sh
[ec2-user@ip-172-31-0-104 ~]$ sts assume-role --role-arn:aws:iam::235351028455:role/s3_account_access
-bash: sts: command not found
[ec2-user@ip-172-31-0-104 ~]$ sts assume-role --role-arn arn:aws:iam::235351028455:role/s3_account_access --role-session-name cross_account --profile tom
-bash: sts: command not found
[ec2-user@ip-172-31-0-104 ~]$ sts assume-role --role-arn arn:aws:iam::235351028455:role/s3_account_access --role-session-namecross_account --profile tom
-bash: sts: command not found
[ec2-user@ip-172-31-0-104 ~]$ aws sts assume-role \
    --role-arn arn:aws:iam::235351028455:role/s3_account_access \
    --role-session-name cross_account \
    --profile tom
{
    "Credentials": {
        "AccessKeyId": "ASIATNTADWLTZNJFSYMI",
        "SecretAccessKey": "jlwVu/zpPsllJv8Bql7Sn1DZhCUtGwmJm7M2uhzE",
        "SessionToken": "IQoJb3JpZ2luX2VjEGYaCmV1LW5vcnRoLTEiRjBEAiBao22GkJamqqeWDIG/9lyGkeOrYS4XNiQEgOXhDvPXJAIgA79QGD8Q0LE/1a5K2gMIVtZ3sEKRpYQwvX7AqiOlaBsqowII7/////
//////ARAAGgwyMzUzNTEwMjg0NTUiDCZSV3mG0Reubz1GByr3ARHxZNxgp4oZ8NMHWtNPb2zH063pPhFlN/l2OOJC883eEXJikc439wevpZwbCU65G99joWmpe+0J+7vC0edJcTb+rpuFgQue3senQ10Wpn58ZyCZQfFsE
zL7TcRXqAWdTenlmnHMUPBiNkoHccDne7YTGq/lGzwdP6JXDedmitEG8tqF/suahfFESYyY/51kw51+ffHyBQAhA/Y7Swjl0enPjIN5r4XNUhLW/4dZimj0LkfQVLmTy+RMCtU9MXgkYNzHC38tQwtKThviFyUAYRlerWgZ
0CfAYchMvNG08o+mD/zxRJml1hsb0Kco8U20TbaY95Eogasw9MHvxgY6ngGrN8IVa0D7pbHOp9CHgJl+xGdDExUOcYK4QbsxXXadM3nlO4uTOgheEKnBKO+8VB78zQWlSV8vDgPxBW+kMGWiKNha2ijmfNqqfRo0zZQG91a
rMklNpbq7w2k76Hf0EW5jBv1J+rzsZ1q4BFpkGg2cDczs61Hw3ljAHJwt1IOH1/Hw1KTipka4Kl182TVDR9K4ZszzRrX+8a3yrv40KA==",
        "Expiration": "2025-09-30T14:53:56+00:00"
    },
    "AssumedRoleUser": {
        "AssumedRoleId": "AROATNTADWLTVT75POVUE:cross_account",
        "Arn": "arn:aws:sts::235351028455:assumed-role/s3_account_access/cross_account"
    }
}
[ec2-user@ip-172-31-0-104 ~]$ 
```

i-09ddfb7b101c551af (practice2)