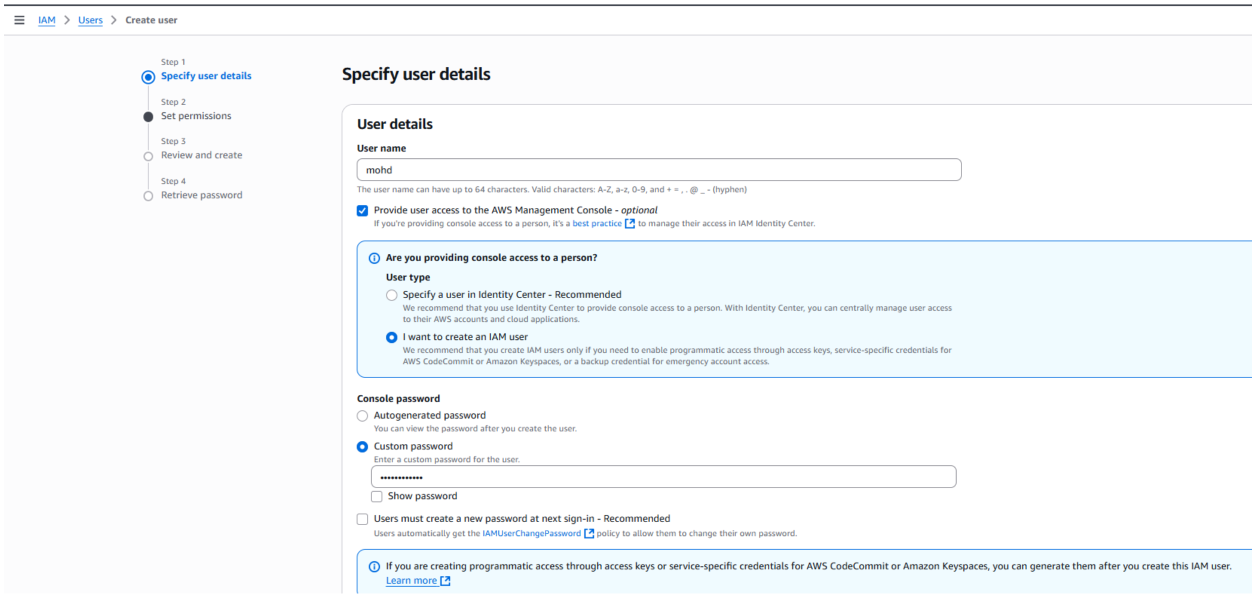
1. Create one IAM user and assign EC2 and S3 full access roles.

Create IAM user:



IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Step 4
Retrieve password

Specify user details

User details

User name

mohd

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.

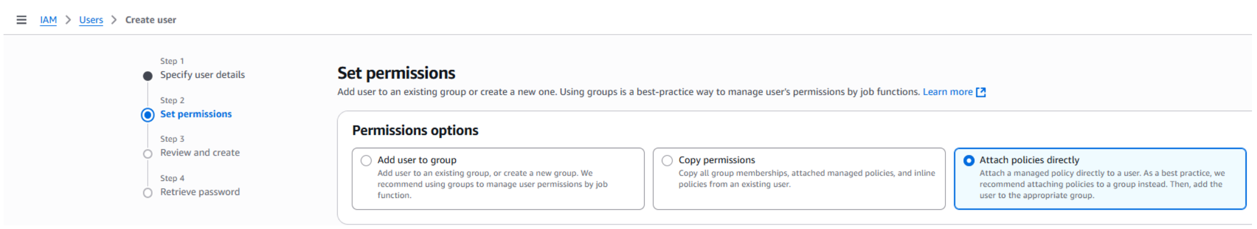
••••••••••••

[ ] Show password

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

Attaching policies Directly:



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

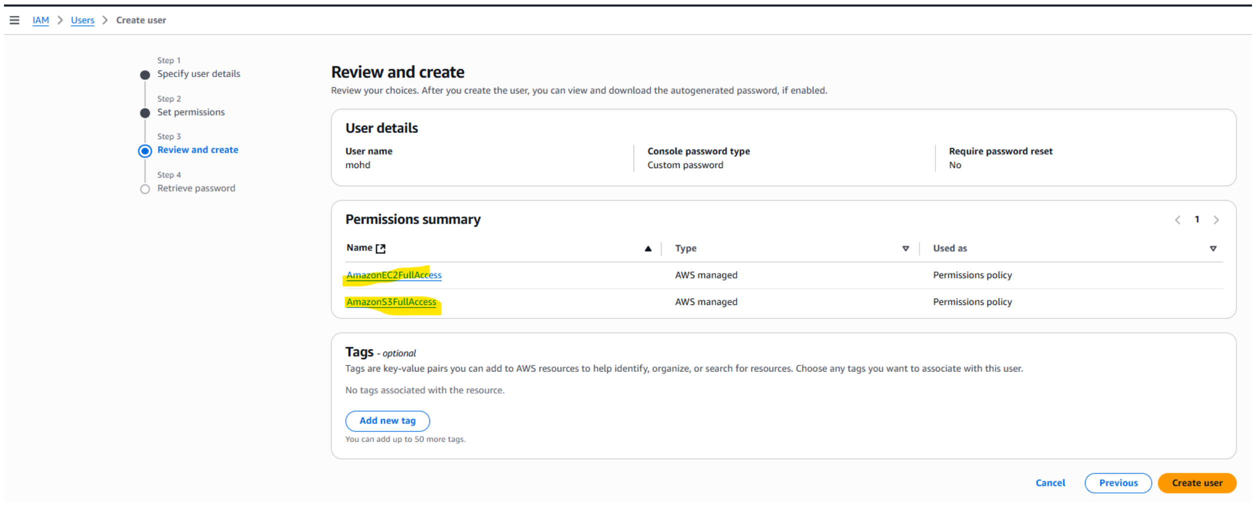
Permissions options

( ) 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.

Add ec2 and s3 full access policies:



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
mohd	Custom password	No

Permissions summary

Name	Type	Used as
AmazonEC2FullAccess	AWS managed	Permissions policy
AmazonS3FullAccess	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.

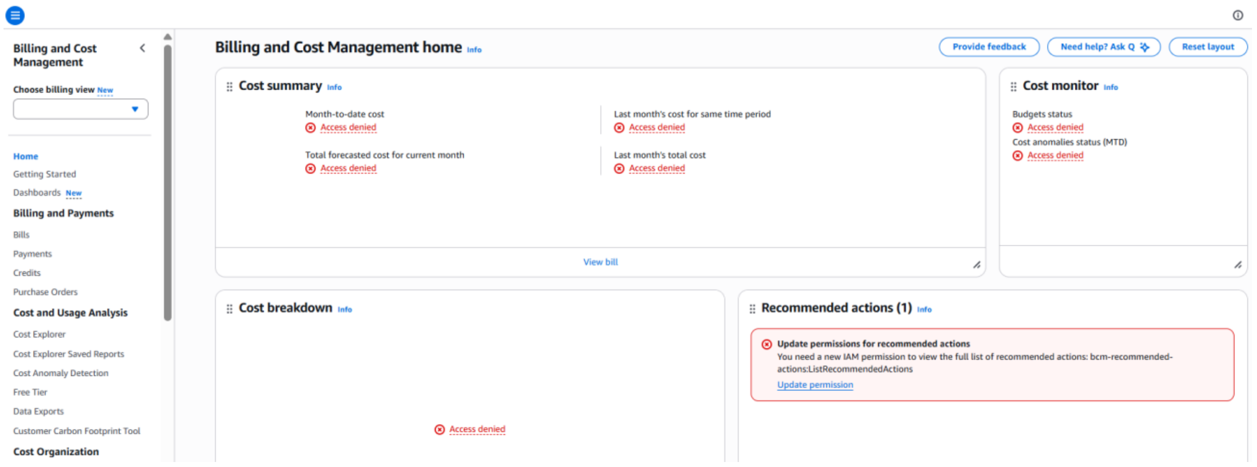
No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel Previous Create user

and login to the user 'mohd' and we can see there is only ec2 and s3 access we get and all other access are denied permission:


Billing and Cost Management

Choose billing view New

Home
Getting Started
Dashboards New
Billing and Payments
Bills
Payments
Credits
Purchase Orders
Cost and Usage Analysis
Cost Explorer
Cost Explorer Saved Reports
Cost Anomaly Detection
Free Tier
Data Exports
Customer Carbon Footprint Tool
Cost Organization

Billing and Cost Management home Info

Provide feedback | Need help? Ask Q | Reset layout

Cost summary Info

Month-to-date cost
Access denied

Last month's cost for same time period
Access denied

Total forecasted cost for current month
Access denied

Last month's total cost
Access denied

View bill

Cost monitor Info

Budgets status
Access denied
Cost anomalies status (MTD)
Access denied

Cost breakdown Info

Access denied

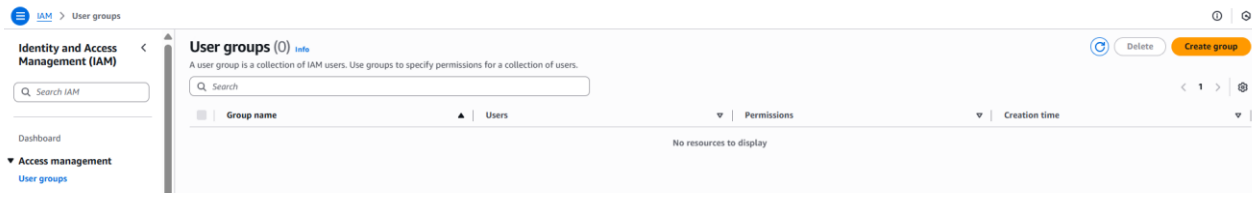
Recommended actions (1) Info

Update permissions for recommended actions
You need a new IAM permission to view the full list of recommended actions: bcm-recommended-actions:ListRecommendedActions
Update permission

-----------------done-------------

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

Goto IAM service, and click on user groups:


IAM > User groups

Identity and Access Management (IAM)

Search IAM

Dashboard
▼ Access management
User groups

User groups (0) Info

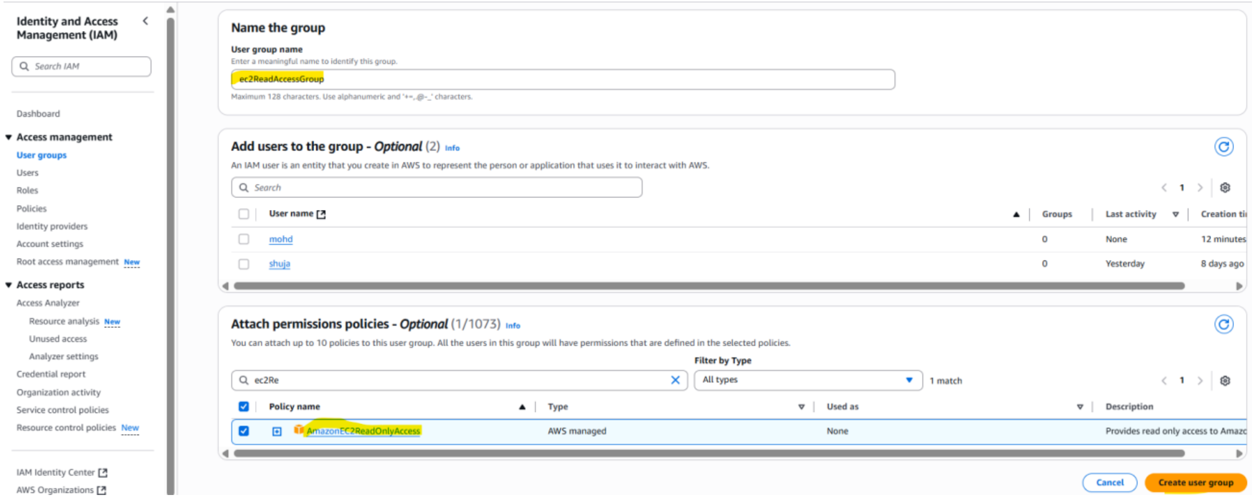
A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

Delete | Create group

Group name	Users	Permissions	Creation time

No resources to display

Give the group name and attach the permission policies:


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
Credential report
Organization activity
Service control policies
Resource control policies New

IAM Identity Center
AWS Organizations

Name the group

User group name
Enter a meaningful name to identify this group.
ec2ReadAccessGroup
Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

Add users to the group - *Optional* (2) 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 ti
☐	mohd	0	None	12 minutes
☐	shuja	0	Yesterday	8 days ago

Attach permissions policies - *Optional* (1/1073) 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.

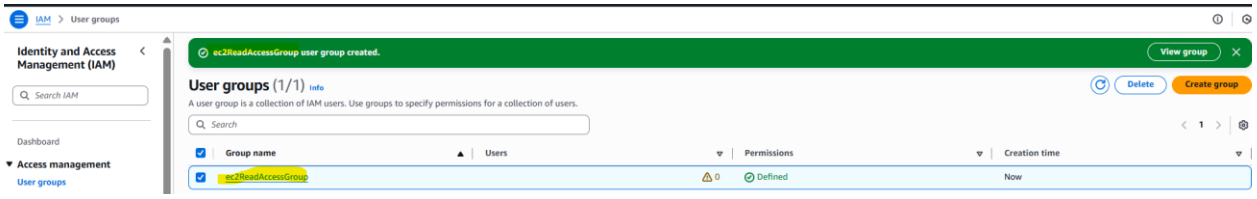
Filter by Type

ec2Re | All types | 1 match

	Policy name	Type	Used as	Description
✓	AmazonEC2ReadOnlyAccess	AWS managed	None	Provides read only access to Amazc

Cancel | Create user group

Group in IAM created with read access for EC2:



IAM > User groups

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

ec2ReadAccessGroup user group created.

View group

User groups (1/1) Info

Delete | Create group

A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

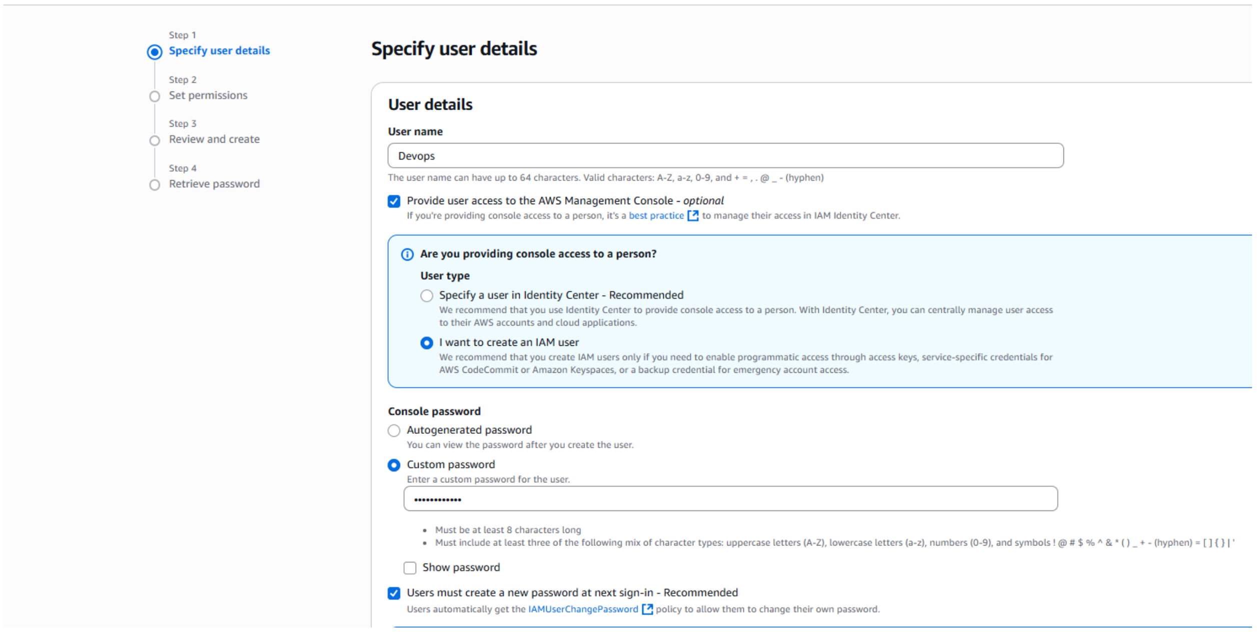
Search

	Group name	Users	Permissions	Creation time
✓	ec2ReadAccessGroup	0	Defined	Now

-----------------done------------

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

Create user 'Devops': goto IAM service and click on users



Step 1 Specify user details

Step 2 Set permissions

Step 3 Review and create

Step 4 Retrieve password

Specify user details

User details

User name

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.

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.

••••••••••••

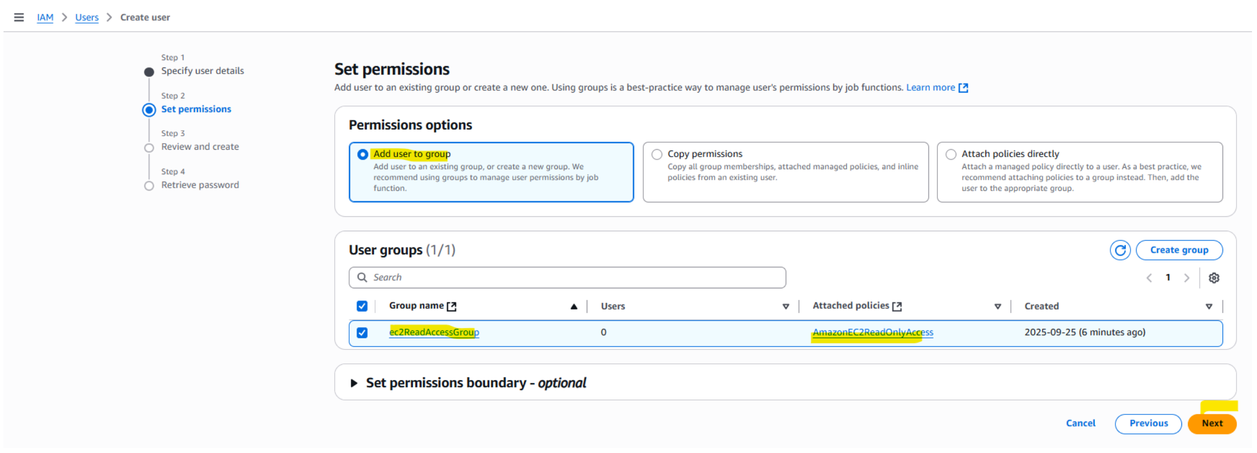
- Must be at least 8 characters long
- Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # $ % ^ & * ( ) _ + - (hyphen) = [ ] { } | '

[ ] Show password

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

Users automatically get the IAMUserChangePassword policy to allow them to change their own password.

Add user to the previously created user group:



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.

User groups (1/1)

Create group

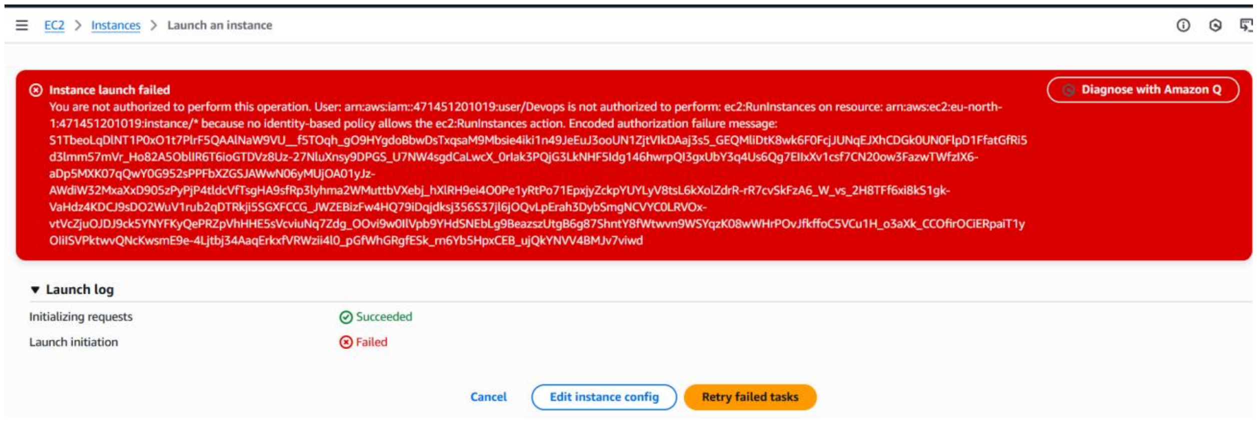
Search

	Group name	Users	Attached policies	Created
✓	ec2ReadAccessGroup	0	AmazonEC2ReadOnlyAccess	2025-09-25 (6 minutes ago)

Set permissions boundary - *optional*

Cancel | Previous | Next

Login to Devops user account and try to create the ec2 then you will be denied because ec2ReadonlyAccess permission is given:



EC2 > Instances > Launch an instance

Instance launch failed

You are not authorized to perform this operation. User: arn:aws:iam::471451201019:user/Devops is not authorized to perform: ec2:RunInstances on resource: arn:aws:ec2:eu-north-1:471451201019:instance/* because no identity-based policy allows the ec2:RunInstances action. Encoded authorization failure message:
S1TbeoLqDlNT1P0xO1t7PlrF5QAAlNaW9VU__f5TOqh_gO9HYgdoBbwDsTxqsaM9Mbsie4iki1n49JeEuJ3ooUN1ZjtVlkDAaj3s5_GEQMliDtK8wk6F0FcjJUNqEJXhCDGk0UN0FlpD1FfatGfRi5
d3lmm57mVr_Ho82A5OblIR6T6ioGTDVz8Uz-27NluXnsy9DPGS_U7NW4sgdCaLwcX_0rlak3PQjG3LkNHF5Idg146hwrpQI3gxUbY3q4Us6Qg7EIlxXv1csf7CN20ow3FazwTWfzIX6-
aDp5MXK07qQwY0G952sPPFbXZGSJAWwN06yMUjOA01yJz-
AWdiW32MxaXxD905zPyPjP4tldcVfTsgHA9sfRp3lyhma2WMuttbVXebj_hXlRH9ei4O0Pe1yRtPo71EpxjyZckpYUYLyV8tsL6kXolZdrR-rR7cvSkFzA6_W_vs_2H8TFf6xi8kS1gk-
VaHdz4KDCJ9sDO2WuV1rub2qDTRkji5SGXFCCG_JWZEBizFw4HQ79iDqjdksj356S37jl6jOQvLpErah3DybSmgNCVYC0LRVOx-
vtVcZjuOJDJ9ck5YNYFKyQePRZpVhHHE5sVcviuNq7Zdg_OOvi9w0IlVpb9YHdSNEbLg9BeazszUtgB6g87ShntY8fWtwvn9WSYqzK08wWHrPOvJfkffoC5VCu1H_o3aXk_CCOfirOCiERpaiT1y
OlilSVPktwvQNcKwsmE9e-4Ljtbj34AaqErkxfVRWzii4l0_pGfWhGRgfESk_rn6Yb5HpxCEB_ujQkYNVV4BMJv7viwd

▼ Launch log

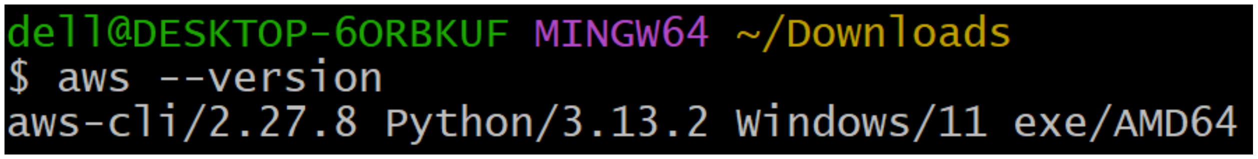
Initializing requests	✓ Succeeded
Launch initiation	✗ Failed

------------------done----------

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

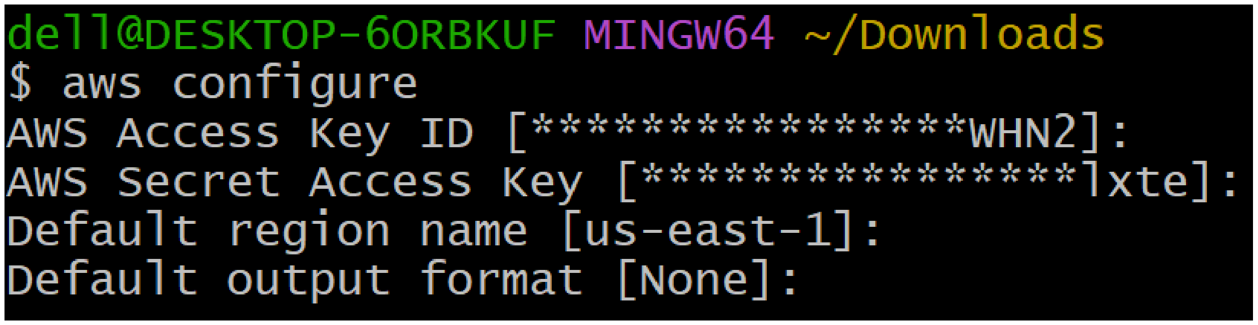
Install AWS CLI

Make sure AWS CLI is installed:



```
dell@DESKTOP-6ORBKUF MINGW64 ~/Downloads
$ aws --version
aws-cli/2.27.8 Python/3.13.2 Windows/11 exe/AMD64
```

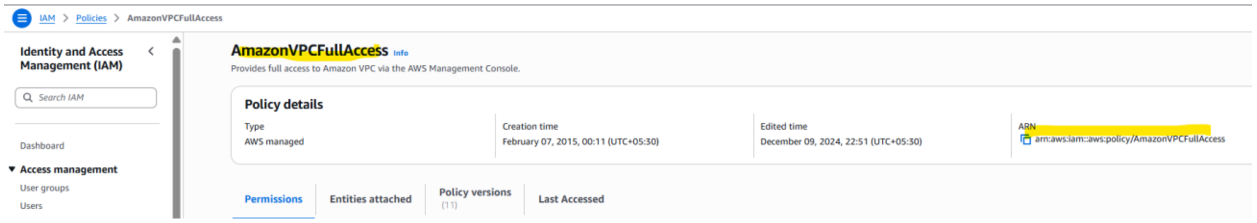
Configure aws credentials:



```
dell@DESKTOP-6ORBKUF MINGW64 ~/Downloads
$ aws configure
AWS Access Key ID [****************WHN2]:
AWS Secret Access Key [****************lxte]:
Default region name [us-east-1]:
Default output format [None]:
```

Note: I have already configured so did not entered else you have to enter the details.

Get the arn of AmazonVPCFullAccess:



AmazonVPCFullAccess

Provides full access to Amazon VPC via the AWS Management Console.

Policy details

Type: AWS managed

Creation time: February 07, 2015, 00:11 (UTC+05:30)

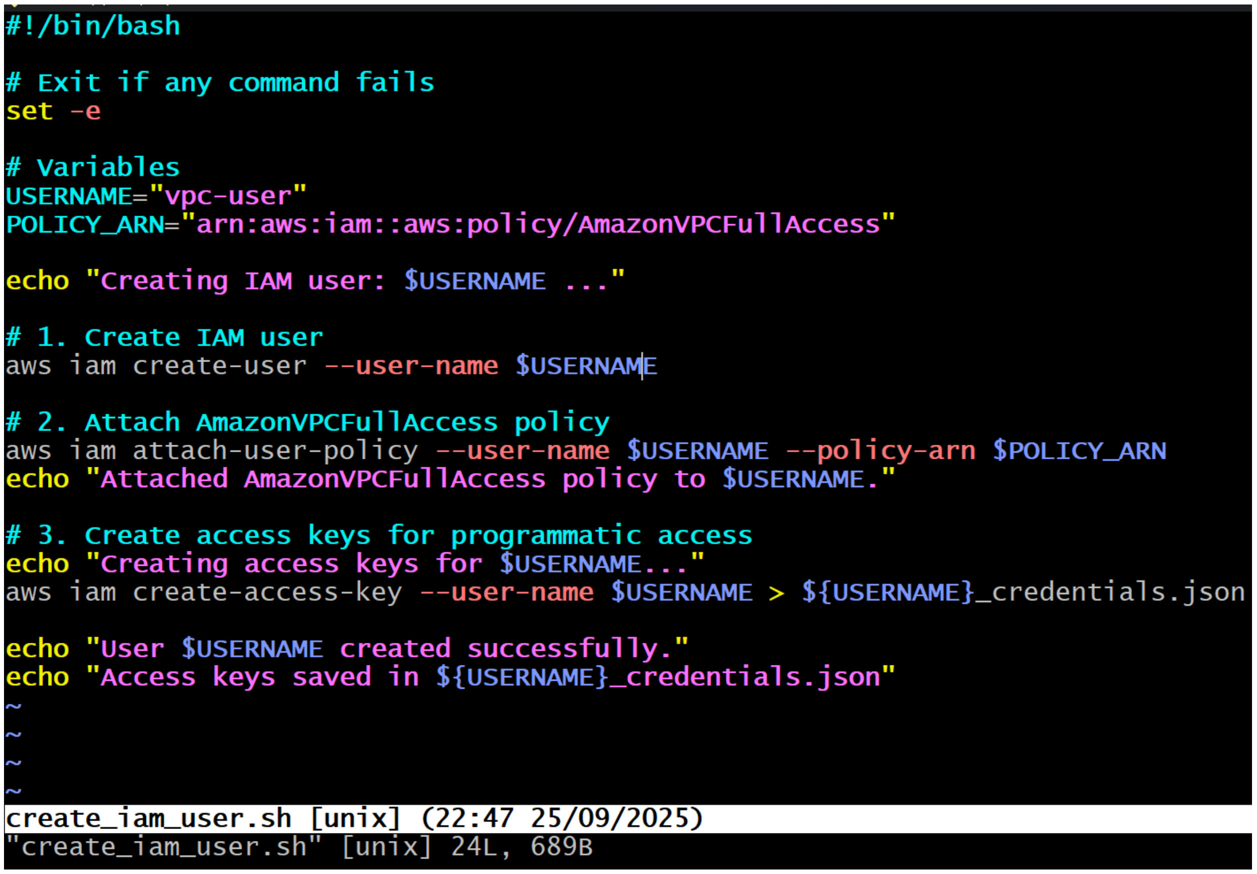
Edited time: December 09, 2024, 22:51 (UTC+05:30)

ARN: arn:aws:iam::aws:policy/AmazonVPCFullAccess

Write a script:

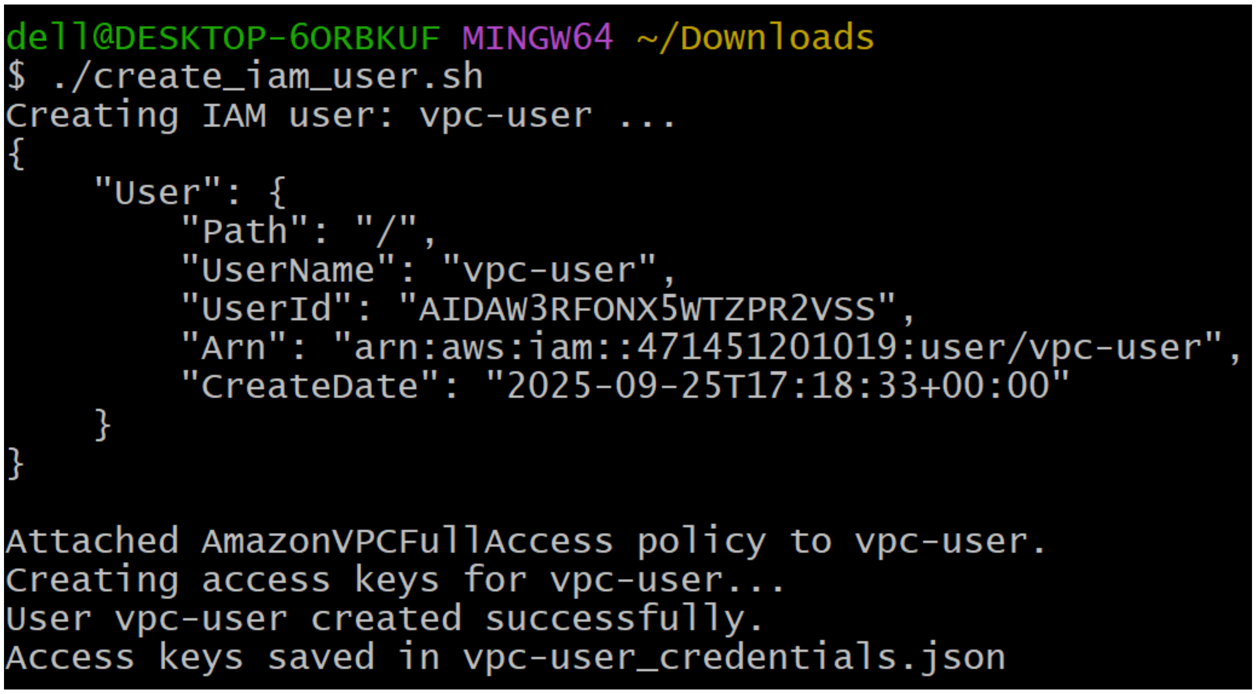
How It Works

1. **Creates IAM user** (aws iam create-user)
2. **Attaches managed policy** AmazonVPCFullAccess
3. **Generates access key + secret key** → stored in vpc-user_credentials.json



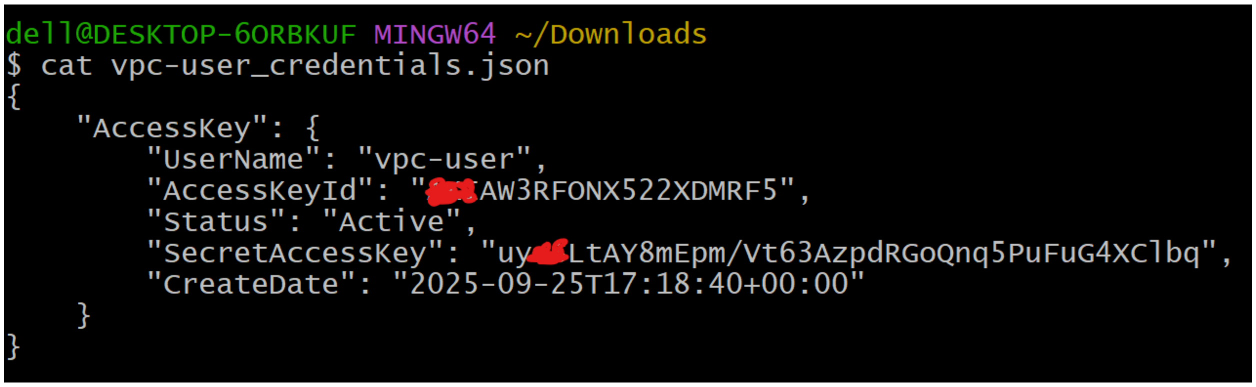
```
#!/bin/bash

# Exit if any command fails
set -e

# Variables
USERNAME="vpc-user"
POLICY_ARN="arn:aws:iam::aws:policy/AmazonVPCFullAccess"

echo "Creating IAM user: $USERNAME ..."

# 1. Create IAM user
aws iam create-user --user-name $USERNAME

# 2. Attach AmazonVPCFullAccess policy
aws iam attach-user-policy --user-name $USERNAME --policy-arn $POLICY_ARN
echo "Attached AmazonVPCFullAccess policy to $USERNAME."

# 3. Create access keys for programmatic access
echo "Creating access keys for $USERNAME..."
aws iam create-access-key --user-name $USERNAME > ${USERNAME}_credentials.json

echo "User $USERNAME created successfully."
echo "Access keys saved in ${USERNAME}_credentials.json"
~
~
~
~
create_iam_user.sh [unix] (22:47 25/09/2025)
"create_iam_user.sh" [unix] 24L, 689B
```

Execute the script:



```
dell@DESKTOP-6ORBKUF MINGW64 ~/Downloads
$ ./create_iam_user.sh
Creating IAM user: vpc-user ...
{
    "User": {
        "Path": "/",
        "UserName": "vpc-user",
        "UserId": "AIDAW3RFONX5WTZPR2VSS",
        "Arn": "arn:aws:iam::471451201019:user/vpc-user",
        "CreateDate": "2025-09-25T17:18:33+00:00"
    }
}

Attached AmazonVPCFullAccess policy to vpc-user.
Creating access keys for vpc-user...
User vpc-user created successfully.
Access keys saved in vpc-user_credentials.json
```

We get the output in vpc-user_credentials.json :



```
dell@DESKTOP-6ORBKUF MINGW64 ~/Downloads
$ cat vpc-user_credentials.json
{
    "AccessKey": {
        "UserName": "vpc-user",
        "AccessKeyId": "[redacted]AW3RFONX522XDMRF5",
        "Status": "Active",
        "SecretAccessKey": "uy[redacted]LtAY8mEpm/Vt63AzpdRGoQnq5PuFuG4XClbq",
        "CreateDate": "2025-09-25T17:18:40+00:00"
    }
}
```

---------------done-------------

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

you want to create an **IAM policy** that allows a user to access **EC2 only in certain regions** (for example, us-east-1 and ap-south-1) and **block all other regions**.

Login to AWS Console with an admin account.

Go to **IAM → Policies**.

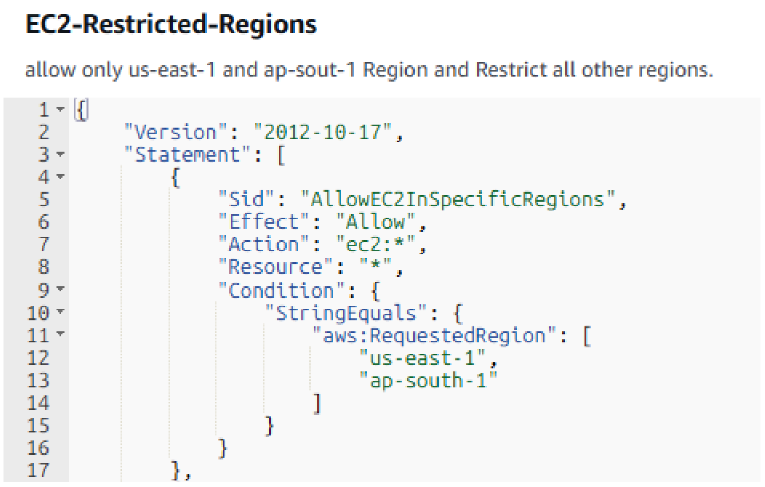
Click **Create policy**.

Choose **JSON** tab and paste the below policy (adjust regions as needed).

Give a **name** (e.g., EC2-Restricted-Regions) and description.

Click **Create policy**.

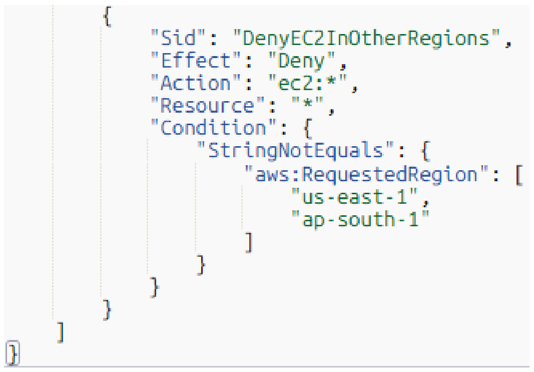
Now go to **IAM → Users** → Select your user → **Permissions** tab → **Add permissions** → **Attach policies directly** → Choose the policy you just created.



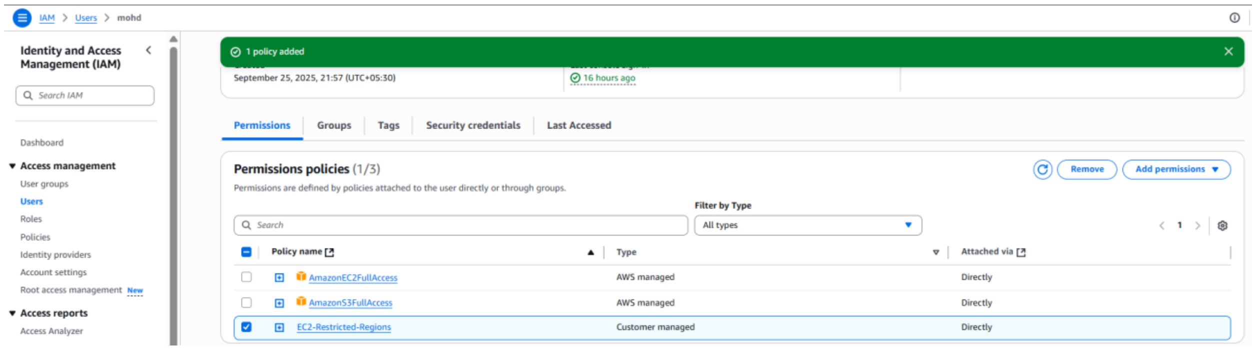
EC2-Restricted-Regions

allow only us-east-1 and ap-sout-1 Region and Restrict all other regions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowEC2InSpecificRegions",
            "Effect": "Allow",
            "Action": "ec2:*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestedRegion": [
                        "us-east-1",
                        "ap-south-1"
                    ]
                }
            }
        },
```



```
        {
            "Sid": "DenyEC2InOtherRegions",
            "Effect": "Deny",
            "Action": "ec2:*",
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:RequestedRegion": [
                        "us-east-1",
                        "ap-south-1"
                    ]
                }
            }
        }
    ]
}
```



IAM > Users > mohd

Identity and Access Management (IAM)

1 policy added

September 25, 2025, 21:57 (UTC+05:30) 16 hours ago

Permissions policies (1/3)

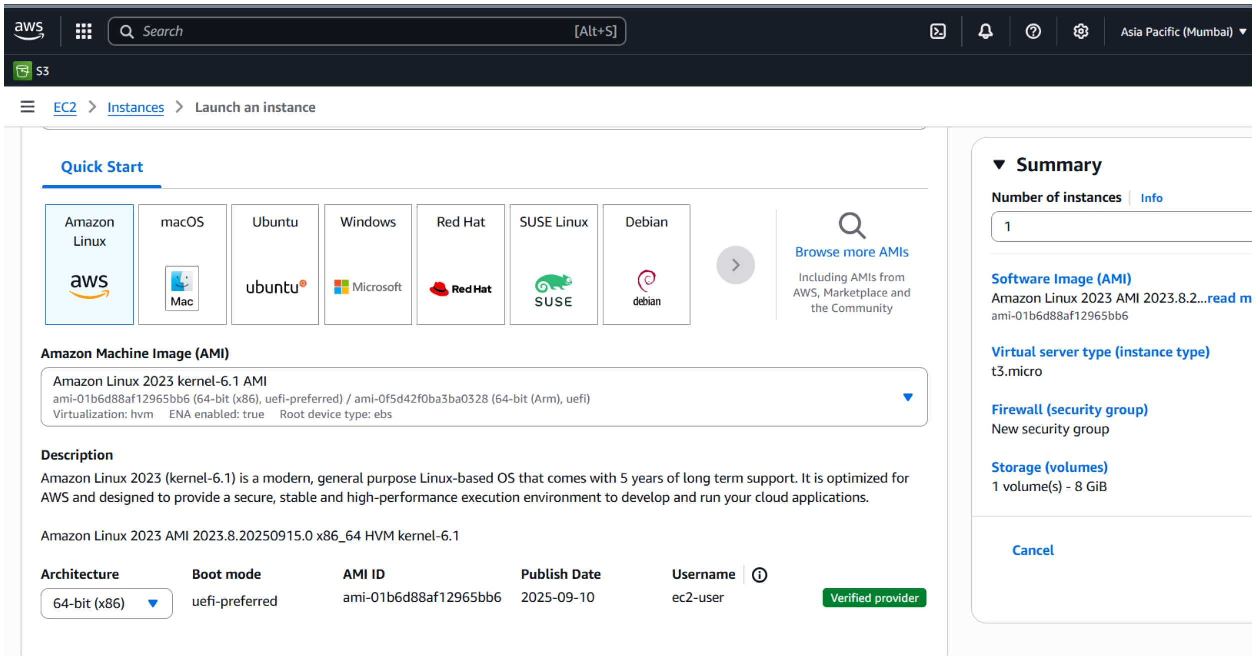
Permissions are defined by policies attached to the user directly or through groups.

Filter by Type: All types

	Policy name	Type	Attached via
☐	AmazonEC2FullAccess	AWS managed	Directly
☐	AmazonS3FullAccess	AWS managed	Directly
☑	EC2-Restricted-Regions	Customer managed	Directly

Then login to the user to which the policy has attached, in this example ‘mohd’ user:

We can see in ap-sout-1 Region we are allowed:



Search [Alt+S]

Asia Pacific (Mumbai)

S3

EC2 > Instances > Launch an instance

Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 kernel-6.1 AMI
ami-01b6d88af12965bb6 (64-bit (x86), uefi-preferred) / ami-0f5d42f0ba3ba0328 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.8.20250915.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	Publish Date	Username
64-bit (x86)	uefi-preferred	ami-01b6d88af12965bb6	2025-09-10	ec2-user

Verified provider

Summary

Number of instances Info
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.8.2...read m
ami-01b6d88af12965bb6

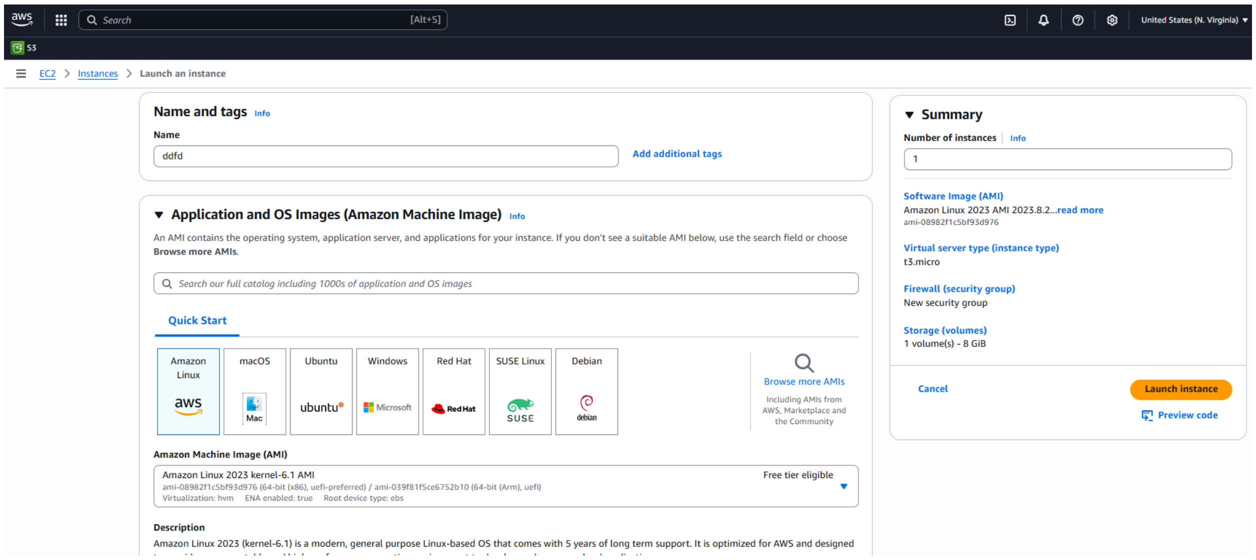
Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Cancel

We can see in us-east-1 Region we are allowed:



Search [Alt+S]

United States (N. Virginia)

S3

EC2 > Instances > Launch an instance

Name and tags Info

Name
ddfd

Add additional tags

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

Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 kernel-6.1 AMI — Free tier eligible
ami-08982f1c5bf93d976 (64-bit (x86), uefi-preferred) / ami-039f81f5ce6752b10 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed

Summary

Number of instances Info
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.8.2...read more
ami-08982f1c5bf93d976

Virtual server type (instance type)
t3.micro

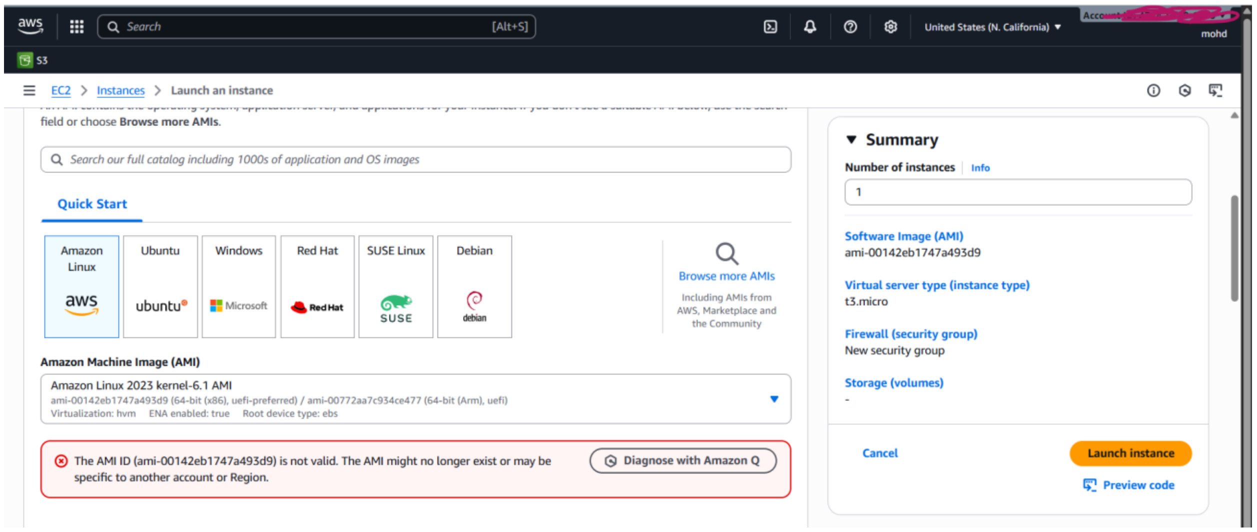
Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Cancel Launch instance

Preview code

We can see if we goto any other Region other than us-east-1 and ap-south-1 then it is not allowing:


Search [Alt+S]

United States (N. California) ▼ [redacted] mohd

S3

EC2 > Instances > Launch an instance

field or choose **Browse more AMIs**.

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

Quick Start

Amazon Linux | Ubuntu | Windows | Red Hat | SUSE Linux | Debian

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)
Amazon Linux 2023 kernel-6.1 AMI
ami-00142eb1747a493d9 (64-bit (x86), uefi-preferred) / ami-00772aa7c934ce477 (64-bit (Arm), uefi)
Virtualization: hvm   ENA enabled: true   Root device type: ebs

The AMI ID (ami-00142eb1747a493d9) is not valid. The AMI might no longer exist or may be specific to another account or Region.

Diagnose with Amazon Q

Summary
Number of instances | Info
1

Software Image (AMI)
ami-00142eb1747a493d9

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

Storage (volumes)
-

Cancel   Launch instance

Preview code

The **Allow** statement gives EC2 access only in the regions you specify.

The **Deny** statement explicitly denies EC2 access in all other regions.

Explicit **Deny** always overrides **Allow**, so the user is effectively restricted.

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

Sign in to Account B (admin) → **IAM → Roles → Create role**.
Choose **Another AWS account** and enter **Account A ID** as the trusted account. (If you want only a specific user to assume, you can edit the trust policy later to the user ARN.)

Continue to permissions: attach a policy that allows the S3 actions you want on project-data-bucket (example below).

Name the role AccountA-S3AccessRole (or whatever you prefer) and create it.

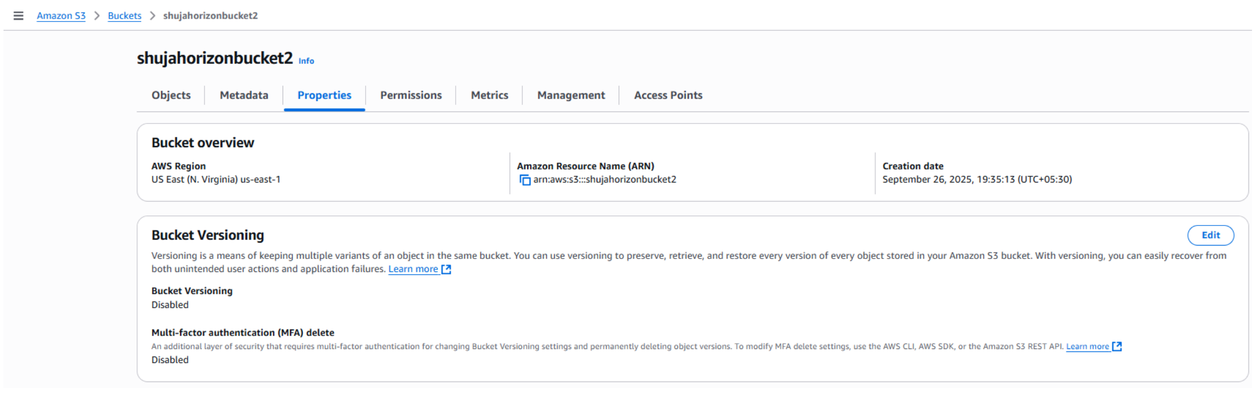
Requirement

Allow a user in **Account A** to access and manage an S3 bucket in **Account B**, including:

- Creating buckets (if needed)
- Listing buckets
- Uploading, downloading, and deleting objects

Step 1: Create the S3 Bucket in Account B

1. Log into **Account B**.
2. Go to **S3 → Create bucket**.
3. Set bucket name (e.g., shujahorizonbucket2).
4. Enable **Bucket owner enforced** for Object Ownership (recommended).
5. Enable **Default encryption** if needed.



Amazon S3 > Buckets > shujahorizonbucket2

shujahorizonbucket2 Info

Objects | Metadata | Properties | Permissions | Metrics | Management | Access Points

Bucket overview

AWS Region: US East (N. Virginia) us-east-1
Amazon Resource Name (ARN): arn:aws:s3:::shujahorizonbucket2
Creation date: September 26, 2025, 19:35:13 (UTC+05:30)

Bucket Versioning — Edit

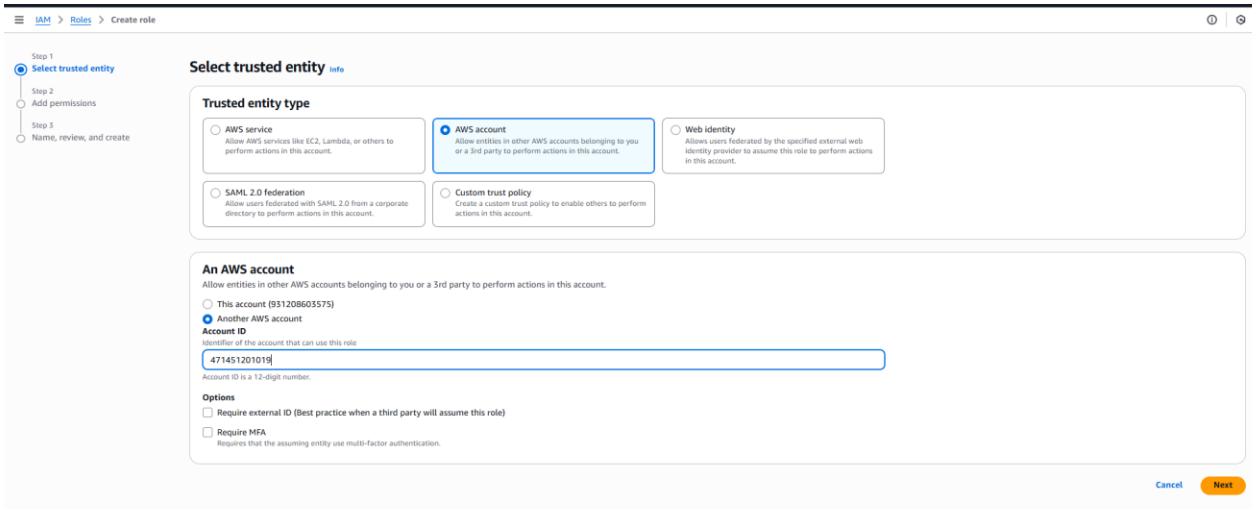
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning
Disabled

Multi-factor authentication (MFA) delete
An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. Learn more
Disabled

Step 2: Create IAM Role in Account B

1. Go to **IAM → Roles → Create Role**.
2. Select **Another AWS account** → enter **Account A ID**.
3. Attach policy (see below).



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.

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

This account (931208603575) [✓ Another AWS account]

Account ID
Identifier of the account that can use this role
471451201019
Account ID is a 12-digit number.

Options
- [ ] Require external ID (Best practice when a third party will assume this role)
- [ ] Require MFA — Requires that the assuming entity use multi-factor authentication.

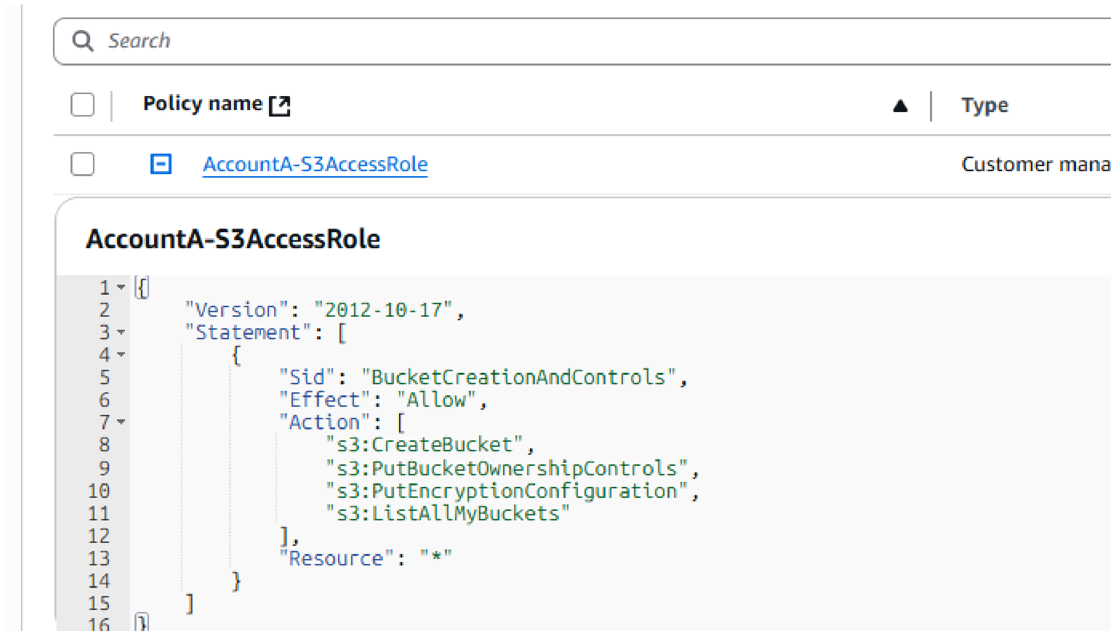
Cancel | Next

IAM Policy for the Role (Account B)

This **single policy** allows:

- Bucket creation
- Listing buckets
- Full access to objects in a specific bucket

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AccountLevelActions",
   "Effect": "Allow",
   "Action": [
    "s3:CreateBucket",
    "s3:PutBucketOwnershipControls",
    "s3:PutEncryptionConfiguration",
    "s3:ListAllMyBuckets"
   ],
   "Resource": "*"
  },
  {
   "Sid": "BucketLevelAccess",
   "Effect": "Allow",
   "Action": [
    "s3:ListBucket",
    "s3:GetBucketLocation"
   ],
   "Resource": "arn:aws:s3:::shujahorizonbucket2"
  },
  {
   "Sid": "ObjectLevelAccess",
   "Effect": "Allow",
   "Action": [
    "s3:GetObject",
    "s3:PutObject",
    "s3:DeleteObject",
    "s3:PutObjectAcl"
   ],
   "Resource": "arn:aws:s3:::shujahorizonbucket2/*"
  }
 ]
}
```

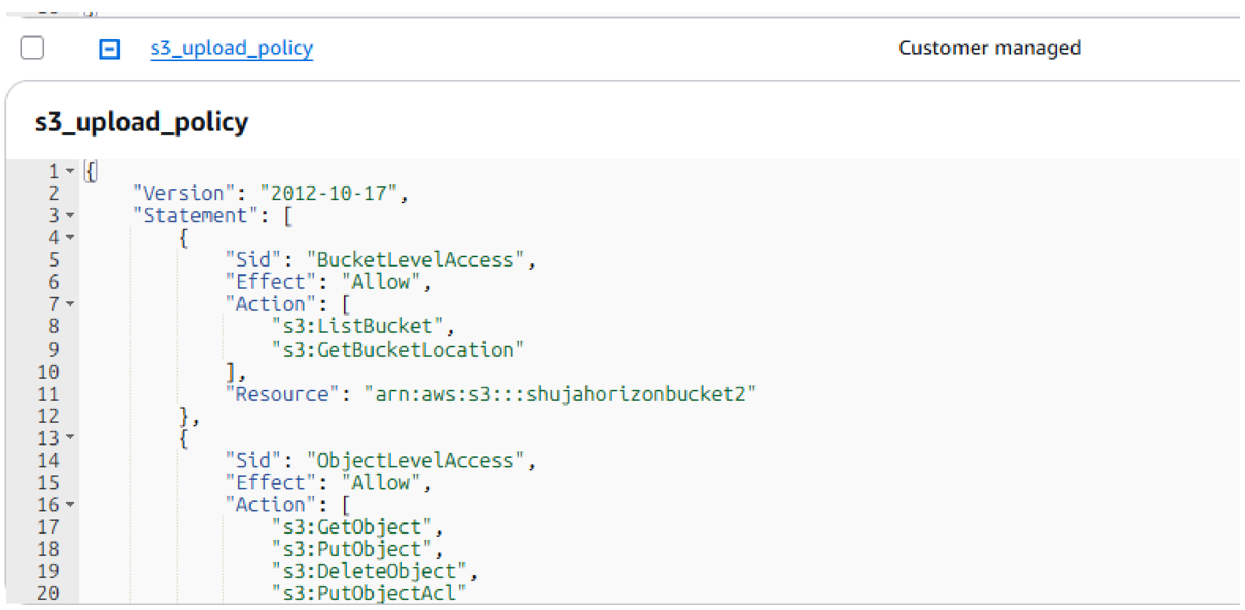


Search

	Policy name	Type
	AccountA-S3AccessRole	Customer mana

AccountA-S3AccessRole

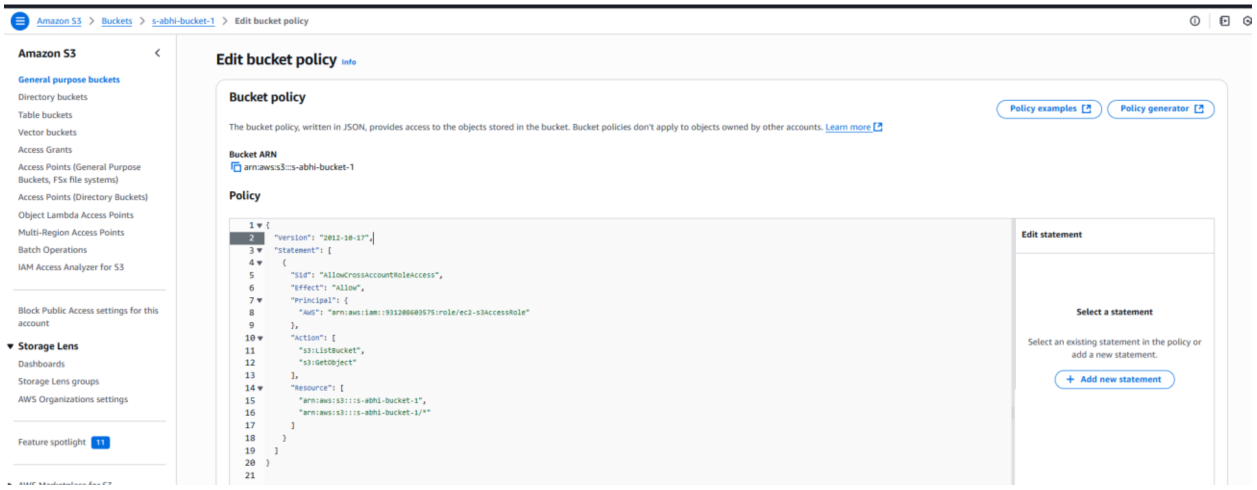
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "BucketCreationAndControls",
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:PutBucketOwnershipControls",
                "s3:PutEncryptionConfiguration",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        }
    ]
}
```



s3_upload_policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "BucketLevelAccess",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::shujahorizonbucket2"
        },
        {
            "Sid": "ObjectLevelAccess",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject",
                "s3:PutObjectAcl"
```

And also add the bucket policy in account B for s3 bucket:


Amazon S3 > Buckets > s-abhi-bucket-1 > Edit bucket policy
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Vector buckets
Access Grants
Access Points (General Purpose Buckets, FSx file systems)
Access Points (Directory Buckets)
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 11
Edit bucket policy Info
Policy examples
Policy generator
Edit statement
Select a statement
Select an existing statement in the policy or add a new statement.
Add new statement

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more

Bucket ARN

arn:aws:s3:::s-abhi-bucket-1

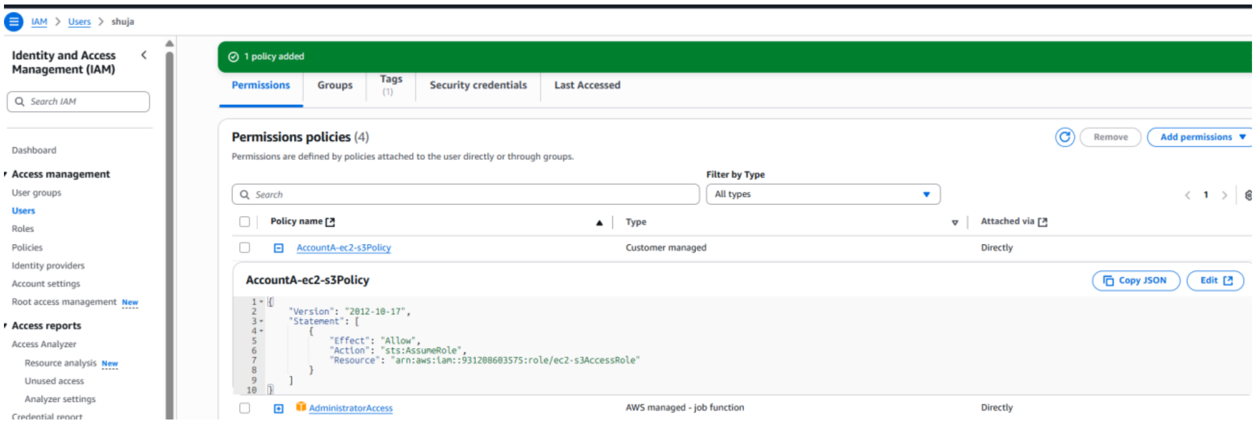
Policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCrossAccountRoleAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::931208603575:role/ec2-s3AccessRole"
      },
      "Action": [
        "s3:ListBucket",
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::s-abhi-bucket-1",
        "arn:aws:s3:::s-abhi-bucket-1/*"
      ]
    }
  ]
}
```

Step 3: Allow User in Account A to Assume the Role

1. Log into **Account A → IAM → Users → select user**.
2. Attach **inline policy** to allow assuming the role in Account B:

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::<AccountB-ID>:role/ec2-s3AccessRole"
    }
  ]
}
```



Permissions policies (4)

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

Policy name	Type	Attached via
AccountA-ec2-s3Policy	Customer managed	Directly
AdministratorAccess	AWS managed - job function	Directly

AccountA-ec2-s3Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::931208603575:role/ec2-s3AccessRole"
        }
    ]
}
```

Step 4: Access the Bucket from Account A

Option 1: CLI with temporary credentials

```
aws sts assume-role \
  --role-arn arn:aws:iam::<AccountB-ID>:role/ec2-s3AccessRole \
  --role-session-name ShujaSession
```

- Export credentials as environment variables:

```
export AWS_ACCESS_KEY_ID=<AccessKeyId>
export AWS_SECRET_ACCESS_KEY=<SecretAccessKey>
export AWS_SESSION_TOKEN=<SessionToken>
```

- Test listing objects:

```
aws s3 ls s3://shujahorizonbucket2
```

- Upload an object:

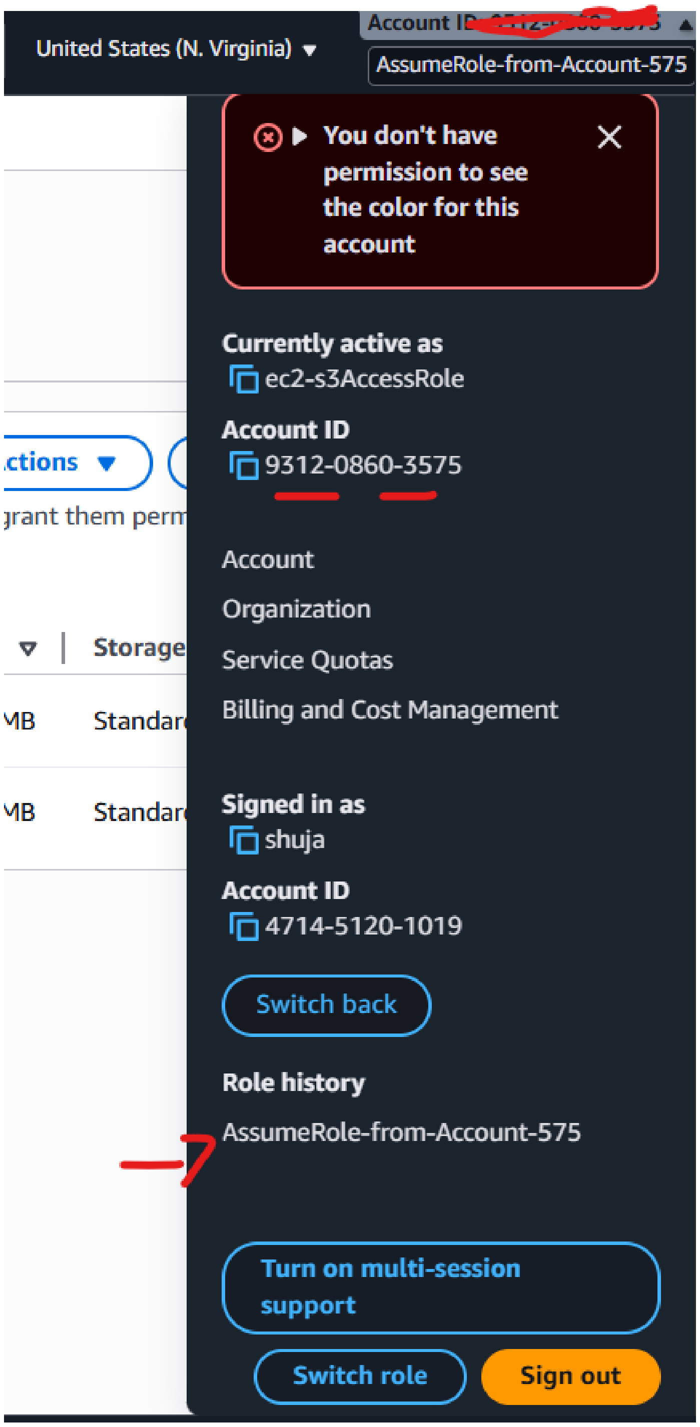
```
aws s3 cp test.txt s3://shujahorizonbucket2/
```

Option 2: Console

Switch role in AWS console → enter **Account B role**

Open **S3 → shujahorizonbucket2**

Upload / download objects


United States (N. Virginia) ▾

Account ID [redacted] ▴

AssumeRole-from-Account-575

ⓧ ▶ You don't have permission to see the color for this account ✕

Currently active as
ec2-s3AccessRole

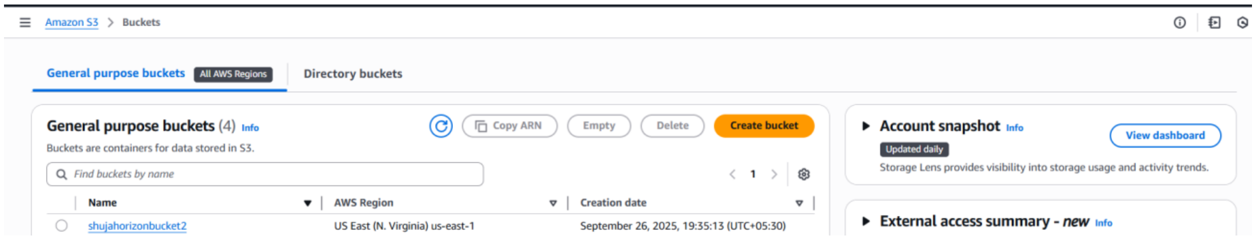
Account ID
9312-0860-3575

Account
Organization
Service Quotas
Billing and Cost Management

Signed in as
shuja

Account ID
4714-5120-1019

Role history
AssumeRole-from-Account-575

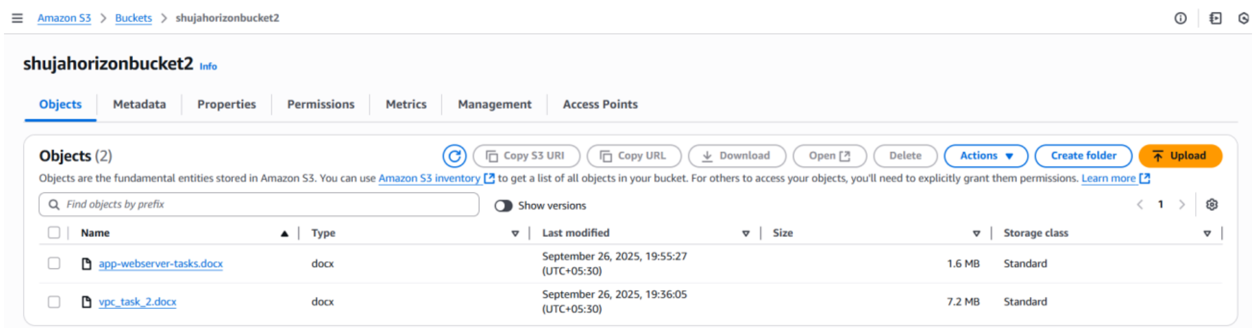


General purpose buckets (4) Info

Buckets are containers for data stored in S3.

Storage Lens provides visibility into storage usage and activity trends.

Name	AWS Region	Creation date
shujahorizonbucket2	US East (N. Virginia) us-east-1	September 26, 2025, 19:35:13 (UTC+05:30)



shujahorizonbucket2 Info

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
app-webserver-tasks.docx	docx	September 26, 2025, 19:55:27 (UTC+05:30)	1.6 MB	Standard
vpc_task_2.docx	docx	September 26, 2025, 19:36:05 (UTC+05:30)	7.2 MB	Standard

-------------completed---------