

# IAM Tasks-01

- **Objective:** To implement and manage AWS IAM users, groups, and policies by applying least-privilege access, automating user creation, restricting permissions by region, and enabling secure cross-account S3 access between AWS accounts.
- **Create one IAM user and assign EC2 and S3 full access roles.**

## 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 Tester	Console password type Custom password	Require password reset No
---------------------	--	------------------------------

**Permissions summary**

Name ↗	Type	Used as
<a href="#">AmazonEC2FullAccess</a>	AWS managed	Permissions policy
<a href="#">AmazonS3FullAccess</a>	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.

The user “Tester” has been created with both permissions.

**Tester** [Info](#) [Delete](#)

**Summary**

ARN <a href="#">arn:aws:iam::526018540742:user/Tester</a>	Console access Enabled without MFA	Access key 1 <a href="#">Create access key</a>
Created January 09, 2026, 14:54 (UTC+05:30)	Last console sign-in Never	

[Permissions](#) [Groups](#) [Tags](#) [Security credentials](#) [Last Accessed](#)

**Permissions policies (2)**

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

Filter by Type		
Policy name ↗	Type	Attached via ↗
<input type="checkbox"/> <a href="#">AmazonEC2FullAccess</a>	AWS managed	Directly
<input type="checkbox"/> <a href="#">AmazonS3FullAccess</a>	AWS managed	Directly

▶ **Permissions boundary** (not set)

# IAM Tasks-01

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

The screenshot shows the 'Testing' group details. It includes a summary section with the user group name 'Testing', creation time (January 09, 2026, 14:58 (UTC+05:30)), and ARN (arn:aws:iam::526018540742:group/Testing). Below this, there are tabs for 'Users', 'Permissions' (which is selected), and 'Access Advisor'. In the 'Permissions' tab, it shows 'Permissions policies (1)'. A table lists one policy: 'AmazonEC2ReadOnlyAccess' (AWS managed, Type: AWS managed). Buttons for 'Simulate' and 'Add permissions' are also visible.

A group named “Testing” has been created with the given permissions.

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

The screenshot shows the 'Testing' user details. It includes a summary section with the user group name 'Testing', creation time (January 09, 2026, 14:58 (UTC+05:30)), and ARN (arn:aws:iam::526018540742:group/Testing). Below this, there are tabs for 'Users (1)', 'Permissions', and 'Access Advisor'. In the 'Users (1)' tab, it shows 'Users in this group (1)'. A table lists one user: 'DevOps' (User name, Groups: Testing, Last activity: None, Creation time: Now). Buttons for 'Remove' and 'Add users' are visible.

A user named “DevOps” was created and added to the group “Testing”.

# IAM Tasks-01

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

```
$ cat IAM.bash
#!/bin/bash

# Exit immediately if any command fails
set -e

# Prompt for IAM username
read -p "Enter IAM username: " user

# Validate input
if [ -z "$user" ]; then
    echo "Error: Username cannot be empty"
    exit 1
fi

policy="arn:aws:iam::aws:policy/AmazonVPCFullAccess"

# Create IAM user
echo "Creating IAM user: $user"
aws iam create-user --user-name "$user"

# Attach VPC Full Access policy
echo "Attaching VPC Full Access policy"
aws iam attach-user-policy \
    --user-name "$user" \
    --policy-arn "$policy"

echo "User '$user' created with VPC Full Access"
```

The script was written so that the user can input any username and create a user with VPC full access.

The arn of the existing VPC full access was used in the policy variable.

```
Viaaas@LAPTOP-VG025G9U MINGW64 /d/DevOps/Me/DevOps-Learning/Scripts (main)
$ bash IAM.bash
Enter IAM username: Script
Creating IAM user: Script
{
    "User": {
        "Path": "/",
        "UserName": "Script",
        "UserId": "AIDAXU6JF6TDDXVQVMQD6",
        "Arn": "arn:aws:iam::526018540742:user/Script",
        "CreateDate": "2026-01-09T09:48:22+00:00"
    }
}
Attaching VPC Full Access policy
User 'Script' created with VPC Full Access
```

The user was successfully created.

# IAM Tasks-01

The user will have to configure AWS CLI with root permissions first before executing the above script.

**Script Info**

**Summary**

ARN arn:aws:iam::526018540742:user/Script	Console access Disabled	Access key 1 <a href="#">Create access key</a>
Created January 09, 2026, 15:18 (UTC+05:30)	Last console sign-in -	

**Permissions** | **Groups** | **Tags** | **Security credentials** | **Last Accessed**

**Permissions policies (1)**

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

Filter by Type	
Q Search	All types
<input type="checkbox"/> Policy name <a href="#">AmazonVPCFullAccess</a>	▲   Type AWS managed
<input type="checkbox"/> <a href="#">AmazonVPCFullAccess</a>	▼   Attached via Directly

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

**EC2AccessSpecificRegions Info**

Allows EC2 actions only when the request is made in us-east-1 or us-west-2.

**Policy details**

Type Customer managed	Creation time January 09, 2026, 15:34 (UTC+05:30)	Edited time January 09, 2026, 15:34 (UTC+05:30)	ARN arn:aws:iam::526018540742:policy/EC2AccessSpecificRegions
--------------------------	--	--	--

**Permissions** | **Entities attached** | **Tags** | **Policy versions (1)** | **Last Accessed**

**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.

```
1+ [ {  
2     "Version": "2012-10-17",  
3     "Statement": [  
4         {  
5             "Sid": "AllowEC2SpecificRegions",  
6             "Effect": "Allow",  
7             "Action": "ec2:*",  
8             "Resource": "*",  
9             "Condition": {  
10                 "StringEquals": {  
11                     "aws:RequestedRegion": [  
12                         "us-east-1",  
13                         "us-west-2"  
14                     ]  
15                 }  
16             }  
17         }  
18     ]  
19 }
```

**Copy** | **Edit** | **Summary** | **JSON**

The policy above allows EC2 actions only when the request is made in “us-east-1” or “us-west-2”.

- The regions can be changed as per requirement.
- The policy was created in json format with the name “EC2AccessSpecificRegions”.
- We can attach this policy to a specific user or a group.

# IAM Tasks-01

- We have two accounts: Account A and Account B. Account A user should access an S3 bucket in Account B.
- First, Account A user needs S3 full access to access the bucket in Account B.
- Then Account B needs to create a bucket policy which gives access to Account A.

```
1▼ (
2  "Version": "2012-10-17",
3  "Statement": [
4    {
5      "Sid": "AllowAccountAUserAccess",
6      "Effect": "Allow",
7      "Principal": {
8        "AWS": "arn:aws:iam::526018540742:user/DevOps"
9      },
10     "Action": [
11       "s3>ListBucket",
12       "s3GetObject"
13     ],
14     "Resource": [
15       "arn:aws:s3:::iambucket701303",
16       "arn:aws:s3:::iambucket701303/*"
17     ]
18   }
19 ]
20 )
```

- Then Account A can use AWS CLI to access the bucket.

```
Viqaas@LAPTOP-VG025G9U MINGW64 ~/Desktop
$ aws sts get-caller-identity
{
  "UserId": "AIDAXU6JF6TDPEQY2VEMM",
  "Account": "526018540742",
  "Arn": "arn:aws:iam::526018540742:user/DevOps"
}

Viqaas@LAPTOP-VG025G9U MINGW64 ~/Desktop
$ aws s3 ls s3://iambucket701303
        PRE bucket folder/
```

- Successfully completed all assigned AWS IAM tasks, including user, group, policy management, automation, and secure cross-account S3 access.