1. Create one IAM user and assign ec2,s3 full access role.

**Create IAM User**

1. **Log into AWS Management Console.**
2. Go to **IAM (Identity and Access Management)**.
3. In the left pane, click **Users**.
4. Click **Add users**.
5. Enter a **User name** (e.g., ec2-s3-admin).
6. Select **Access key - Programmatic access** if you want CLI/SDK access, or **AWS Management Console access** for console access.
7. Click **Next: Permissions**.

**Assign Permissions**

1. Choose **Attach existing policies directly**.
2. In the search bar, type:
   * AmazonEC2FullAccess
   * AmazonS3FullAccess
3. Check both policies.
4. Click **Next: Tags**
5. Click **Next: Review**.
6. Review the details and click **Create user**.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Create one Group in IAM and Assign Read access for ec2.

**Open IAM Service**

* In the AWS search bar at the top, type IAM.
* Click on **IAM**.

**. Create IAM Group**

* In the left sidebar, click on **User groups**.
* Click **Create group**.
* **Enter group name**, e.g., EC2-ReadOnly-Group.
* Click **Next**.

**Attach EC2 Read-Only Policy**

* In the policy search box, type AmazonEC2ReadOnlyAccess.
* Check the box for **AmazonEC2ReadOnlyAccess**.
* Click **Next**.
* Review the group and click **Create group**.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Create a new user with name Devops and add to the group created in task2.

**Create IAM User**

* In the left menu, click on **Users**.
* Click **Add users**.
* **Enter user name**: Devops.
* **Select access type**:
  + Check **AWS Management Console access** if the user needs to log into the console and set a password.
  + Optionally, check **Programmatic access** if you want CLI/SDK access.
* Click **Next: Permissions**.

**Add User to Existing Group**

* Select **Add user to group**.
* Check the group **EC2-ReadOnly-Group** (created in Task 2).
* Click **Next: Tags** (optional).
* Click **Next: Review**.
* Review and click **Create user**.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

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

**Configure AWS CLI**

* Run: aws configure
* Enter:
  + **AWS Access Key ID**
  + **AWS Secret Access Key**
  + **Default region (e.g. us-east-1)**
  + **Output format (e.g. json)**
* This ensures CLI can connect to your AWS account and has permission.
* Run : aws sts get-caller-identity

A computer screen with white text

AI-generated content may be incorrect.

Open your terminal

Create a new file called createvpcuser.sh.

Run: vi createvpcuser.sh

Then write script

#!/bin/bash

# Set user name variable

USER\_NAME="vpc-full-access-user"

echo "Creating IAM user: $USER\_NAME"

# Create IAM user

aws iam create-user --user-name "$USER\_NAME"

# Attach AmazonVPCFullAccess policy to user

aws iam attach-user-policy --user-name "$USER\_NAME" --policy-arn arn:aws:iam::aws:policy/AmazonVPCFullAccess

echo "User '$USER\_NAME' created and AmazonVPCFullAccess policy attached."

Then run : chmod +x createvpcuser.ch

Run: ./createvpcuser.sh

Vpc full access will be created and attached

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

1. Create a IAM policy to access ec2 for a specific user in specific regions only.

**Open Policy Creation in IAM**

1. Go to **IAM → Policies → Create policy**.
2. Select **Visual editor** tab.

**Define Service and Actions**

1. In the **Service** field, search for and select EC2.
2. In **Actions**, select:
   * **All EC2 actions (ec2:\*)**  
     This allows all EC2 actions like launch, modify, stop, etc

**Add Resources (Optional for EC2)**

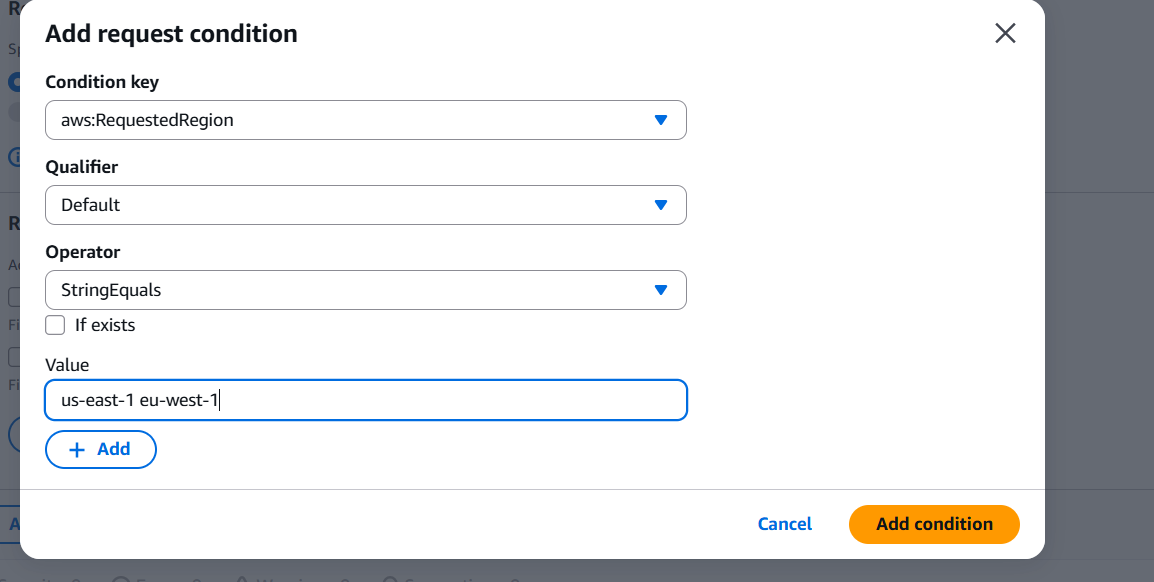
* You can keep it as **All resources** for EC2 services.
* For more restriction, you can specify ARNs of resources (optional).

**Add Region Restriction Using Condition**

1. Scroll down to **Request conditions**.
2. Click on **Add condition**.
3. Under **Key**, select aws:RequestedRegion.
4. **Operator**: StringEquals.
5. **Value**: Enter the regions you want to allow, e.g.:
   * us-east-1
   * eu-west-1

A screenshot of a computer

AI-generated content may be incorrect.



1. Click **Add condition**.

**Review & Create Policy**

1. Click **Next: Review**.
2. Enter:
   * **Policy name:** EC2restrictedregions
   * **Description:** Allows EC2 actions only in us-east-1 and eu-west-1.
3. Click **Create policy**.

**Attach the Policy to a User**

1. Go to **IAM → Users**.
2. Select the user (Devops).
3. Click **Add permissions** → **Attach existing policies directly**.
4. Search and select your newly created policy (EC2restrictedregions).
5. Click **Add permissions**.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. We have two accounts Account A and Account B, Account A user should access s3 bucket in Account B. (Collaborate with team member and execute this.Mostly asked in every interview)

Cross-account access scenario (S3 Bucket access)  
Bucket Policy Method:  
ACCOUNT B-  
● Log in to the Account B AWS Management Console.  
● Go to the S3 service.  
● Click on the name of the target bucket (moranithinbucket).  
● Go to the Permissions tab.  
● Scroll down to the Bucket policy section and click Edit.  
● Create a Policy depending on what permission or to what   
extent Account B wants to grant Account A’s user access to   
their s3 Bucket.

● Click Save changes.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

ACCOUNT A-  
● Ensure your AWS CLI is configured in Account A with   
credentials for the specific user.  
● Open your terminal. You can now access the bucket in   
Account B using AWS CLI commands, specifying the bucket   
name and your Account A profile  
● aws s3 ls s3://moranithinbucket --profile s3bucket –

List Objects in Account B’s bucket.

A white background with black text

AI-generated content may be incorrect.

A computer screen with white text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

