

IAM TASKS

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

In the AWS Console, search for and click IAM in the search bar.

1. In the left sidebar, click Users.
2. Click the "Add users" button.

Add User Details

1. User name: Enter a name (e.g., **ec2-s3-user**).
2. Access type:
 - Check "Programmatic access" if the user needs API/CLI access.
 - Check "AWS Management Console access" if the user will use the web console.

Click Next: Permissions.

Set Permissions

1. Choose "Attach policies directly".
2. In the search bar, find and check these policies:
 - **AmazonEC2FullAccess**
 - **AmazonS3FullAccess**

Click Next: Tags.

Step 5: Add Tags (Optional)

You can add tags like:

- Key: **Purpose**, Value: EC2 and S3 access

Click Next: Review.

Step 6: Review and Create

1. Review the user information.
2. Click "Create user".

Step 7: Save Credentials

Once the user is created:

- You'll see the Access Key ID and Secret Access Key (if you enabled programmatic access).

IAM Dashboard

Security recommendations

- Add MFA for root user
- Deactivate or delete access keys for root user

IAM resources

Resources in this AWS Account

User groups | Users | Roles | Policies | Identity providers

AWS Account

Account ID: 664418957525

Account Alias: Create

Sign-in URL for IAM users in this account: https://664418957525.signin.aws.amazon.com/console

Quick Links

My security credentials

This screenshot shows the AWS IAM Dashboard. It includes sections for security recommendations (Add MFA for root user, Deactivate or delete access keys for root user), IAM resources (User groups, Users, Roles, Policies, Identity providers), and AWS account details (Account ID: 664418957525, Account Alias: Create, Sign-in URL: https://664418957525.signin.aws.amazon.com/console). There are also quick links for My security credentials.

Users

Users (0)

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
No resources to display					

This screenshot shows the AWS IAM Users page. It displays a table with columns for User name, Path, Groups, Last activity, MFA, and Password. A message indicates "No resources to display".

Create user

Specify user details

User details

User name: Lalitha

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

This screenshot shows the "Specify user details" step of the AWS IAM Create user wizard. It includes fields for User name (Lalitha), a note about valid characters, and a checkbox for providing console access to the AWS Management Console. A sidebar lists steps: Step 1 (selected), Step 2 (Set permissions), Step 3 (Review and create), Step 4 (Retrieve password).

AWS | Search [Alt+S] Global ▾ lalithaganta

IAM > Users > Create user

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.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keypairs, you can generate them after you create this IAM user. [Learn more](#)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

AWS | Search [Alt+S] Global ▾ lalithaganta

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.

Permissions policies (1347)

Choose one or more policies to attach to your new user. [Create policy](#)

aws | Search [Alt+S] | Global ▾ | lalithaganta ▾

IAM > Users > Create user

Step 1: Specify user details
Step 2: Set permissions (selected)
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.

Permissions policies (2/1347)

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

Create policy

aws | Search [Alt+S] | Global ▾ | lalithaganta ▾

IAM > Users > Create user

Permissions policies (2/1347)

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

Filter by Type

Policy name	Type	Attached entities
<input type="checkbox"/> AccessAnalyzerServiceR...	AWS managed	0
<input type="checkbox"/> AdministratorAccess	AWS managed - job function	0
<input type="checkbox"/> AdministratorAccess-Am...	AWS managed	0
<input type="checkbox"/> AdministratorAccess-AW...	AWS managed	0
<input type="checkbox"/> AIOpsAssistantPolicy	AWS managed	0
<input type="checkbox"/> AIOpsConsoleAdminPolicy	AWS managed	0

Screenshot of the AWS IAM 'Create user' wizard Step 3: Set permissions.

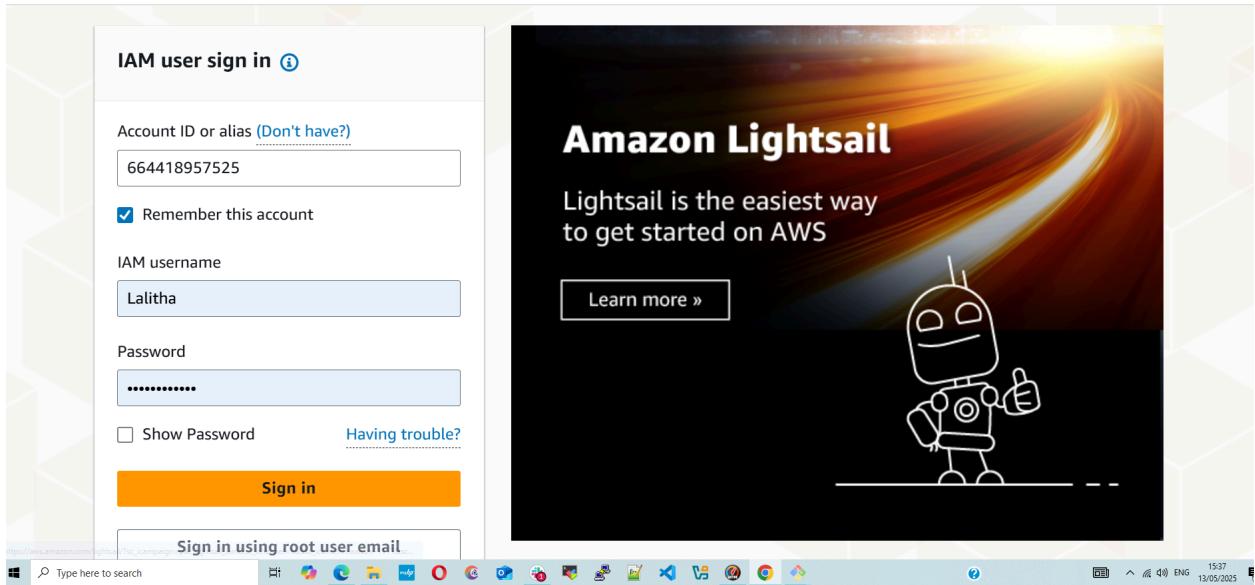
The page shows two attached policies: 'AmazonS3FullAccess' (AWS managed) and 'AmazonS3FullAccess' (Permissions boundary). A 'Tags - optional' section is present, showing a tag 'purpose' with value 'EC2 and S3 access'. Buttons for 'Cancel', 'Previous', and 'Create user' are at the bottom.

Screenshot of the AWS IAM 'Create user' wizard Step 4: Review and create.

A green success message states 'User created successfully'. It includes a 'View user' button and a note: 'You can view and download the user's password and email instructions for signing in to the AWS Management Console.' On the left, a sidebar shows 'Step 3: Review and create' and 'Step 4: Retrieve password' (selected). The right panel shows 'Console sign-in details' with a 'Console sign-in URL' (https://664418957525.signin.aws.amazon.com/console), 'User name' (Lalitha), and 'Console password' (redacted). Buttons for 'Email sign-in instructions', 'Cancel', 'Download .csv file', and 'Return to users list' are available.

Screenshot of the AWS IAM 'Users' page.

The left sidebar shows the 'Identity and Access Management (IAM)' navigation path: 'Dashboard' → 'Access management' → 'Users'. A search bar and a 'Create user' button are at the top right. The main table lists one user: 'Lalitha' (Info). The table columns include 'User name', 'Path', 'Group', 'Last activity', 'MFA', and 'Password'. Lalitha's status is '2 min'. Buttons for 'Delete' and 'Create user' are also visible.



This screenshot shows the AWS IAM 'Users' page. The sidebar indicates 'Identity and Access Management (IAM)'. The main area shows a table titled 'Users (0)' with a single row for 'User name' (Lalitha). A red box highlights an 'Access denied' message: 'You don't have permission to iam>ListUsers. To request access, copy the following text and send it to your AWS administrator.' It includes a link to 'Learn more about troubleshooting access denied errors.' Below this, a detailed error message is shown: 'User: arn:aws:iam::664418957525:user/Lalitha', 'Action: iam>ListUsers', 'On resource(s): arn:aws:iam::664418957525:user/', and 'Context: no permissions boundary allows the action'. A 'Diagnose with Amazon Q' button is also visible.

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

Open IAM Service

1. In the top search bar, type "IAM" and open the IAM service.
2. From the left panel, click on "User groups".
3. Click the "Create group" button.

Set Group Name

1. Enter a group name, such as: **EC2ReadOnlyGroup**.

Attach Permissions to the Group

In the permissions policy list, search for:

AmazonEC2ReadOnlyAccess

Click Next.

Step 5: Review and Create

1. Review the group name and attached policy.
2. Click "Create group".

Screenshot of the AWS IAM User groups page.

The left sidebar shows the navigation path: IAM > User groups.

The main content area displays the title "User groups (0) Info". A descriptive message states: "A user group is a collection of IAM users. Use groups to specify permissions for a collection of users." Below this is a search bar and a table header with columns: Group name, Users, Permissions, and Creation time.

The table body shows a message: "No resources to display".

Screenshot of the "Create user group" wizard.

The left sidebar shows the navigation path: IAM > User groups > Create user group.

The main content area has a title "Create user group". It starts with a "Name the group" step, where the "User group name" is set to "EC2ReadOnlyGroup".

Next is the "Add users to the group - Optional (1/1)" step, showing one user selected: "Lalitha".

Screenshot of the "Attach permissions policies - Optional (1/1045)" step.

The left sidebar shows the navigation path: IAM > User groups > Create user group.

The main content area shows the title "Attach permissions policies - Optional (1/1045) Info". A message says: "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." Below this is a "Filter by Type" section and a table.

The table has columns: Policy name, Type, Used as, and Description. One policy is listed: "AmazonEC2ReadOnlyAccess" (AWS managed, None, Provides read only access to Amazon).

At the bottom, there is a "Copy JSON" button.

The screenshot shows the AWS IAM User groups page. A green success message at the top states "EC2ReadOnlyGroup user group created." Below it, the "User groups (1)" section is displayed with a single entry: "EC2ReadOnlyGroup". The table columns are Group name, Users, Permissions, and Creation time. The "EC2ReadOnlyGroup" row shows "Defined" under Permissions and "Now" under Creation time.

The screenshot shows the AWS EC2 Instances page. A red box highlights an error message: "You are not authorized to perform this operation. User: arn:aws:iam::664418957525:user/Lalitha is not authorized to perform: ec2:DescribeInstances because no permissions boundary allows the ec2:DescribeInstances action". Below the error message are "Retry" and "Diagnose with Amazon Q" buttons. The "Select an instance" section is visible below the error message.

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

Step 1: Open AWS Console and Go to IAM

1. Sign in to the [AWS Management Console](#).

2. Search for and open IAM.

Step 2: Add New User

1. In the left sidebar, click "Users".

2. Click the "Add users" button.

Step 3: Enter User Details

• User name: **Devops**

• Access type (choose as needed) :

Programmatic access (for CLI/API)

AWS Management Console access (for web access)

Set a password if enabling console access.

Click Next: Groups.

Step 4: Add User to Group

From the list of groups, find and check the box for:

EC2ReadOnlyGroup

1. This will assign the permissions attached to that group (EC2 read-only).

Click Next: Tags.

Step 5: (Optional) Add Tags

Example:

- **Key: Department, Value: DevOps**

Click Next: Review.

Step 6: Review and Create

1. Verify:

- **Username: Devops**
- **Group: EC2ReadOnlyGroup**
- **Permissions: AmazonEC2ReadOnlyAccess (via group)**

2. Click Create user.

Screenshot of the AWS IAM User groups page.

User groups (1) Info

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

Group name	Users	Permissions	Creation time
EC2ReadOnlyGroup	1	Defined	7 minutes ago

Identity and Access Management (IAM)

- Dashboard
- Access management**
 - [User groups](#)
 - Users
 - Roles
 - Policies
 - Identity providers
 - Account settings
- Root access management [New](#)

Screenshot of the AWS IAM Users page.

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
Lalitha	/	1	-	-	27 min

Identity and Access Management (IAM)

- Dashboard
- Access management**
 - User groups
 - [Users](#)
 - Roles
 - Policies
 - Identity providers
 - Account settings
- Root access management [New](#)

Screenshot of the AWS IAM Create user page.

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

aws | Search [Alt+S] Global ▾ lalithaganta ▾

≡ 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)

Search

Create group

aws | Search [Alt+S] Global ▾ lalithaganta ▾

≡ IAM > Users > Create user

function.
Then, add the user to the appropriate group.

User groups (1/1)

Search

Group name	Users	Attached policies	Created
EC2ReadOnlyGroup	1	AmazonEC2ReadOnlyA...	2025-05-12 (11 ...)

▶ Set permissions boundary - optional

Cancel Previous Next

aws | Search [Alt+S] Global ▾ lalithaganta ▾

≡ 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 devops	Console password type Custom password	Require password reset Yes
---------------------	--	-------------------------------

Permissions summary

Name	Type	Used as
EC2ReadOnlyGroup	Group	Permissions group
IAMUserChangePassword	AWS managed	Permissions policy

aws | Search [Alt+S] Global | lalithaganta

≡ IAM > Users > Create user

EC2ReadOnlyGroup	Group	Permissions group
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.

Key Value - optional

Department Devops

You can add up to 49 more tags.

Cancel Previous Create user

aws | Search [Alt+S] Global | lalithaganta

≡ IAM > Users > Create user

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

Step 3 Review and create
Step 4 Retrieve password

Console sign-in details

Console sign-in URL <https://664418957525.signin.aws.amazon.com/console>

User name devops

Console password [Show](#)

Email sign-in instructions

Cancel Download .csv file Return to users list

aws | Search [Alt+S] Global | lalithaganta

≡ IAM > Users

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users**
- Roles
- Policies
- Identity providers
- Account settings
- Root access management [New](#)

Users (2) Info
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Search

User name	Path	Groups	Last activity	MFA	Password
devops	/	1	-	-	-
Lalitha	/	-	35 min	-	-

Users

User group memberships [Edit](#)

- EC2ReadOnlyGroup

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

AWS CLI is installed.

```
aws --version
```

AWS CLI is configured with admin credentials:

```
aws configure
```

You have permission to create IAM users and attach policies.

Create a new file: `nano create-vpc-user.sh`

Paste the following script:

```
bash
#!/bin/bash

# Variables
IAM_USER="vpc-full-access-user"
POLICY_ARN="arn:aws:iam::aws:policy/AmazonVPCFullAccess
"

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

echo "Attaching AmazonVPCFullAccess policy"
aws iam attach-user-policy --user-name "$IAM_USER"
--policy-arn "$POLICY_ARN"

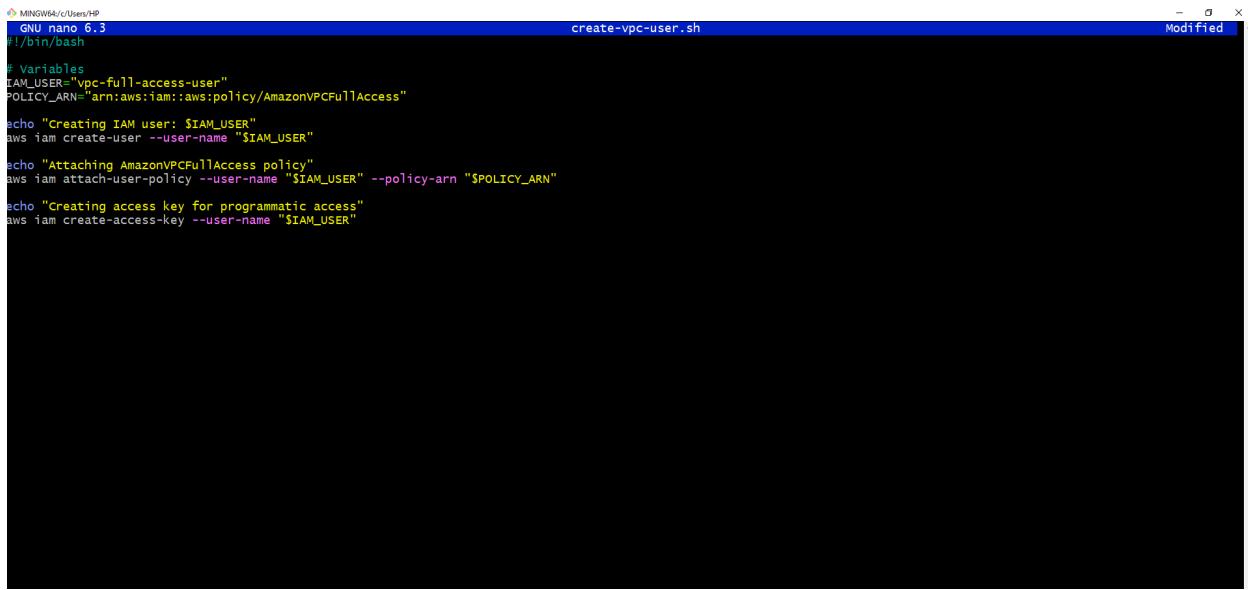
echo "Creating access key for programmatic access"
aws iam create-access-key --user-name "$IAM_USER"
```

Step 2: Make the Script Executable

chmod +x create-vpc-user.sh

Step 3: Run the Script

./create-vpc-user.sh



The screenshot shows a terminal window titled "create-vpc-user.sh" running on a Windows system (MINGW64). The terminal displays the contents of the "create-vpc-user.sh" script, which uses AWS CLI commands to create an IAM user and attach a policy. The script variables are defined at the top, followed by commands to create the user, attach a policy, and create access keys.

```
#!/bin/bash

# Variables
SIAM_USER="vpc-full-access-user"
POLICY_ARN="arn:aws:iam::aws:policy/AmazonVPCFullAccess"

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

echo "Attaching AmazonVPCFullAccess policy"
aws iam attach-user-policy --user-name "$SIAM_USER" --policy-arn "$POLICY_ARN"

echo "Creating access key for programmatic access"
aws iam create-access-key --user-name "$SIAM_USER"
```

```
MINGW64/c/Users/HP
$ aws --version
aws-cli/1.40.12 Python/3.12.6 Windows/10 botocore/1.38.13
$ aws configure
AWS Access Key ID [*****Q2NV]: AWS Secret Access Key [*****fItA]:
Default region name [eu-north-1]:
Default output format [json]:
$ nano create-vpc-user.sh
$ chmod +x create-vpc-user.sh
$ ./create-vpc-user.sh
Creating IAM user: vpc-full-access-user
{
    "User": {
        "Path": "/",
        "UserName": "vpc-full-access-user",
        "UserId": "AIDAZVMTUXDKZ6L075MOI",
        "Arn": "arn:aws:iam::664418957525:user/vpc-full-access-user",
        "CreateDate": "2025-05-12T09:46:00Z"
    }
}
Attaching AmazonVPCFullAccess policy
Creating access key for programmatic access
{
    "AccessKey": {
        "UserName": "vpc-full-access-user",
        "AccessKeyId": "AKIAZVMTUXDKS7JQA5LS",
        "Status": "Active",
        "SecretAccessKey": "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",
        "CreateDate": "2025-05-12T09:46:14Z"
    }
}

HP@LAPTOP-LPTR344H MINGW64 ~ (main)
$ |
```

```
HP@LAPTOP-LPTR344H MINGW64 ~ (main)
$ ./create-vpc-user.sh
Creating IAM user: vpc-full-access-user
{
    "User": {
        "Path": "/",
        "UserName": "vpc-full-access-user",
        "UserId": "AIDAZVMTUXDKZ6L075MOI",
        "Arn": "arn:aws:iam::664418957525:user/vpc-full-access-user",
        "CreateDate": "2025-05-12T09:46:00Z"
    }
}
Attaching AmazonVPCFullAccess policy
Creating access key for programmatic access
{
    "AccessKey": {
        "UserName": "vpc-full-access-user",
        "AccessKeyId": "AKIAZVMTUXDKS7JQA5LS",
        "Status": "Active",
        "SecretAccessKey": "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",
        "CreateDate": "2025-05-12T09:46:14Z"
    }
}

HP@LAPTOP-LPTR344H MINGW64 ~ (main)
$ |
```

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

Navigate to IAM > Policies

1. In the AWS Console, go to IAM

2. In the left panel, click Policies

3. Click "Create policy"

Step 3: Create Custom Policy JSON

1. Click the JSON tab

2. Paste the following policy:

```
json
CopyEdit
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EC2RegionRestrictedAccess",
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": [
            "us-east-1",
            "us-west-2"
          ]
        }
      }
    }
  ]
}
```

```
        }
    }
}
]
```

Name and Create the Policy

1. Click Next: Tags (optional)

2. Click Next: Review

3. Set:

- Name: **EC2SpecificRegionsPolicy**
- Description: Allows EC2 access only in
us-east-1 and **us-west-2**

4. Click Create policy

Attach Policy to a Specific User

1. In the IAM console, go to Users

2. Click on the user you want to assign the policy to
(e.g., **DevopsUser**)

3. Go to Permissions > Add permissions

4. Choose Attach policies directly

5. Search for and select EC2SpecificRegionsPolicy

6. Click Next > Review > Add permissions

The screenshot shows the AWS IAM Policies page. On the left, there's a sidebar with 'Identity and Access Management (IAM)' selected under 'Access management'. The main area displays a table titled 'Policies (1347)'. The table has columns for Policy name, Type, Used as, and Description. Some visible policy names include 'AccessAnalyzerServiceRole', 'AdministratorAccess', and 'AIOpsConsoleAdministratorPolicy'. A search bar and a filter by type dropdown are at the top of the table.

The screenshot shows the 'Create policy' wizard at Step 1: Specify permissions. The left sidebar shows 'Step 1: Specify permissions' and 'Step 2: Review and create'. The main area is titled 'Specify permissions' and contains a 'Policy editor' section. The policy JSON is displayed as:

```
1 ─ {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Sid": "EC2RegionRestrictedAccess",  
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 }
```

To the right of the editor, there's an 'Edit statement' button and a note: 'Select a statement Select an existing statement in the policy or add a new statement. + Add new statement'.

aws | Search [Alt+S] | Global ▾ | Lalithaganta ▾

≡ IAM > Policies > Create policy

```

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 }
20 |

```

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

aws | Search [Alt+S] | Global ▾ | Lalithaganta ▾

≡ IAM > Policies > Create policy

Step 1 Specify permissions
Step 2 Review and create

Review and create Info

Review the permissions, specify details, and tags.

Policy details

Policy name
Enter a meaningful name to identify this policy.

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

Description - optional
Add a short explanation for this policy.

Maximum 1,000 characters. Use alphanumeric and '+,-,@,_' characters.

aws | Search [Alt+S] | Global ▾ | Lalithaganta ▾

≡ IAM > Policies > Create policy

Service	Access level	Resource	Request conditions
EC2	Full access	All resources	aws:RequestedRegion eq us-east-1,us-west-2

Add tags - optional Info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

[Add new tag](#)
You can add up to 50 more tags.

[Cancel](#) [Previous](#) [Create policy](#)

The screenshot shows the AWS IAM Policies page. A green success banner at the top right says "Policy EC2SpecificRegionsPolicy created." with a "View policy" button. Below it, the main title is "Policies (1348) Info". A sub-header says "A policy is an object in AWS that defines permissions." There are buttons for "Actions", "Delete", and "Create policy". A search bar and a "Filter by Type" dropdown are also present. The main table lists three policies:

	Policy name	Type	Used as	Description
1	AccessAnalyzerServiceRole	AWS managed	None	-
2	AdministratorAccess	AWS managed - job ...	None	Provides full access to AWS services.
3	AdministratorAccess	AWS managed	None	Grants account administrative permis...

The screenshot shows the AWS IAM console. The left sidebar is titled "Identity and Access Management (IAM)" and includes sections for "Dashboard", "Access management" (with "User groups", "Users", "Roles", and "Policies" listed), "Identity providers", "Account settings", and "Root access management". The "Policies" section is currently selected. The main content area displays the "EC2SpecificRegionsPolicy" details. The policy name is "EC2SpecificRegionsPolicy" with an "Info" link. A description states "Allows EC2 access only in us-east-1 and us-west-2". There are "Edit" and "Delete" buttons. Below this, the "Policy details" section shows the "Type" as "Customer managed", "Creation time" as "May 12, 2025, 15:27 (UTC+05:30)", and "Edited time" as "May 12, 2025, 15:27 (UTC+05:30)". The ARN is listed as "arn:aws:iam::664418957525:policy/EC2SpecificRegion sPolicy". At the bottom, tabs for "Permissions", "Entities attached", "Tags", "Policy versions (1)", and "Last Accessed" are present, with "Permissions" being the active tab. The "Permissions defined in this policy" section has an "Edit" button, a "Summary" button, and a "JSON" link.

The screenshot shows the AWS IAM Users page. The left sidebar is titled "Identity and Access Management (IAM)" and includes sections for Dashboard, Access management (User groups, Users, Roles, Policies, Identity providers, Account settings, Root access management), and a "Search IAM" bar. The main content area is titled "Users (1/2) Info" and contains a table with two rows:

	User name	Path	Group	Last activity	MFA	Password
<input type="checkbox"/>	devops	/	1	-	-	22 min
<input checked="" type="checkbox"/>	Lalitha	/	1	-	-	56 min

At the top right of the main content area are buttons for "Delete" and "Create user". A search bar is also present at the top of the main content area.

Screenshot of the AWS IAM User Details page for 'Lalitha'.

Summary

ARN	arn:aws:iam::664418957525:user/Lalitha	Console access	Enabled without MFA
Created	May 12, 2025, 14:32 (UTC+05:30)	Last console sign-in	Never
Access key 1	Create access key		

Permissions (1) **Groups** (1) **Tags** (1) **Security credentials** **Last Accessed**

Permissions policies (3)

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

Screenshot of the AWS IAM User Permissions page for 'Lalitha'.

Permissions (1) **Security credentials** **Last Accessed**

Permissions policies (3)

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

Policy name	Type	Attached via
AmazonEC2FullAccess	AWS managed	Directly
AmazonEC2ReadOnlyAccess	AWS managed	Group EC2ReadOnlyGroup
AmazonS3FullAccess	AWS managed	Directly

Permissions boundary (set)

Screenshot of the AWS IAM Add permissions page for 'Lalitha'.

Add permissions

Step 2

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 (1346)

Filter by Type

Search	All types								
1	2	3	4	5	6	7	...	68	>

Screenshot of the AWS IAM 'Add permissions' step. The user 'Lalitha' is selected. Three options are shown:

- Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- Copy all group memberships, attached managed policies, inline policies, and any existing permissions boundaries from an existing user.
- 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.

The 'Permissions policies' section shows a search bar for 'EC2SPE', a filter for 'All types', and one result: 'EC2SpecificRegionsPolicy' (Customer managed). Buttons for 'Cancel' and 'Next' are at the bottom.

Screenshot of the AWS IAM 'Review' step. The user 'Lalitha' has been assigned the 'EC2SpecificRegionsPolicy'. The 'User details' section shows 'User name: Lalitha'. The 'Permissions summary' section shows the assigned policy. Buttons for 'Cancel', 'Previous', and 'Add permissions' are at the bottom.

Screenshot of the AWS IAM 'Identity and Access Management (IAM)' dashboard. A green notification bar says '1 policy added' for user 'Lalitha'. The 'Permissions' tab is selected in the 'Users' section, showing 'Permissions policies (4)'. A table lists the policies attached to Lalitha. Buttons for 'Remove' and 'Add permissions' are available.

Identity and Access Management (IAM)

Dashboard

Access management

- User groups
- Users**
- Roles
- Policies
- Identity providers
- Account settings
- Root access management [New](#)

Filter by type

Policy name	Type	Attached via
AmazonEC2FullAccess	AWS managed	Directly
AmazonEC2ReadOnlyAccess	AWS managed	Group EC2ReadOnlyGroup
AmazonS3FullAccess	AWS managed	Directly
EC2SpecificRegionsPolicy	Customer managed	Directly

▶ Permissions boundary (set)

▼ Generate policy based on CloudTrail events

6) 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)

STEP 1 : Create IAM user in account A And S3 bucket in account B

STEP 2 : The created user gives awsfullaccess to the S3 bucket and provide permission to the below policy

The below Inline policy :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3>ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::<bucketname>",
        "arn:aws:s3:::<bucketname>/*"
      ]
    }
  ]
}
```

```
]
}
]
}
```

STEP 4 : Create a S3 bucket in Account B and give the below policy to the S3 Bucket

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::750471539774:user/crossregion-IAM"
      },
      "Action": [
        "s3:GetObject",
        "s3>ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::task-s3bucket-29",
        "arn:aws:s3:::task-s3bucket-29/*"
      ]
    }
  }
}
```

```
 ]  
 }  
 }
```

Then save all the changes

Open git bash and run the below commands

```
aws configure --profile crossregion-IAM
```

Provide access key and security key

The screenshot shows the AWS IAM 'Create user' wizard at the 'Set permissions' step. The left sidebar lists three steps: 'Specify user details' (completed), 'Set permissions' (selected), and 'Review and create'. The main area is titled 'Set permissions' with the sub-section 'Permissions options'. It contains three choices: 'Add user to group' (disabled), 'Copy permissions' (disabled), and 'Attach policies directly' (selected). Below this is a section for 'Permissions policies' with a count of 1/1350. A search bar and a 'Create policy' button are present. At the bottom, there's a 'Filter by Type' dropdown set to 's3full', a 'Policy name' filter for 'AmazonS3FullAccess', and a note about setting a 'permissions boundary - optional'.

Screenshot of the AWS IAM 'Create user' wizard - 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
IAM-02	None	No

Permissions summary

Name	Type	Used as
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 add.

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS IAM 'Users' page.

Identity and Access Management (IAM)

Users (4) Info

User created successfully

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

Users (4)

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
Lalitha	/	1	Yesterday	-	2 days
ypc-full-access-user	/	0	-	-	-

View user Delete Create user

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws | Search [Alt+S] | Global ▾ | lalithaganta ▾

IAM > Users > IAM-02 > Edit policy

Step 1: Modify permissions in crossregion Step 2: Review and save

Modify permissions in crossregion Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

Visual | **JSON** | Actions ▾

```
1 ▼ {
  2   "Version": "2012-10-17",
  3   "Statement": [
  4     {
  5       "Effect": "Allow",
  6       "Action": [
  7         "s3:GetObject",
  8         "s3>ListBucket"
  9       ],
  10      "Resource": [
  11        "arn:aws:s3:::iamcnct123",
  12        "arn:aws:s3:::iamcnct123/*"
  13      ]
  14    }
  15  ]
```

Edit statement | Remove

Add actions

Choose a service Filter services

Included S3

Available AI Operations

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws | Search [Alt+S] | Global ▾ | lalithaganta ▾

IAM > Users > IAM-02 > Create policy

Review and create

Policy details

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

Permissions defined in this policy Info

Edit

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

Search

Show remaining 438 services

Allow (1 of 439 services)

Service	Access level	Resource	Request conditions

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

C2

Amazon S3 > Buckets > iamcnct123

Amazon S3

General purpose buckets

- Factory buckets
- Role buckets
- Access Grants
- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access Analyzer for S3

Check Public Access settings for this account

Storage Lens

- Dashboards
- Storage Lens groups

Successfully edited bucket policy.

```
{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "AWS": "arn:aws:iam::664418957525:user/IAM-02" }, "Action": [ "s3:GetObject", "s3>ListBucket" ], "Resource": [ "arn:aws:s3:::iamcnct123", "arn:aws:s3:::iamcnct123/*" ] } ] }
```

Copy

aws | Search [Alt+S]

IAM > Users > IAM-02 > Create access key

Access key created

This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Step 1
● Access key best practices & alternatives

Step 2 - optional
● Set description tag

Step 3
● Retrieve access keys

Retrieve access keys Info

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
	AKIAZVMTUXDK67FMG46T <small>*****</small>

Access key best practices

- Never store your access key in plain text, in a code repository, or in code

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

```
MINGW64/c/Users/HP/OneDrive/Desktop
$ aws configure --profile iam-02
AWS Access Key ID [None]: AKIAZVMUXDK67FMG46T
AWS Secret Access Key [None]: smJqlstrnozmlefjG5i0uc3+QRyjNEmksfercUJ7
Default region name [None]: us-east-1
Default output format [None]: json

$ aws s3 ls s3://iamcnct123 --profile iam-02
2025-05-15 11:42:21    41 s3check.txt

$ |
```