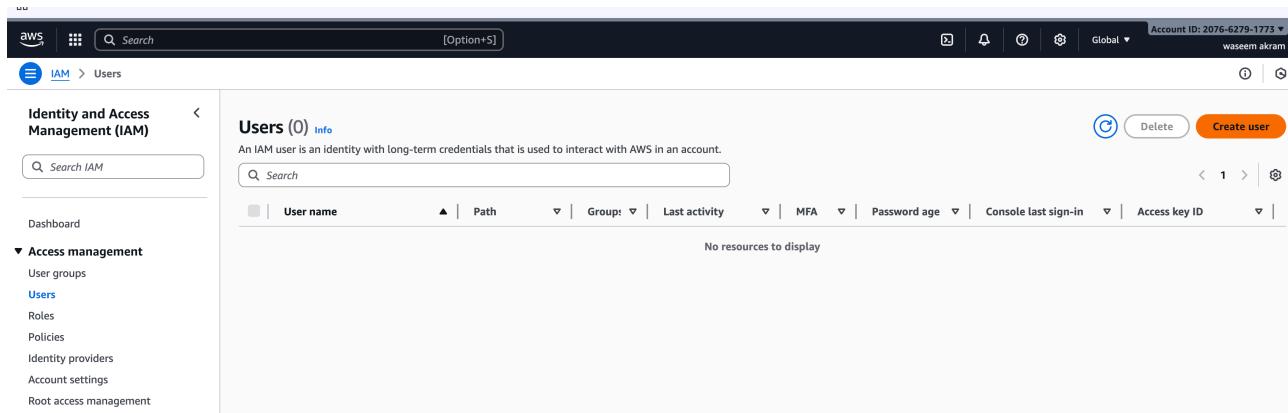


IAM TASK

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

—> go to IAM and —> users —> create user —> give user name as myuser —> click next

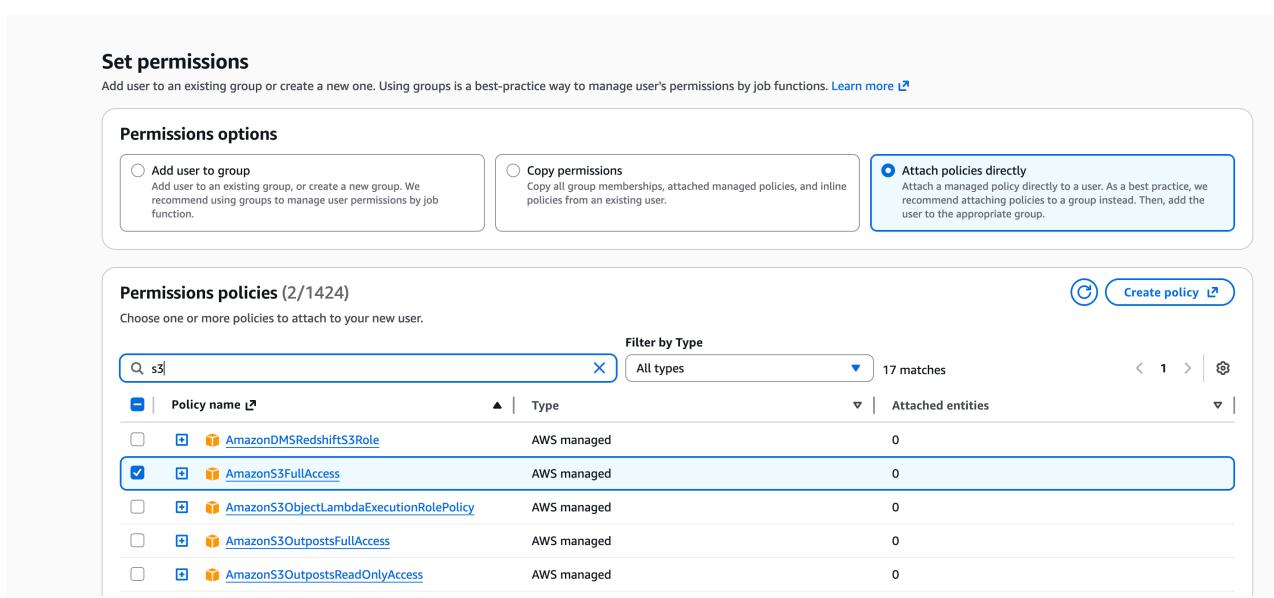


The screenshot shows the AWS IAM Users page. The left sidebar has sections for Identity and Access Management (IAM), Access management (User groups, Users, Roles, Policies, Identity providers, Account settings, Root access management), and a search bar. The main area is titled 'Users (0)' and contains a search bar and a table header with columns: User name, Path, Group:, Last activity, MFA, Password age, Console last sign-in, and Access key ID. Below the table, it says 'No resources to display'.

—> 2) Assign Permissions

On the "Permissions" page:

1. Select **Attach existing policies directly**
2. Search for:
 - o **AmazonEC2FullAccess**
 - o **AmazonS3FullAccess**
3. Tick both checkboxes.



The screenshot shows the 'Set permissions' page. It has a 'Permissions options' section with three choices: 'Add user to group', 'Copy permissions', and 'Attach policies directly' (which is selected). Below this is a 'Permissions policies (2/1424)' section with a search bar, a 'Create policy' button, and a table of policies. The table includes columns for Policy name, Type, and Attached entities. Policies listed include 'AmazonDMSRedshiftS3Role' (AWS managed, 0 attached), 'AmazonS3FullAccess' (AWS managed, 0 attached, checked), 'AmazonS3ObjectLambdaExecutionRolePolicy' (AWS managed, 0 attached), 'AmazonS3OutpostsFullAccess' (AWS managed, 0 attached), and 'AmazonS3OutpostsReadOnlyAccess' (AWS managed, 0 attached).

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 (2/1424)
Choose one or more policies to attach to your new user.

Filter by Type		
<input type="text" value="ec2"/>	All types	47 matches
<input type="checkbox"/> Policy name ↗	Type	Attached entities
<input type="checkbox"/> AmazonEC2ContainerRegistryFullAccess	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerRegistryPowerUser	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerRegistryPullOnly	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerRegistryReadOnly	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerServiceAutoscaleRole	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerServiceEventsRole	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerServiceforEC2Role	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerServiceRole	AWS managed	0
<input checked="" type="checkbox"/> AmazonEC2FullAccess	AWS managed	0
<input type="checkbox"/> AmazonEC2ImageReferencesAccessPolicy	AWS managed	0

—> we have given s3 all access and ec2 all access to the user as permission

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 myuser	Console password type None	Require password reset 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](#)

—>for consol password

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 myuser	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
In addition to console access, users with SigninLocalDevelopmentAccess permissions can use the same console credentials for programmatic access without the need for access keys.

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](#)

[Cancel](#) [Next](#)

—> copy the console url and past in browser and add the details

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](#)

Step 1 Specify user details
Step 2 Set permissions
Step 3 Review and create
Step 4 Retrieve password

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://207662791773.signin.aws.amazon.com/console>

User name
[myuser](#)

Console password
[*****](#) [Show](#)

[Email sign-in instructions](#)

[Cancel](#) [Download .csv file](#) [Return to users list](#)

—> user created successfully

Identity and Access Management (IAM)

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	Group	Last activity	MFA	Password age	Console last sign-in	Access key ID
myuser	/	0	2 minutes				

[Delete](#) [Create user](#)

Console sign-in

Console sign-in link
<https://207662791773.signin.aws.amazon.com/console>

Console password
Updated Now (2025-11-27 15:18 GMT+5:30)

Last console sign-in
[Never](#)

—> account number is given with url number

IAM user sign in [?](#)

Account ID or alias (Don't have?)
207662791773

Remember this account

IAM username
myuser

Password

Show Password [Having trouble?](#)

[Sign in](#)

[Sign in using root user email](#)

[Create a new AWS account](#)

By continuing, you agree to [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.

aws

Amazon Lightsail
Lightsail is the easiest way to get started on AWS
[Learn more »](#)

→ login to the IAM USER

The screenshot shows the AWS Console Home page. At the top right, it displays "Account ID: 2076-6279-1773" and "myuser". The main content area is divided into several sections:

- Recently visited:** Shows a cube icon and a message "No recently visited services". Below it are links to EC2, S3, Aurora and RDS, and Lambda.
- Applications:** Shows 0 applications. A red box highlights an "Access denied" message for the "servicecatalog>ListApplications" action, with a "Diagnose with Amazon Q" button below it.
- Welcome to AWS:** Includes sections for "Getting started with AWS" (with a rocket icon) and "Training and certification" (with a person icon).
- AWS Health:** Shows "No health data" and a message "You don't have permissions to access".
- Cost and usage:** Shows current month, forecasted month end, and savings opportunities, all with "Access denied" messages.

→ given s3full access

The screenshot shows the Amazon S3 General purpose buckets page. At the top right, it displays "Account ID: 2076-6279-1773" and "myuser". The left sidebar includes sections for Buckets, Access management and security, and Storage management and insights. The main content area shows:

- General purpose buckets:** One bucket named "my-vpc-bucket-flowlog" is listed. It was created on November 23, 2025, at 20:17:26 (UTC+05:30). A "Create bucket" button is available.
- Account snapshot:** Provides visibility into storage usage and activity trends.
- External access summary - new:** Helps identify bucket permissions that allow public access or access from other AWS accounts.

→ given ec2 full access

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed, showing the main navigation menu. The main content area has a header "Instances Info" with a search bar and filters for "All states". It displays a message "No instances" and a "Launch instances" button. Below this, there's a section titled "Select an instance" with a dropdown menu.

AWS Account Information: Account ID: 2076-6279-1773, Region: Europe (Stockholm), User: myuser

EC2 > Instances

Instances Info

Find Instance by attribute or tag (case-sensitive)

All states

No instances

You do not have any instances in this region

Launch instances

Select an instance

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

Go to IAM Groups

1. Open AWS Console
 2. Go to **IAM**
 3. On the left menu → click **User groups**
 4. Click **Create group**

IAM > User groups > Create user group

entity and Access Management (IAM) <

Search IAM

ashboard

ss management

r groups

s

is

ties

ility providers

ount settings

t access management

porary delegation requests

ess reports

ss Analyzer

source analysis New

Jnused access

analyzer settings

ential report

anization activity

rice control policies

urce control policies

Create user group

Name the group

User group name

Enter a meaningful name to identify this group.

EC2-ReadOnly-Group

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

Add users to the group - Optional (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.

<input type="checkbox"/>	User name	Group:	Last activity	Creation time
<input type="checkbox"/>	myuser	0	9 minutes ago	15 minutes ago

Attach permissions policies - Optional (1/1096) 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

<input type="checkbox"/>	Policy name	Type	Used as	Description
<input checked="" type="checkbox"/>	AmazonEC2ReadOnlyAccess	AWS managed	None	Provides read only access to Amazon E...

[Cancel](#) [Create user group](#)

- > give the group name as EC2-readonly-group
- > attach the permission

Attach Policy

- > In the permissions list, search:

AmazonEC2ReadOnlyAccess

- > and then click create

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

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

In addition to console access, users with SigninLocalDevelopmentAccess permissions can use the same console credentials for programmatic access without the need for access keys.

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](#)

Cancel Next

—> give name devops while creating another user

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)

Group name	Users	Attached policies	Created
EC2-ReadOnly-Group	0	AmazonEC2ReadOnlyAccess	2025-11-27 (2 minutes ago)

Set permissions boundary - optional

Create group

Cancel Previous Next

→ add user to group

The screenshot shows the 'Review and create' step of the IAM 'Create user' wizard. On the left, a vertical navigation bar lists steps: Step 1 (Specify user details), Step 2 (Set permissions), Step 3 (Review and create) which is selected, and Step 4 (Retrieve password). The main area contains three sections: 'User details' (User name: devops, Console password type: Autogenerated, Require password reset: Yes), 'Permissions summary' (listing EC2-ReadOnly-Group as a Group with Permissions group and IAMUserChangePassword as an AWS managed policy with Permissions policy), and 'Tags - optional' (with an 'Add new tag' button). At the bottom right are 'Cancel', 'Previous', and 'Create user' buttons.

After creating the devops user go and check the user in that group
Which we created in task2

→ devops named user is available in EC2-ReadOnly-Group

The screenshot shows the 'EC2-ReadOnly-Group' page in the IAM console. The left sidebar has 'Identity and Access Management (IAM)' selected. Under 'Access management', 'User groups' is also selected. The main content area shows the 'Summary' of the group, including its ARN (arn:aws:iam::207662791773:group/EC2-ReadOnly-Group). Below this, the 'Users' tab is selected, showing one user named 'devops'. There are buttons for 'Edit', 'Delete', 'Remove', and 'Add users'. A table at the bottom lists the user 'devops' with details like Groups (none), Last activity (7 minutes ago), and Creation time (9 minutes ago).

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

- > go to your terminal
- > aws configure (for configuring the aws account in your terminal)
- > give the details and enter
- > aws iam list-users (will lists the users in iam)
- > we have already created two iam users one myuser from task1 and devops user from task3

```
[waseemakram@waseem downloads % aws configure
AWS Access Key ID [*****5PVR]: AKIATAWNKXR02WUZEQMJ
AWS Secret Access Key [*****9e7Y]: 9aWGEpbK/gs30FsJ4H3eCdZ10je+oIiZIwusH4pd
Default region name [eu-north-1]:
Default output format [json]:
[waseemakram@waseem downloads % aws iam list-users

{
  "Users": [
    {
      "Path": "/",
      "UserName": "devops",
      "UserId": "AIDATAWNKXR03HF3FXC75",
      "Arn": "arn:aws:iam::207662791773:user/devops",
      "CreateDate": "2025-11-27T10:07:44+00:00",
      "PasswordLastUsed": "2025-11-27T10:10:04+00:00"
    },
    {
      "Path": "/",
      "UserName": "myuser",
      "UserId": "AIDATAWNKXR0QU2ECUNPA",
      "Arn": "arn:aws:iam::207662791773:user/myuser",
      "CreateDate": "2025-11-27T09:45:23+00:00",
      "PasswordLastUsed": "2025-11-27T10:11:34+00:00"
    }
  ]
}
[waseemakram@waseem downloads % vi create_vpc_user.sh
```

- > now we have to write a script for creating the iam user

- > after aws configure
- > create a file for bashscripting for creating a new iam user
- > we created a file name create_vpc_user.sh as file name
- > added script Inside it

```
[waseemakram@waseem downloads % vi create_vpc_user.sh
[waseemakram@waseem downloads % chmod 755 create_vpc_user.sh
[waseemakram@waseem downloads % ./create_vpc_user.sh
Creating IAM user: vpc-user
{
  "User": {
    "Path": "/",
    "UserName": "vpc-user",
    "UserId": "AIDATAWNKXR04MXHN4TXR",
    "Arn": "arn:aws:iam::207662791773:user/vpc-user",
    "CreateDate": "2025-11-27T11:41:31+00:00"
  }
}
User 'vpc-user' created and VPC Full Access policy attached.
waseemakram@waseem downloads %
```

- > this the script we used to create a iam user

```
#!/bin/bash

USERNAME="vpc-user"

echo "Creating IAM user: $USERNAME"

aws iam create-user --user-name $USERNAME

aws iam attach-user-policy \
--user-name $USERNAME \
--policy-arn arn:aws:iam::aws:policy/AmazonVPCFullAccess

echo "User '$USERNAME' created and VPC Full Access policy attached."
```

- > now go and check in IAM —> USERS
- > we will find the created user inside the users

- > we created vpc-user inside our terminal and we can find that user in side IAM

	User name	▲	Path	▼	Groups	▼	Last activity	▼	MFA	▼	Password age	▼	Console last sign-in	▼	Access key ID
□	devops		/		1		1 hour ago		-		1 hour		1 hour ago		-
□	myuser		/		0		1 hour ago		-		1 hour		1 hour ago		-
□	vpc-user		/		0		-		-		-		-		-

- > aws iam list-users will display all our iam users inside our aws

```
waseemakram@waseem downloads % aws iam list-users
```

```
{  
    "Users": [  
        {  
            "Path": "/",  
            "UserName": "devops",  
            "UserId": "AIDATAWNKXR03HF3FXC75",  
            "Arn": "arn:aws:iam::207662791773:user/devops",  
            "CreateDate": "2025-11-27T10:07:44+00:00",  
            "PasswordLastUsed": "2025-11-27T10:10:04+00:00"  
        },  
        {  
            "Path": "/",  
            "UserName": "myuser",  
            "UserId": "AIDATAWNKXR0QU2ECUNPA",  
            "Arn": "arn:aws:iam::207662791773:user/myuser",  
            "CreateDate": "2025-11-27T09:45:23+00:00",  
            "PasswordLastUsed": "2025-11-27T10:11:34+00:00"  
        },  
        {  
            "Path": "/",  
            "UserName": "vpc-user",  
            "UserId": "AIDATAWNKXR04MXHN4TXR",  
            "Arn": "arn:aws:iam::207662791773:user/vpc-user",  
            "CreateDate": "2025-11-27T11:41:31+00:00"  
        }  
    ]  
}
```

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

→ to create a iam policy to a user
→ iam →policys→create policy

1. Go to **IAM → Policies → Create policy**
2. Select **JSON** -->give the policy to allow ec2 access for user in
3. Click **Next → Create policy**

The screenshot shows the AWS IAM 'Create policy' interface. At the top, there are tabs for 'Visual' (disabled), 'JSON' (selected), and 'Actions'. Below the tabs, the title 'Specify permissions' is shown with a 'Info' link. A note says 'Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.' On the left, a sidebar shows 'View and create' and 'Policy editor' with line numbers 1 through 28. The JSON code is as follows:

```
1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Sid": "AllowEC2Stockholm",
6        "Effect": "Allow",
7        "Action": "ec2:*",
8        "Resource": "*",
9        "Condition": {
10          "StringEquals": {
11            "aws:RequestedRegion": "eu-north-1"
12          }
13        }
14      },
15      {
16        "Sid": "DenyEC2OtherRegions",
17        "Effect": "Deny",
18        "Action": "ec2:*",
19        "Resource": "*",
20        "Condition": {
21          "StringNotEquals": {
22            "aws:RequestedRegion": "eu-north-1"
23          }
24        }
25      }
26    ]
27  }
```

On the right, there are sections for 'Edit statement' (with a 'Remove' button), 'Add actions' (with a 'Choose a service' dropdown and a search bar), 'Included' (with 'EC2' selected), 'Available' (with options like 'AI Operations', 'AMP', 'API Gateway', etc.), and 'Add a resource' (with an 'Add' button). The status bar at the bottom shows '1 / 1'.

This policy will:

- **Allow EC2 full access in Stockholm (eu-north-1)**
- **Deny EC2 access in every other region**
- **Apply to a specific IAM user**

—> next we have to give the policy name which we are creating

Review the permissions, specify details, and tags.

Step 2
Review and create

Policy details

Policy name
Enter a meaningful name to identify this policy.
EC2-Only-Stockholm
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 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.

Allow (1 of 454 services)

Show remaining 453 services

Service	Access level	Resource	Request condition
EC2	Full access	All resources	aws:RequestedRegion = eu-north-1

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.

—> Attach this policy to specific user
—> iam —> users —>devops

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 (1424)

Filter by Type: All types | 1 match

Policy name	Type	Attached entities
EC2-Only-Stockholm	Customer managed	0

Cancel **Next**

—>Go to IAM → Users

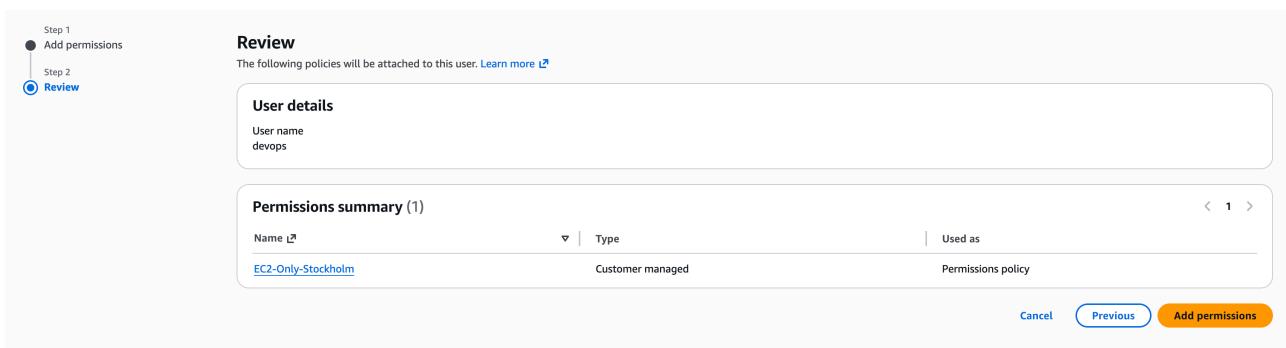
—>Click the user you want (example:we are having user devops)

—>Click Add permissions

—>Choose Attach policies directly

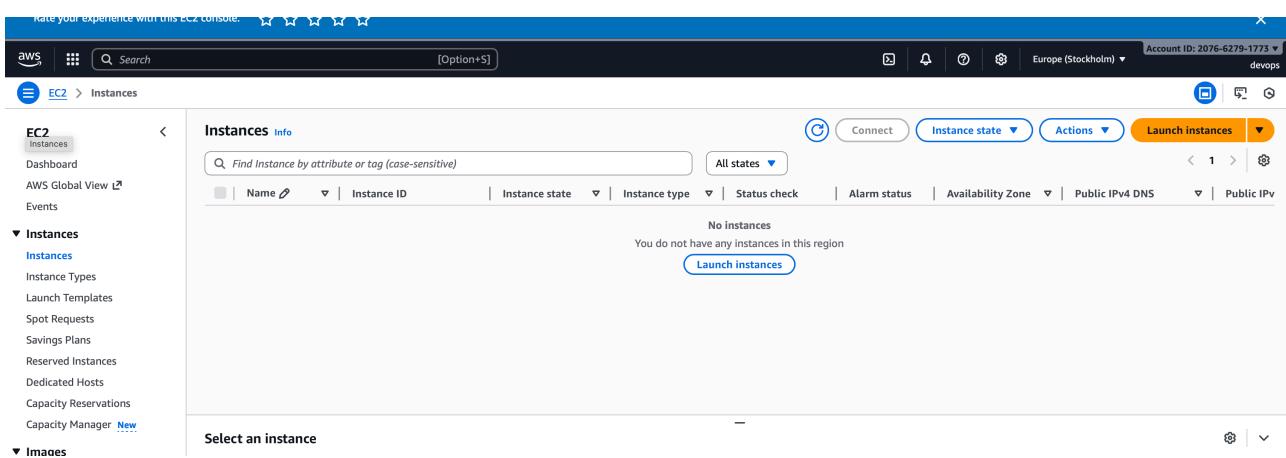
—>Select the policy you created

—>Click Add permissions



—>verification

—> as we have given only eu-north-1 (stockholm)



—> we are able to have access to that region

—>as I changed to other regions like Virginia we are getting access denied

The screenshot shows the AWS EC2 Instances page. The left sidebar is titled 'EC2' and includes links for Dashboard, EC2 Global View, Events, Instances (which is expanded), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, and Capacity Reservations. The main content area has a header 'Instances Info' with filters for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IPv6. Below the header is a search bar with placeholder 'Find Instance by attribute or tag (case-sensitive)' and a dropdown set to 'All states'. A red-bordered error message box contains the text: 'You are not authorized to perform this operation. User: arn:aws:iam::207662791773:user/devops is not authorized to perform: ec2:DescribeInstances with an explicit deny in an identity-based policy'. There are 'Retry' and 'Diagnose with Amazon Q' buttons at the bottom of the message box.

They will get **AccessDenied**.

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

—> so I have accessed the bucket of my friend

—> so the account A is me and my friend is account B

So from his side he has created a bucket policy for me to have access To his bucket

The screenshot shows the 'Bucket policy' section of an S3 bucket's properties. At the top, it says 'Bucket policy' and provides a link to learn more about bucket policies. It also notes that public access is blocked because Block Public Access settings are turned on for this bucket. The policy itself is displayed in JSON format:

```
{ "Version": "2012-10-17", "Statement": [ { "Sid": "AllowAccountAUserAccess", "Effect": "Allow", "Principal": { "AWS": "arn:aws:iam::207662791773:root" }, "Action": [ "s3:ListBucket", "s3:GetObject" ], "Resource": [ "arn:aws:s3:::friend-cross-bucket", "arn:aws:s3:::friend-cross-bucket/*" ] } ] }
```

There are 'Edit' and 'Delete' buttons above the JSON code. At the bottom, there is an 'Object Ownership' section with a note about control ownership and access control lists (ACLs).

—> so the bucket name is friend-cross-bucket
We have attached the bucket and added my aws account id with user to send there access
—> so after creating the bucket policy I have to go to the IAM

—>users —> devops—>create policy and add the policy of bucket of friend

The screenshot shows the 'Specify permissions' step of the IAM policy creation wizard. The left sidebar shows 'Step 1 Specify permissions' is selected. The main area is titled 'Policy editor' and contains a JSON code block:

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": [  
7         "s3:ListBucket",  
8         "s3:GetObject"  
9       ],  
10      "Resource": [  
11        "arn:aws:s3:::friend-cross-bucket",  
12        "arn:aws:s3:::friend-cross-bucket/*"  
13      ]  
14    }  
15  ]  
16}  
17
```

Below the JSON code, there are buttons for '+ Add new statement' and 'Edit statement'. A note indicates '1864 of 2048 characters remaining'. The right sidebar shows a placeholder for a statement with a button '+ Add new statement'.

—> we have added the bucket name and allow the access of that bucket

The screenshot shows the 'Review and create' step of the IAM policy creation wizard. The left sidebar shows 'Step 1 Specify permissions' is completed. The main area has a title 'Policy details' and a 'Policy name' input field containing 'AccessFriendBucket'. Below it, a note says 'Maximum 128 characters. Use alphanumeric and '+-_@-' characters.' The 'Permissions defined in this policy' section lists 'Allow (1 of 454 services)' for the service 'S3' with the action 'Limited: List, Read' and resource 'Multiple'. A note says 'Show remaining 453 services'. At the bottom, there are 'Cancel', 'Previous', and 'Create policy' buttons.

And for configuring we have to go to terminal were our aws access is there

```
waseemakram@waseem downloads % aws s3 ls s3://friend-cross-bucket  
2025-11-28 12:36:24      1237950 Bash_script -2.pdf  
waseemakram@waseem downloads %
```

—> aws s3 ls://friend-cross-bucket as we can see we are able to have access of my friend bucket which he has

