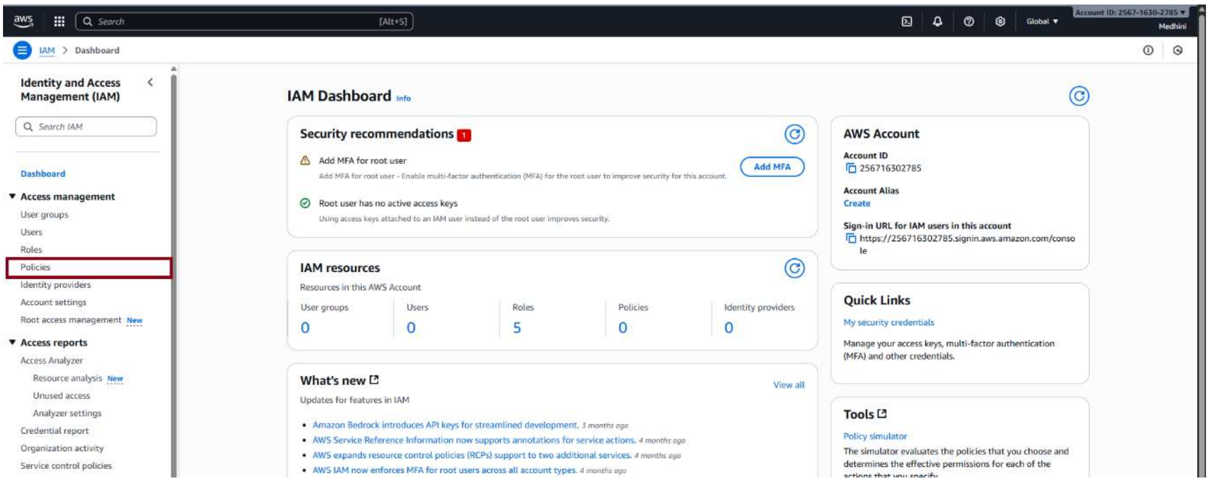
1. Create policy number 1 which lets the users to:

a) Access S3 completely
b) Only create EC2 instances
c) Full access to RDS

Step 1: Login to your account and search for **IAM** and click on **IAM,** click on **“Policies”** as shown in below picture



Security recommendations 1

Add MFA for root user

Add MFA for root user - Enable multi-factor authentication (MFA) for the root user to improve security for this account.

Root user has no active access keys

Using access keys attached to an IAM user instead of the root user improves security.

IAM resources

Resources in this AWS Account

User groups	Users	Roles	Policies	Identity providers
0	0	5	0	0

What's new

Updates for features in IAM

- Amazon Bedrock introduces API keys for streamlined development. *3 months ago*
- AWS Service Reference Information now supports annotations for service actions. *4 months ago*
- AWS expands resource control policies (RCPs) support to two additional services. *4 months ago*
- AWS IAM now enforces MFA for root users across all account types. *4 months ago*

AWS Account

Account ID
256716302785

Account Alias
Create

Sign-in URL for IAM users in this account
https://256716302785.signin.aws.amazon.com/console

Quick Links

My security credentials

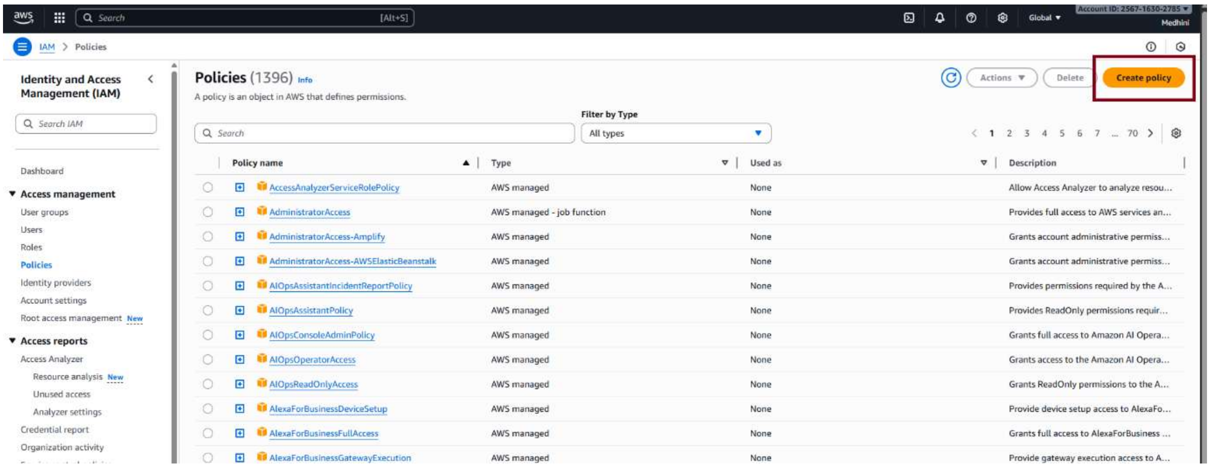
Manage your access keys, multi-factor authentication (MFA) and other credentials.

Tools

Policy simulator

The simulator evaluates the policies that you choose and determines the effective permissions for each of the actions that you specify

Step 2: The below dashboard will appear, now click on **“Create Policy”** as shown in below picture

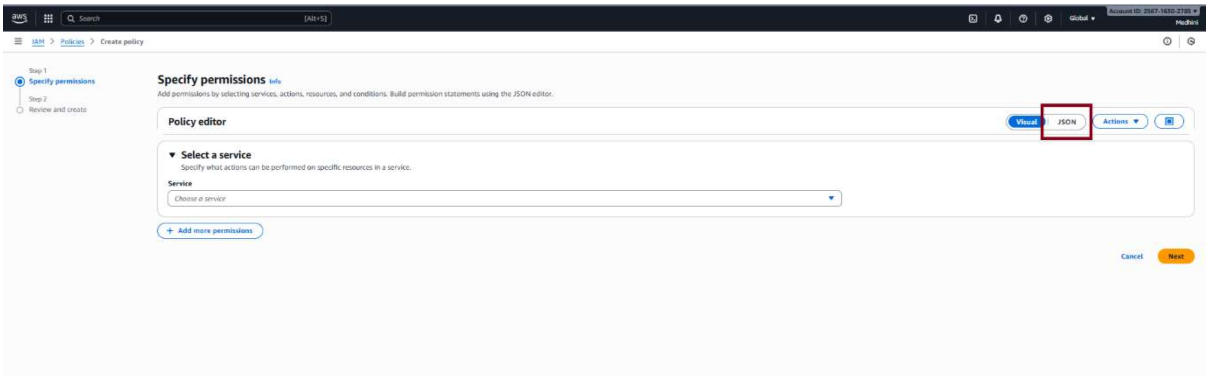


Policies (1396) Info

A policy is an object in AWS that defines permissions.

Policy name	Type	Used as	Description
AccessAnalyzerServiceRolePolicy	AWS managed	None	Allow Access Analyzer to analyze resou...
AdministratorAccess	AWS managed - job function	None	Provides full access to AWS services an...
AdministratorAccess-Amplify	AWS managed	None	Grants account administrative permiss...
AdministratorAccess-AWSElasticBeanstalk	AWS managed	None	Grants account administrative permiss...
AIOpsAssistantIncidentReportPolicy	AWS managed	None	Provides permissions required by the A...
AIOpsAssistantPolicy	AWS managed	None	Provides ReadOnly permissions requir...
AIOpsConsoleAdminPolicy	AWS managed	None	Grants full access to Amazon AI Opera...
AIOpsOperatorAccess	AWS managed	None	Grants access to the Amazon AI Opera...
AIOpsReadOnlyAccess	AWS managed	None	Grants ReadOnly permissions to the A...
AlexaForBusinessDeviceSetup	AWS managed	None	Provide device setup access to AlexaFo...
AlexaForBusinessFullAccess	AWS managed	None	Grants full access to AlexaForBusiness ...
AlexaForBusinessGatewayExecution	AWS managed	None	Provide gateway execution access to A...

Step 3: The below dashboard appears, in the step 1, you should click on **“JSON”** as shown in below picture



Specify permissions Info

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

Policy editor

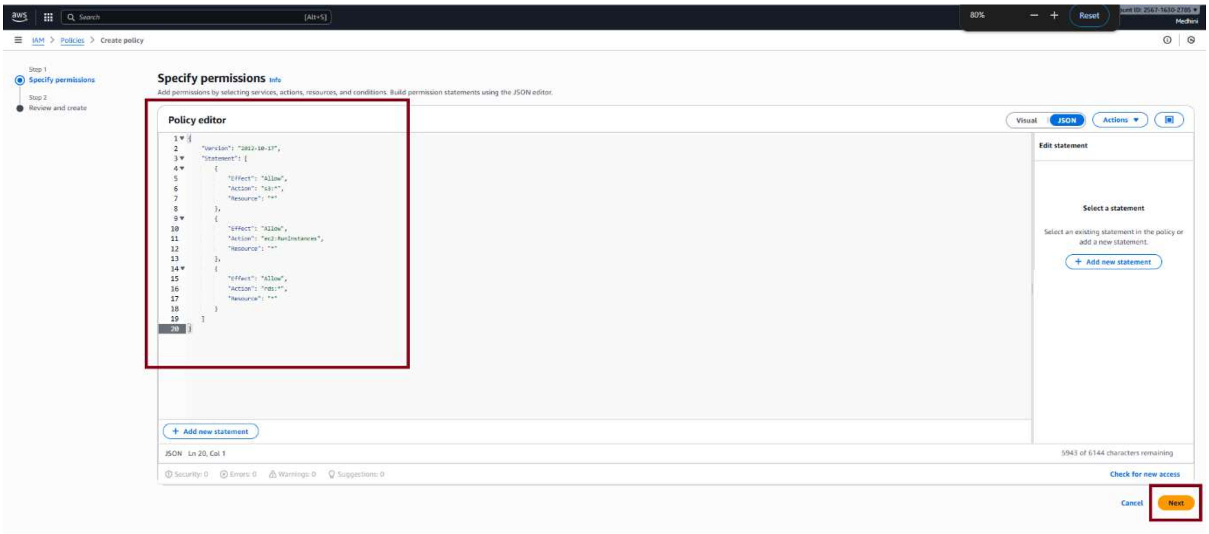
Select a service

Specify what actions can be performed on specific resources in a service.

Service: Choose a service

Step 4: Add the Policy mentioned below, and click on **“Next”** as shown in below picture

```
{
        "Version": "2012-10-17",
        "Statement": [
                {
                        "Effect": "Allow",
                        "Action": "s3:*",
                        "Resource": "*"
                },
                {
                        "Effect": "Allow",
                        "Action": "ec2:RunInstances",
                        "Resource": "*"
                },
                {
                        "Effect": "Allow",
                        "Action": "rds:*",
                        "Resource": "*"
                }
        ]
}
```



IAM > Policies > Create policy

Step 1 Specify permissions

Step 2 Review and create

Specify permissions

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

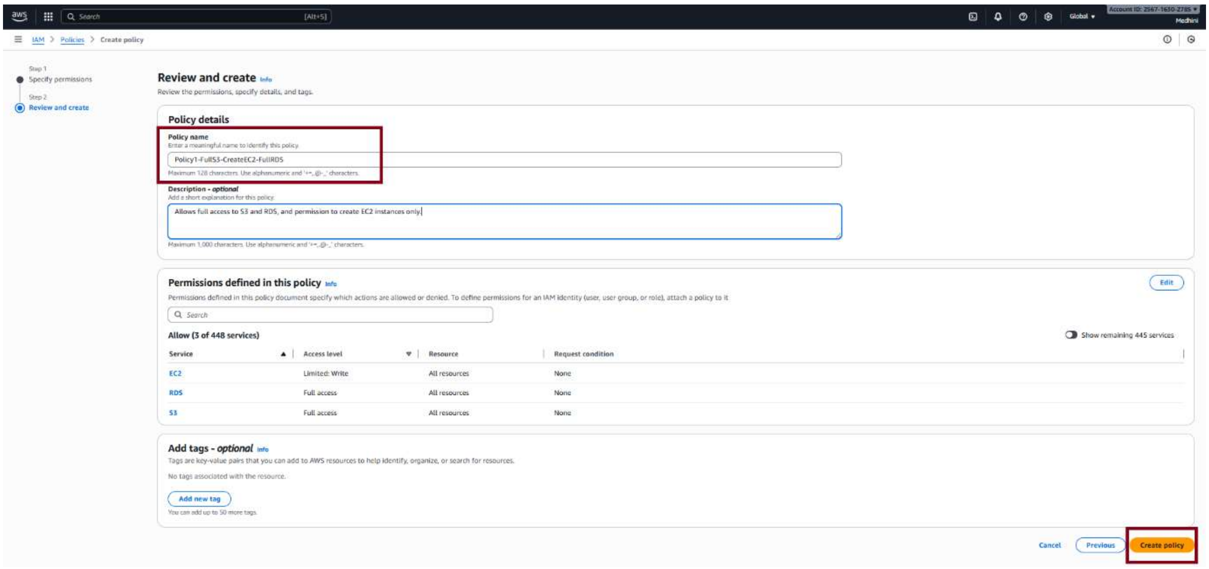
Policy editor

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "rds:*",
            "Resource": "*"
        }
    ]
}
```

JSON Ln 20, Col 1

5943 of 6144 characters remaining

Step 5: Give name for policy and (description is optional) and then click on **“Create Policy”**



IAM > Policies > Create policy

Step 1 Specify permissions

Step 2 Review and create

Review and create

Review the permissions, specify details, and tags.

Policy details

Policy name: Policy1-FullS3-CreateEC2-FullRDS

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

Description - *optional*: Allows full access to S3 and RDS, and permission to create EC2 instances only.

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

Permissions defined in this policy

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 (3 of 448 services)

Service	Access level	Resource	Request condition
EC2	Limited: Write	All resources	None
RDS	Full access	All resources	None
S3	Full access	All resources	None

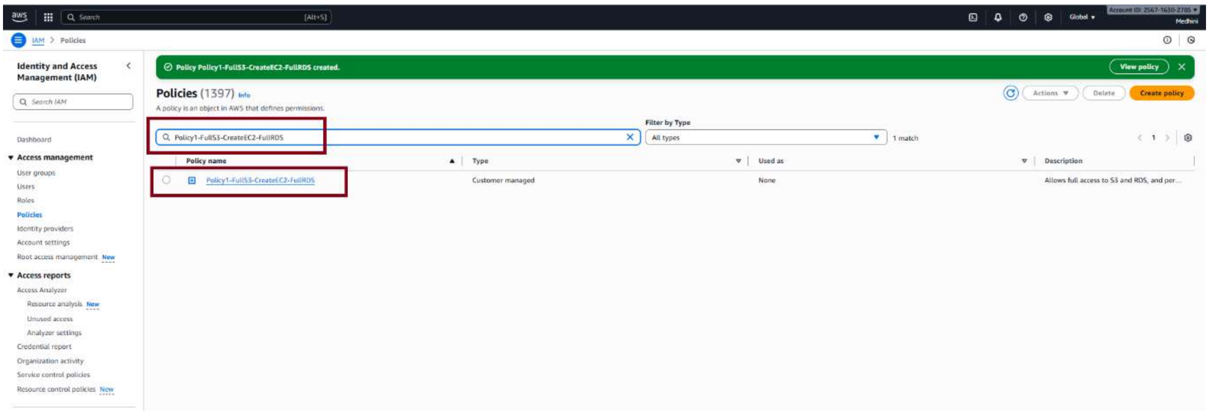
Add tags - *optional*

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.

You can add up to 50 more tags.

Step 6: Once Policy is created go to search and you can search your Policy name once done as shown in below picture. Now your policy is created accordingly which **Allows full access to S3 and RDS, and permission to create EC2 instances only.**



IAM > Policies

Policy Policy1-FullS3-CreateEC2-FullRDS created.

Policies (1397)

A policy is an object in AWS that defines permissions.

Search: Policy1-FullS3-CreateEC2-FullRDS

Filter by Type: All types

1 match

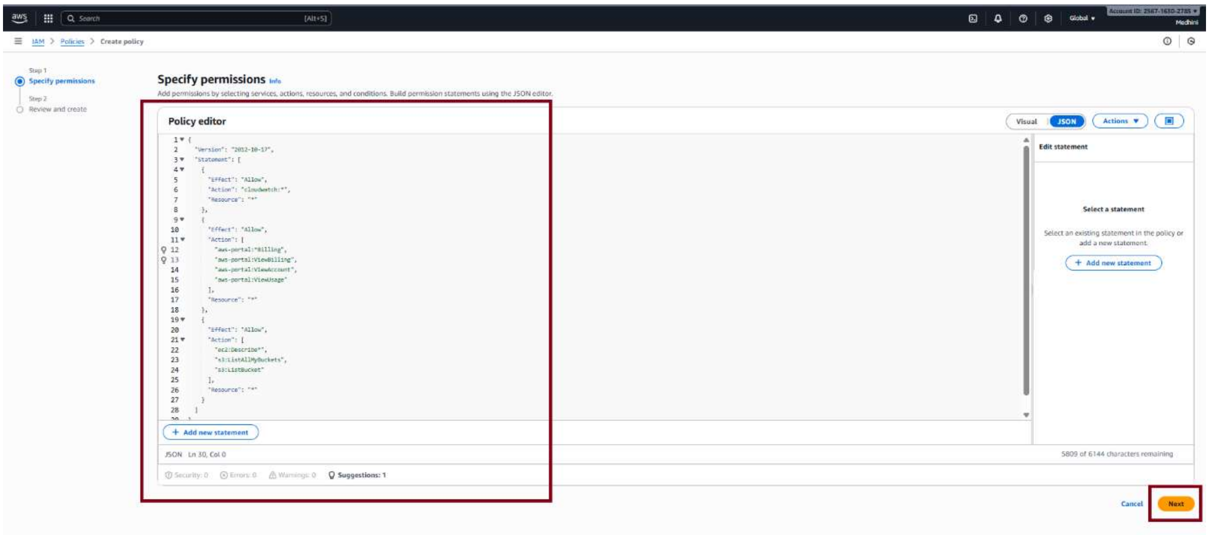
Policy name	Type	Used as	Description
Policy1-FullS3-CreateEC2-FullRDS	Customer managed	None	Allows full access to S3 and RDS, and per...

2. Create a policy number 2 which allows the users to:

a) Access CloudWatch and billing completely

b) Can only list EC2 and S3 resources

Step 1: Follow the above from step1 to step 3 and the paste the below code on **"JSON"** tab, then click on **"Next"** as shown in below picture,



Specify permissions Info

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

Policy editor

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "cloudwatch:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "aws-portal:*Billing",
        "aws-portal:ViewBilling",
        "aws-portal:ViewAccount",
        "aws-portal:ViewUsage"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:Describe*",
        "s3:ListAllMyBuckets",
        "s3:ListBucket"
      ],
      "Resource": "*"
    }
  ]
```

JSON Ln 30, Col 0

5809 of 6144 characters remaining

Security: 0 Errors: 0 Warnings: 0 Suggestions: 1

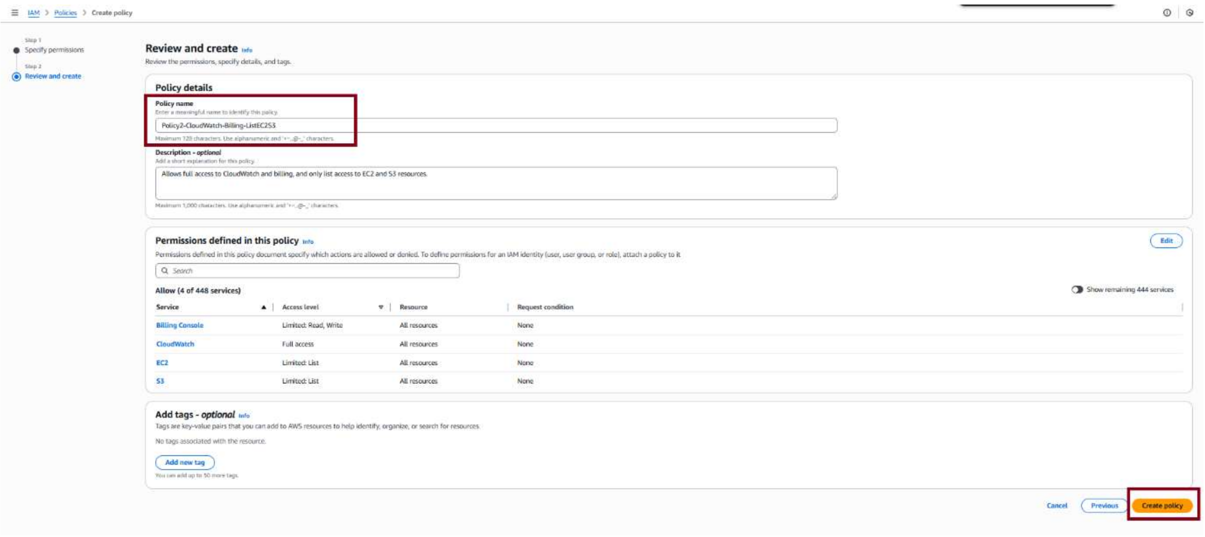
Edit statement

Select a statement

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

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
    "Effect": "Allow",
    "Action": "cloudwatch:*",
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "aws-portal:*Billing",
      "aws-portal:ViewBilling",
      "aws-portal:ViewAccount",
      "aws-portal:ViewUsage"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:Describe*",
      "s3:ListAllMyBuckets",
      "s3:ListBucket"
    ],
    "Resource": "*"
  }
 ]
}
```

Step 2: Give a name for **“Policy”** and click on **“Create policy”** as shown in below picture



Review and create

Review the permissions, specify details, and tags.

Policy details

Policy name

Enter a meaningful name to identify this policy.

Policy2-CloudWatch-Billing-ListEC2S3

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

Description - *optional*

Add a short explanation for this policy.

Allows full access to CloudWatch and billing, and only list access to EC2 and S3 resources.

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

Permissions defined in this policy

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 (4 of 448 services)

Service	Access level	Resource	Request condition
Billing Console	Limited: Read, Write	All resources	None
CloudWatch	Full access	All resources	None
EC2	Limited: List	All resources	None
S3	Limited: List	All resources	None

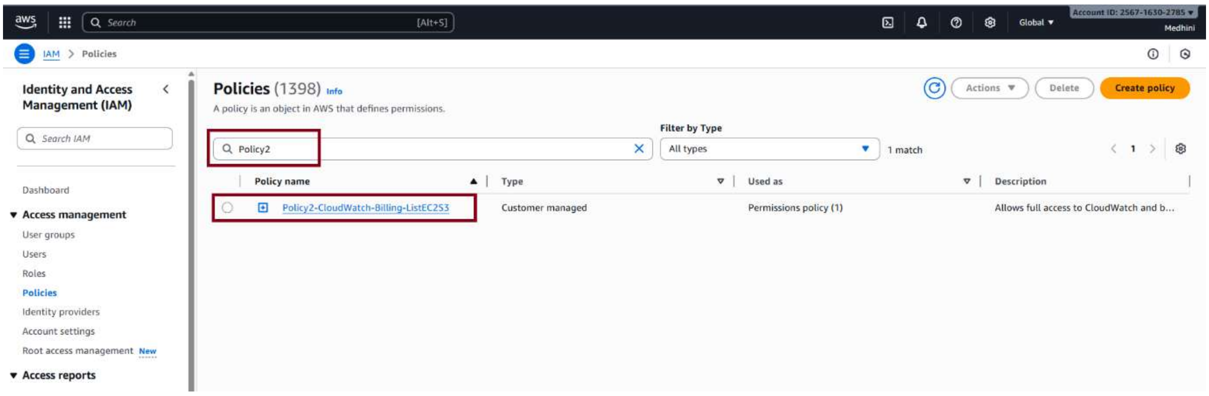
Add tags - *optional*

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.

You can add up to 50 more tags.

Step 3: Once Policy is created go to search and you can search your Policy name once done as shown in below picture. Now your policy is created accordingly which **Allows full access to CloudWatch and billing, and only list access to EC2 and S3 resources.**



IAM > Policies

Identity and Access Management (IAM)

Policies (1398) Info

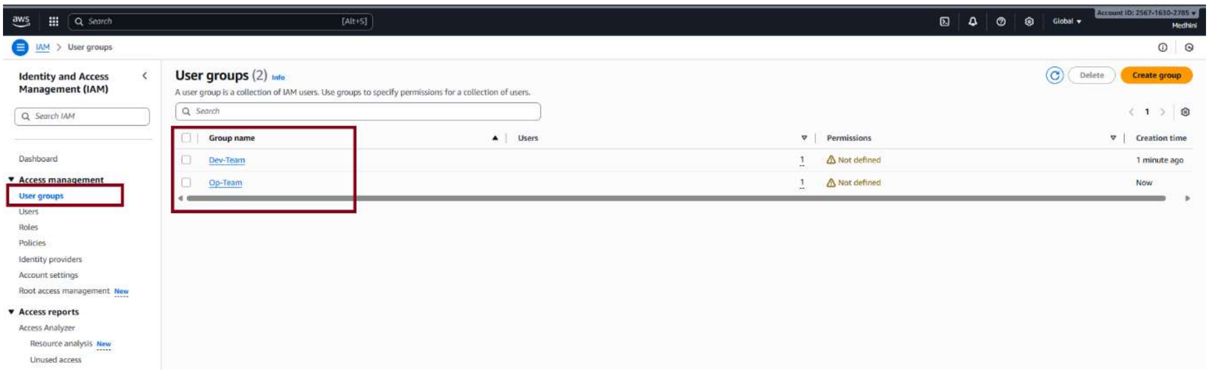
A policy is an object in AWS that defines permissions.

Search: Policy2 — Filter by Type: All types — 1 match

Policy name	Type	Used as	Description
Policy2-CloudWatch-Billing-ListEC2S3	Customer managed	Permissions policy (1)	Allows full access to CloudWatch and b...

Attach Policy Number 1 to “Dev-Team”

Step 1: Click on **“User Groups”,** the two user groups which you have created will be visible a shown in below picture



IAM > User groups

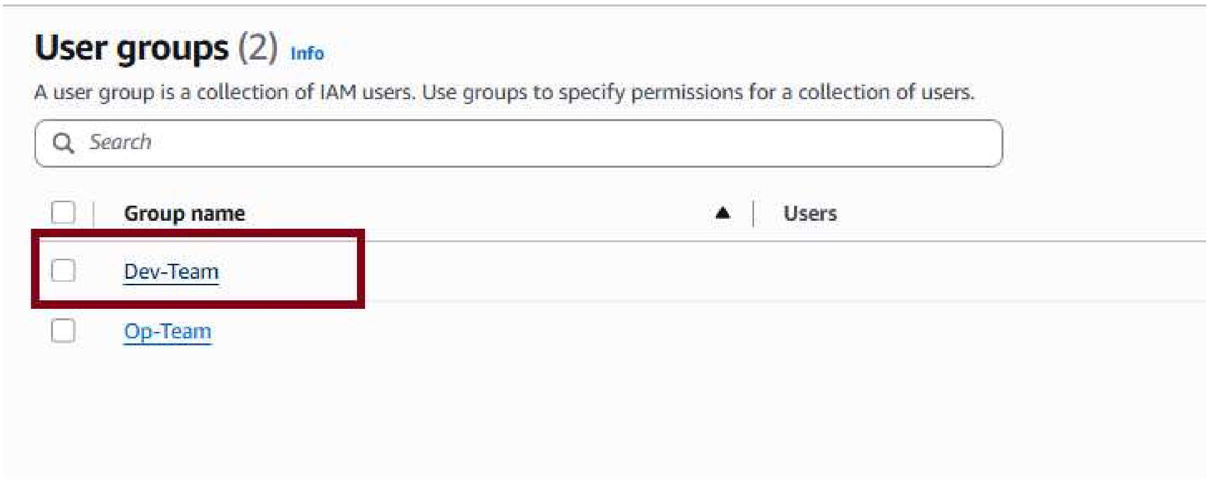
Identity and Access Management (IAM)

User groups (2) 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
Dev-Team	1	⚠ Not defined	1 minute ago
Op-Team	1	⚠ Not defined	Now

Step 2: Click on “User Group” as shown in below picture

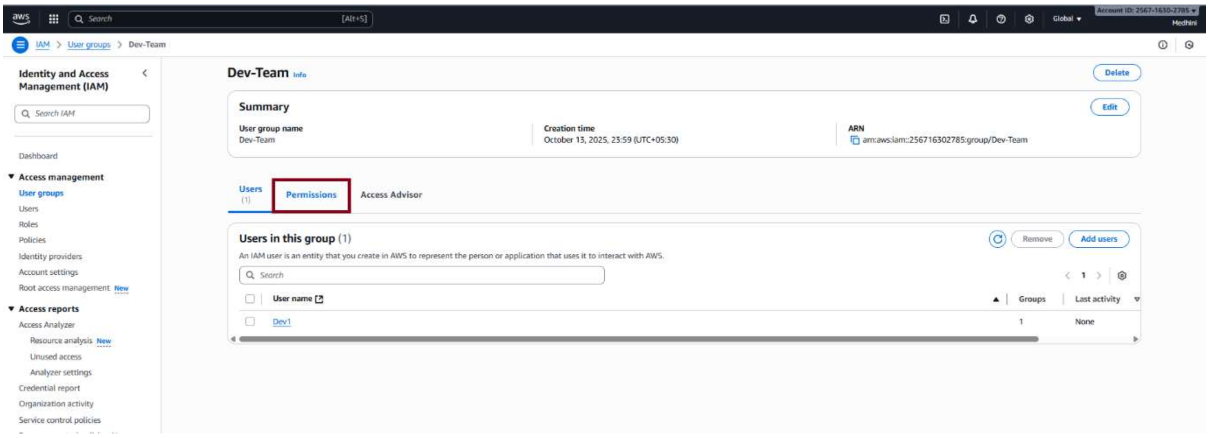


User groups (2) Info

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

Group name	Users
Dev-Team	
Op-Team	

Step 3: After the Step 2, the below dashboard will appear, Click on **"Permissions"**



Dev-Team Info

Summary

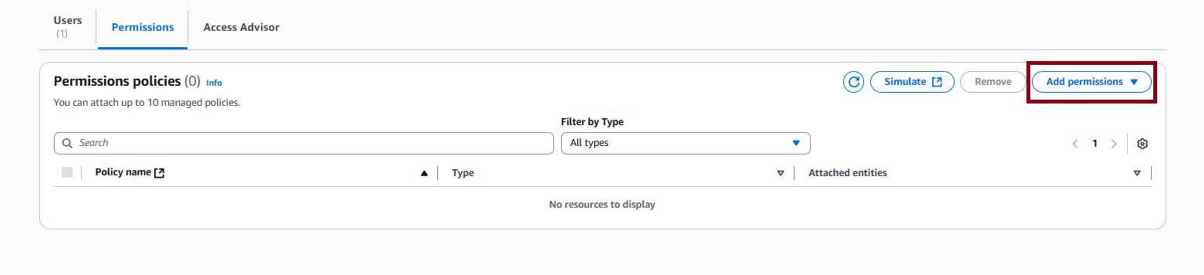
User group name	Creation time	ARN
Dev-Team	October 13, 2025, 23:59 (UTC+05:30)	arn:aws:iam::256716302785:group/Dev-Team

Users in this group (1)

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
Dev1	1	None

Step 4: Click on **"Add Permission"** as shown in below picture



Permissions policies (0)

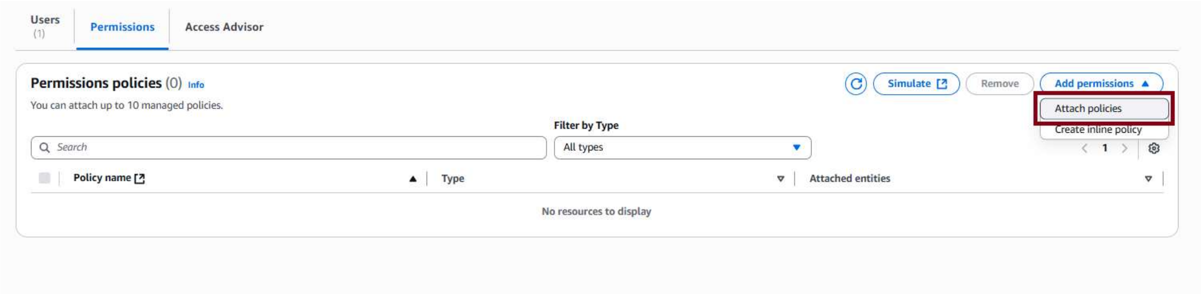
You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities

No resources to display

Step 5: Click on **"Attach Policies"** as shown in below picture



Permissions policies (0)

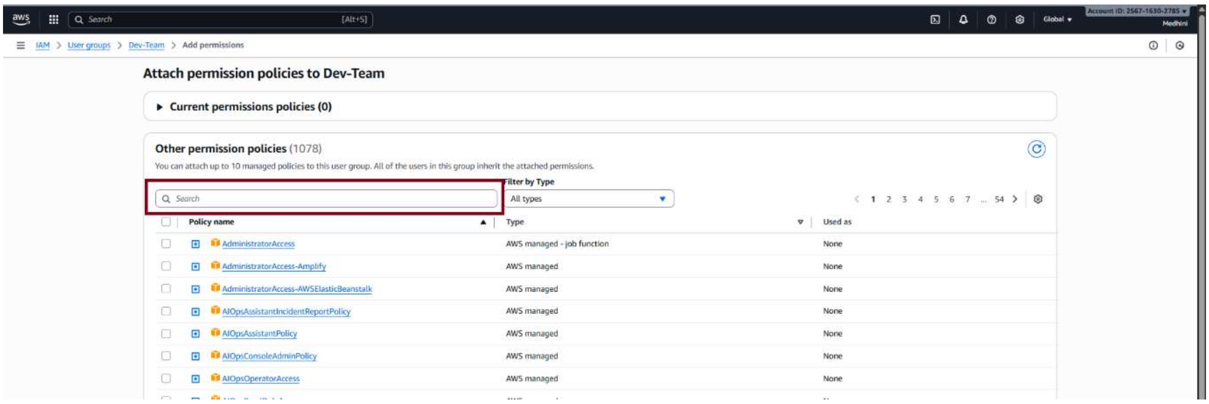
You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities

No resources to display

Step 6: In the below Dashboard, go to search



aws
Search [Alt+5]
Global ▾
Account ID: 2567-1630-2785 ▾
Medhini
IAM > User groups > Dev-Team > Add permissions
Search
All types
< 1 2 3 4 5 6 7 ... 54 >

Attach permission policies to Dev-Team

▸ Current permissions policies (0)

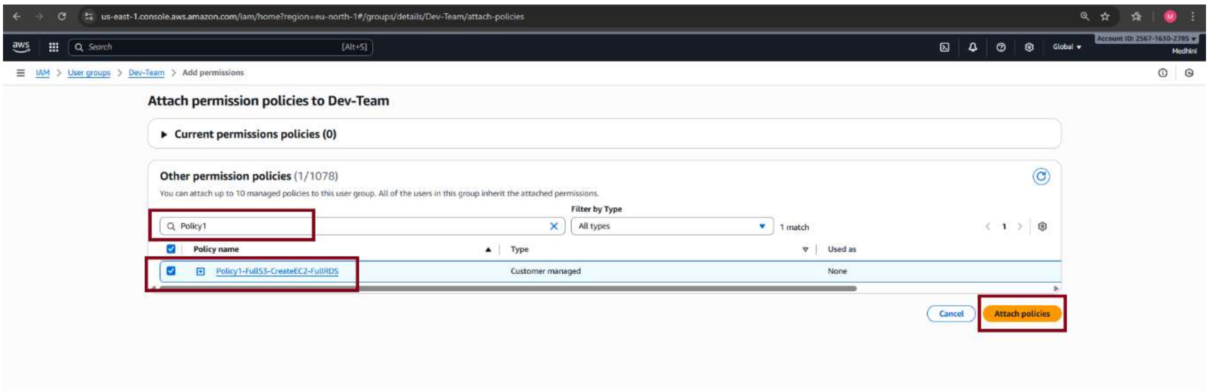
Other permission policies (1078)

You can attach up to 10 managed policies to this user group. All of the users in this group inherit the attached permissions.

Filter by Type: All types

Policy name	Type	Used as
AdministratorAccess	AWS managed - job function	None
AdministratorAccess-Amplify	AWS managed	None
AdministratorAccess-AWSElasticBeanstalk	AWS managed	None
AIOpsAssistantIncidentReportPolicy	AWS managed	None
AIOpsAssistantPolicy	AWS managed	None
AIOpsConsoleAdminPolicy	AWS managed	None
AIOpsOperatorAccess	AWS managed	None

Step 7: Search the policy name which is to be attached, and click on **"Attach Policy"** as shown in below picture



us-east-1.console.aws.amazon.com/iam/home?region=eu-north-1#/groups/details/Dev-Team/attach-policies

IAM > User groups > Dev-Team > Add permissions

Attach permission policies to Dev-Team

▸ Current permissions policies (0)

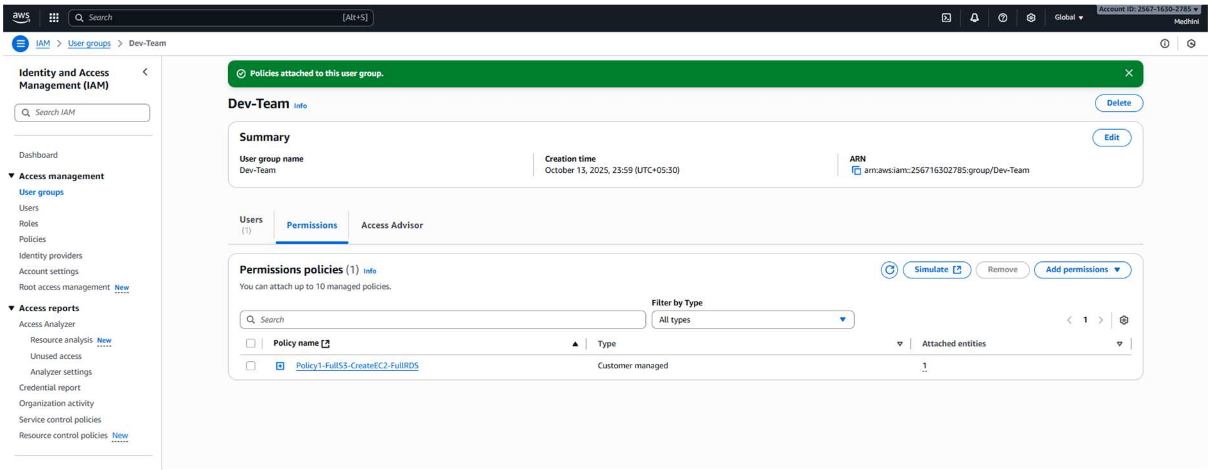
Other permission policies (1/1078)

You can attach up to 10 managed policies to this user group. All of the users in this group inherit the attached permissions.

Search: Policy1 — Filter by Type: All types — 1 match

☑	Policy name	Type	Used as
☑	Policy1-FullS3-CreateEC2-FullRDS	Customer managed	None

Step 8: The Policy number 1 is attached to **"Dev-Team"** as shown in below picture



IAM > User groups > Dev-Team

Policies attached to this user group.

Dev-Team Info

Summary

User group name: Dev-Team

Creation time: October 13, 2025, 23:59 (UTC+05:30)

ARN: arn:aws:iam::256716302785:group/Dev-Team

Permissions policies (1) Info

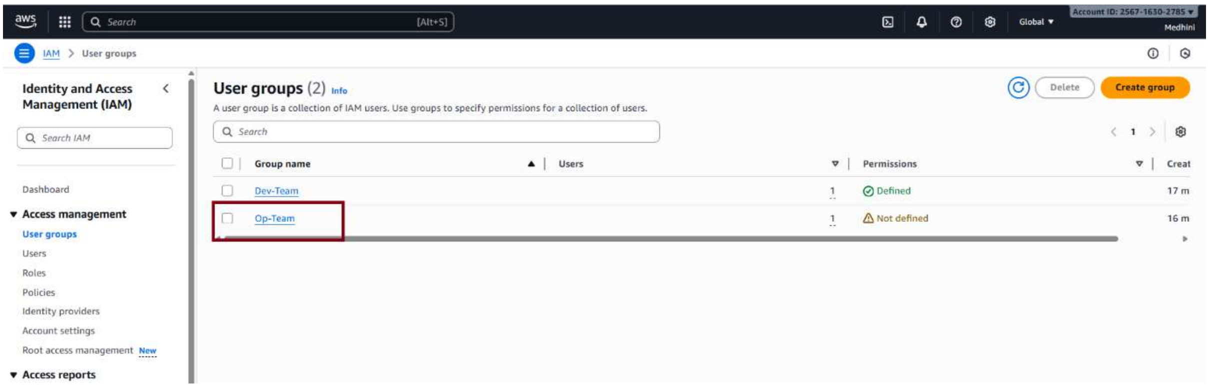
You can attach up to 10 managed policies.

Filter by Type: All types

☐	Policy name	Type	Attached entities
☐	Policy1-FullS3-CreateEC2-FullRDS	Customer managed	1

Attach Policy Number 2 to "Op-Team"

Step 1: Go to user groups and click on **"Op-Team"** user group as shown in below picture



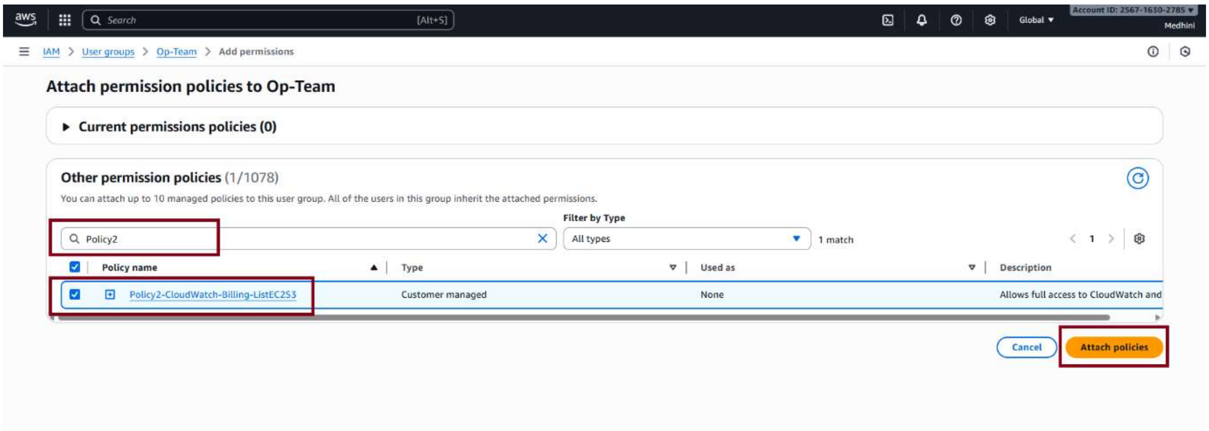
IAM > User groups

User groups (2) Info

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

☐	Group name	Users	Permissions	Creat
☐	Dev-Team	1	Defined	17 m
☐	Op-Team	1	Not defined	16 m

Step 2: Follow the above steps from ***Step 3 - Step 6*** and add Policy2 which we have previously created then click on **"Attach Policy"**


aws
Search [Alt+S]
Global ▾
Account ID: 2567-1630-2785 ▾
Medhini
IAM > User groups > Op-Team > Add permissions
Cancel
Attach policies

Attach permission policies to Op-Team

▸ Current permissions policies (0)

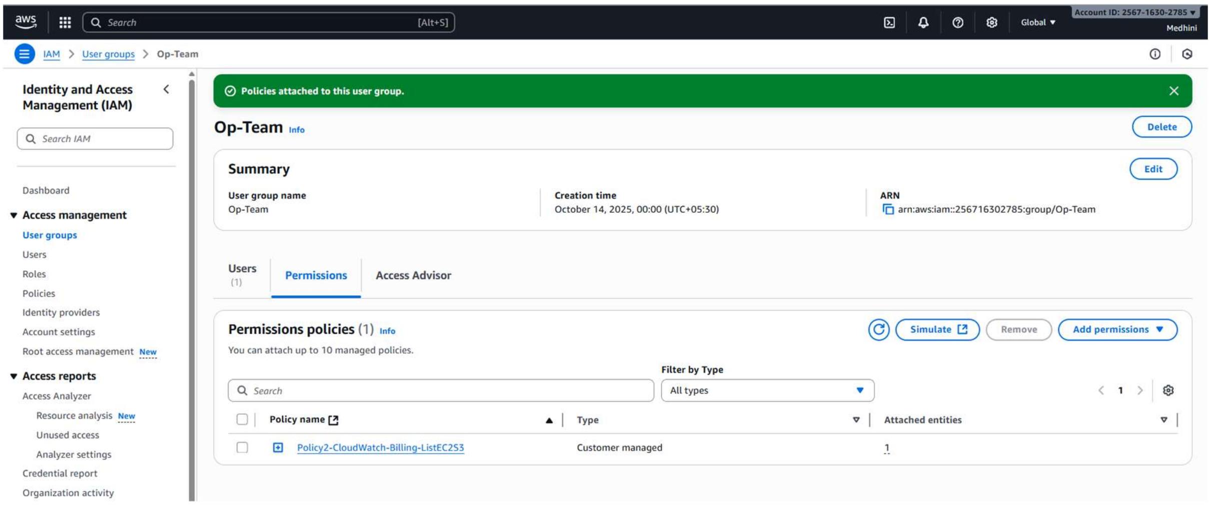
Other permission policies (1/1078)

You can attach up to 10 managed policies to this user group. All of the users in this group inherit the attached permissions.

Search: Policy2 — Filter by Type: All types — 1 match

☑	Policy name	Type	Used as	Description
☑	Policy2-CloudWatch-Billing-ListEC2S3	Customer managed	None	Allows full access to CloudWatch and

Step 3: Now Policy Number 2 is attached to **“Op-Team”** as shown in below picture



Policies attached to this user group.

Op-Team Info

Summary

User group name: Op-Team

Creation time: October 14, 2025, 00:00 (UTC+05:30)

ARN: arn:aws:iam::256716302785:group/Op-Team

Users (1) | Permissions | Access Advisor

Permissions policies (1) Info

You can attach up to 10 managed policies.

Search — Filter by Type: All types

□	Policy name	Type	Attached entities
□	Policy2-CloudWatch-Billing-ListEC2S3	Customer managed	1