

# Managing Permissions with AWS IAM

Sristi Acharya



Sristiacharya13



[linkedin.com/in/sristiacharya](https://www.linkedin.com/in/sristiacharya)

# WHAT IS AWS IAM?

What it does:

It is a web service that helps you securely control access to Aws resources.

Why it's useful:

It helps to manage and grant access to resources as a form of providing protection.

How I'm using it in this project:

To create instances, manage users and groups by assigning permission to be able to access resources by allowing and denying them.

# SETTING UP TAGS

- I have set up two EC2 instances to test the effectiveness of the permission settings I will set up in AWS IAM. I have used tags to label them.
- Tags are like labels assigned to Aws resources for an organization to help to identify and manage resources in a more efficient manner. Tags are useful for cost allocation and budgeting, resources organizations and setting security policies.
- The tag have used on my EC2 instances is called Env. The value I have assigned for my instances are Production and development.

i-04efd77d13b057c7a (project-production-sristi)

Details

Status and alarms

Monitoring

Security

Networking

Storage

Tags

Tags

Q

Key	Value
Name	project-production-sristi
Env	production

i-060c654a0afa863e6 (project-development-sristi)

Details

Status and alarms

Monitoring

Security

Networking

Storage

Tags

Tags

Q

Key	Value
Env	development
Name	project-development-sristi

# IAM POLICIES

- IAM Policies are set of rules that helps to allow/deny users or resources access or permission to perform a particular actions.
- For this project, I have set up a policy using JSON editor.
- I have created a Policy that will allow permission to EC2 instances that contains a tags **Env** and a value of **development** in the JSON policy file.

When writing JSON Policy statements, we have to specify the

**Effect:** This will allow/deny for any action. Action: This are things that you can do on a EC2 instances by allowing/deny based on the policy.

**Resource:** refer to the AWS entities on which actions specified in the policy are allowed or denied.

## Policy editor

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "ec2:*",
7       "Resource": "*",
8       "Condition": {
9         "StringEquals": {
10           "ec2:ResourceTag/Env": "development"
11         }
12       }
13     },
14     {
15       "Effect": "Allow",
16       "Action": "ec2:Describe*",
17       "Resource": "*"
18     },
19     {
20       "Effect": "Deny",
21       "Action": [
22         "ec2:DeleteTags",
23         "ec2:CreateTags"
24       ],
25       "Resource": "*"
26     }
27   ]
28 }
```

# AWS ACCOUNT ALIAS

- When new users get onboarded onto my AWS account, they get access by signing into a unique URL created for my account's Account ID.
- An account alias is a friendly name for your AWS account you can use to sign into AWS management console instead of your account.

Create alias for AWS account 738565846221


Preferred alias

project-alias-sristi

Must be not more than 63 characters. Valid characters are a-z, 0-9, and - (hyphen).

New sign-in URL

https://project-alias-sristi.signin.aws.amazon.com/console

 IAM users will still be able to use the default URL containing the AWS account ID.

Cancel

Create alias

- Now, my new AWS console sign-in URL is <https://project-alias-sristi.signin.aws.amazon.com/console>

# IAM USERS + USER GROUPS

- IAM Users are entities created and granted access to Aws resources and managed by an Administrator of that account/resources.
- I also created a User Group. User Groups are useful for managing and granting access to user a in a particular group.
- My User Group is called dev-sristi, I attached the Policy I created to this User Group, which means all users in that group will be granted the access to that resources based on the group permissions set.


**User details**

User name

dev-sristi

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 applications.

☒ I want to create an IAM user

We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

- When I created a new User, I had to tick a checkbox that will grant users to access the Aws management console.
- Once my new user was set up, there were two ways I could share its sign-in details:  
Firstly, by emailing the user with the credentials. Secondly, by downloading the csv.file that contains user credentials.
- My new user had a unique sign-in URL - <https://project-alias-sristi.signin.aws.amazon.com/console>

## 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)

Search

<input checked="" type="checkbox"/>	Group name	Users	Attached policies	Created
<input checked="" type="checkbox"/>	dev-group	0	SristiDevEnvironmentPolicy	2024-07-13 (8 minutes ago)

► Set permissions boundary - optional

Cancel

Previous

Next

## My user's sign-in credentials

### 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

IAM > Users > Create 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

Email sign-in instructions

Console sign-in URL

https://project-alias-sristi.signin.aws.amazon.com/console

User name

dev-sristi

Console password

\*\*\*\*\* Show

Cancel

Download .csv file

Return to users list

# IAM USER IN ACTION

- Now with my IAM Policy, IAM User Group and IAM User all set up, I logged into my AWS account as the new user.
- To log in as my IAM User, I used the URL to login in order to get access to the Aws management console after creating the user.



## Sign in as IAM user

Account ID (12 digits) or account alias

project-alias-sristi

IAM user name

dev-sristi

Password

.....

☐ Remember this account

Sign in

[Sign in using root user email](#)

[Forgot password?](#)



Once I logged in, I saw so may access denied from different panels on the dashboard. It looks different from then AWS Management Console.

Applications (0)

Info

Region: Europe (Stockholm)

Create application

eu-north-1 (Current Region) ▼

Find applications

<

1

>

Name ▲	Description ▼	Region ▼	Originating account
--------	---------------	----------	---------------------

⊗

▶ Access denied

Security

Info

Region: Europe (Stockholm)

⊗

▶ Access denied

Cost and usage

Info

Current month costs

⊗

Access denied

Cost breakdown

⊗

Access denied

Forecasted month end costs

⊗

Access denied

Savings opportunities

⊗

Access denied

# IAM POLICIES IN ACTION

I tested the JSON IAM policy I set up by stopping the development and production instances by triggering an alert. When I tried to stop the production instance, i had a red popup notice/alert. The user did not have the right permission to stop the instance from running.

A red fail banner pops up if I stop the production instance

## ⊗ Failed to stop the instance i-04efd77d13b057c7a

You are not authorized to perform this operation. User: arn:aws:iam::738565846221:user/dev-sristi is not authorized to perform: ec2:StopInstances on resource: arn:aws:ec2:us-west-2:738565846221:instance/i-04efd77d13b057c7a because no identity-based policy allows the ec2:StopInstances action. Encoded authorization failure message: dsE7vAxLnVNdS2QVBRit4KULai9P7HCFts3GMUbkmQxI5\_CsGY4-fMgRZ8l2gu1h\_szhavz71ZV1mO1\_n\_fy5CgCCRbRjRz-nrN-wAXTqcCXqC0iXTn9GZsMt4POdG0mpR2NG1cSXCNOGWD2VLVdCuXz9SOTtR8mBn2EG\_1dVMHD7kCh3xjWUYBzpYogMq2dnPeeElF0OMK6RC5ULrBJQ9FMGexIX3Wrvvd0tA-rQ8UiLaLf1MRKI5u3xokbdsta2rphHQ\_jZG8zkOxOed5bVVVMNYkWoykStGKGTU-5XeQWshj8qh27wOdRsroTG2VerTkc-K-PeX-TDGALDzORvFEob\_xK5vNSHLcR27O1F\_T5BFobSeYRf7MvjUr2EcDj-tAG5VAnk\_IDjVsAZpu2GHEAwSpOzJ\_1t8GJGABIGjJTz\_Uf6exOZo5lRsm1tD9J8HNO9UG2bAQQ8rVo-rPSEGLVQ-pWM0UXbVC8dfd2S9q8qNwMijKexJ2GydrYXYWGkuRPg\_nbNttZp83lFN06vufKeuxGDF7qBkZ5oq4yLsOILYidUixxboISQPelazG06wH-xc9e5Gufim-9VvCpOwUsHKLsTK49EWKLO-goq7xR58UOlrsuVXyud1bpFGwp4Lh6LSvZ3Pvx3oYg1prTJSXOsPB4gov7IUV3T8hPaIHlvMuAN3k9HLLkfkCpTkKL-KGJhaR55JCNfkDmhiALYu8NO35X6lO5FfKGR\_LnJLnVjvVrDirIAHlo8ikwShZERmKrtxvhsxKgHyvTTItzesQizi5tZCVLaoT17-YJnaWhWsgsldLS8DrsTzHvLOH\_ba5N0

Next, when I tried to stop the development instance, I had a green popup alert “**Successfully initiated stopped**”. The user was able to stopped the instance from running because of the group policy set which the user was part of.

A green success banner pops up if I stop the development instance

✔ Successfully initiated stopping of i-0988b5a1ec3bc8b02

Instances (1/2) Info

Find Instance by attribute or tag (case-sensitive)

All states ▼

[-]

Name ✎ ▼

Instance ID

Instance state ▼

Instance type ▼

Status check

Instances (1/2) Info

Find Instance by attribute or tag (case-sensitive)

All states ▼

< 1 > ⚙

[-]

Name ✎ ▼

Instance ID

Instance state ▼

Instance type ▼

Status check

Alarm status

Availability Zone ▼

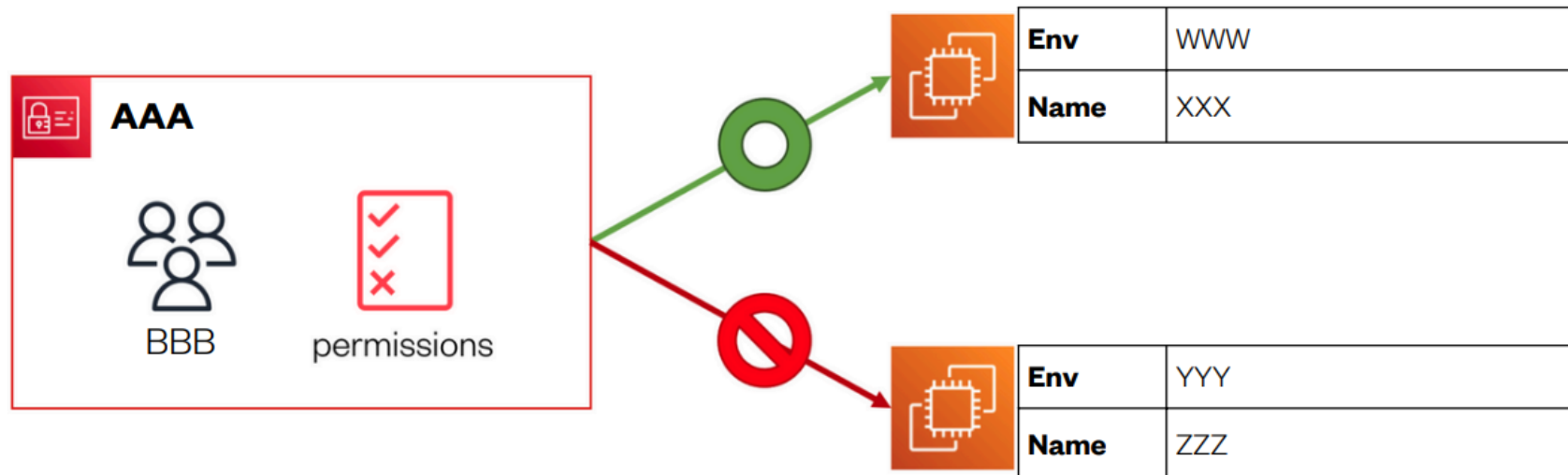
Public IPv4

<input checked="" type="checkbox"/>	project-produ...	i-04efd77d13b057c7a	✔ Running 🔍 🔍	t2.micro	✔ 2/2 checks passed	⊗ User: arn:aws:i	us-west-2a	ec2-35-93-
<input type="checkbox"/>	project-develo...	i-060c654a0afa863e6	⊖ Stopped 🔍 🔍	t2.micro	-	⊗ User: arn:aws:i	us-west-2a	-

# TO SUMMARISE

I created:

- An IAM User Group called **dev-group** with defined permissions using an IAM Policy.
- An IAM User called **dev-sristi** is added to the user group
- An EC2 instance with the Env tag development and Name **project-development-sristi**
- An EC2 instance with the Env tag production and Name **project-production-sristi**



# My Key Learnings

- What are IAM Policies are set of rules helps to allow and deny users certain permissions to a resources.
- What is an AWS Account Alias are friendly names created that can be used to access Aws management console instead of using a actual account to login.
- What are IAM Users are entities created and granted permission to perform or access certain resources by a policy. It helps to create and manage by allowing and deny based and rules .
- What are IAM UserGroups is a folder or container that users are managed by setting a group policy which allows or deny what those users under that group can have access to what resources. It is easy to manage a lot of users by just setting a particular policy and all users get affected.
- I have learnt how to create instances ,assigned policies to users and groups, how to stop instances with certain users or group users based on set policy and permission given. Also how to set a JSON policy file.