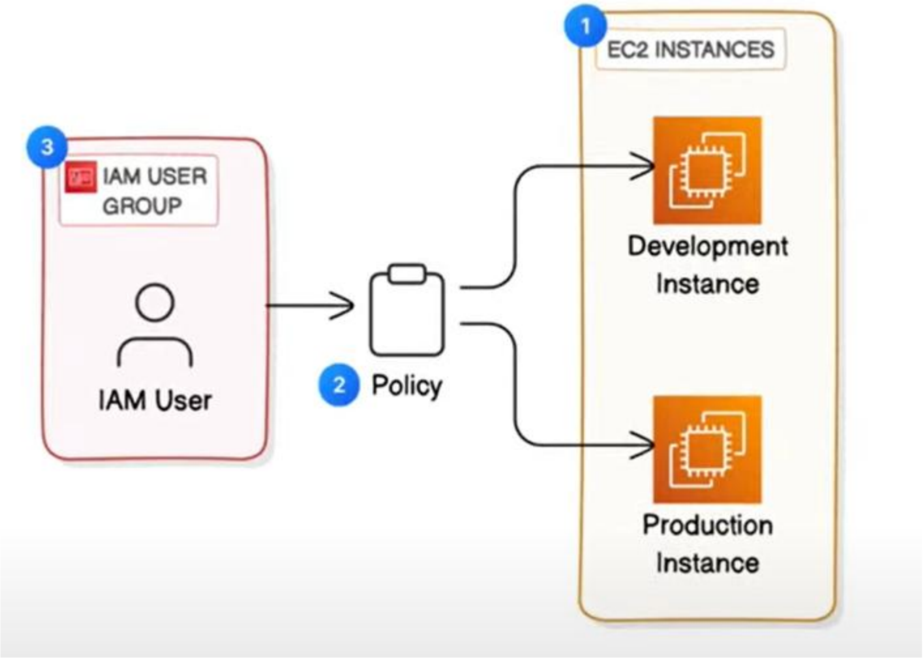
Cloud Security with AWS IAM:

We'll be using the AWS Identity and Access Management (IAM) tool to control who is authenticated (signed in) and authorized (has permissions) in your AWS console.

We'll launch an EC2 instance, then control who has access to it by creating some IAM policies and user groups. It will look something like this:



Steps:

Step #1: Launch EC2 Instances

Step #2: Create an IAM Policy (it decided who can do what like who can read, edit & delete resources)

Step #3: Create an AWS Account Alias

Step #4: Create IAM Users and User

Step #5: Test your intern's access

Secret Mission: IAM Policy Simulator

All Done: Nice work!

Before You Go: Delete Your Resources

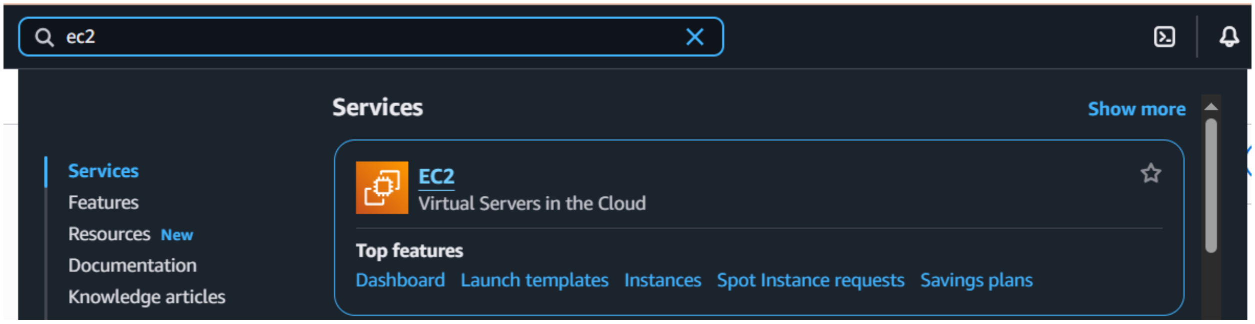
We will be using two EC2 instances (1 for the production environment & 1 for the development environment)

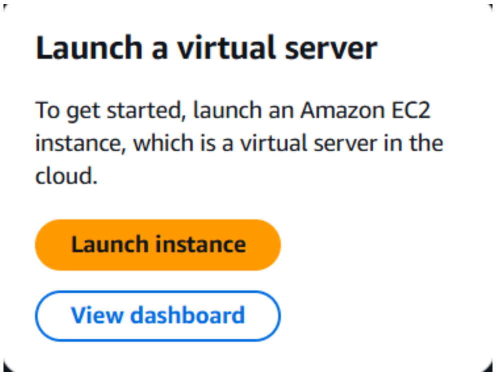
The **development** environment is where developers write, test, and debug code before it's deployed to **production**, which is the live environment that your end users can use.

- **Development Environment** is where developers codes and make applications.
- **Production Environment** is where we see the outcome of code and interact with.

Launching an EC2 Instance:

- Log in to your AWS Management Console.
- Open your EC2 console - search for it at the search bar.

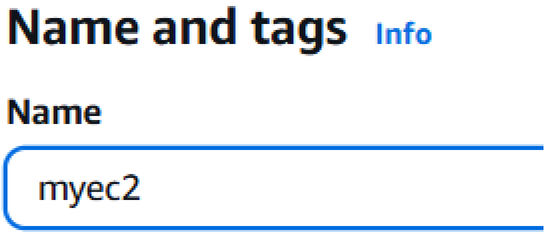



Launch a virtual server

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

- Click on Launch instance.
- Give it a name like this:

You can use tags. Tag is just a label. Tagging helps us with identifying all resources with the same tag at once (they are useful filters when you're searching for something), cost allocation, and applying policies based on environment types. It is useful when you have multiple instances and you want to apply policies on instances which have same tag. You can make a group of them and apply policies.

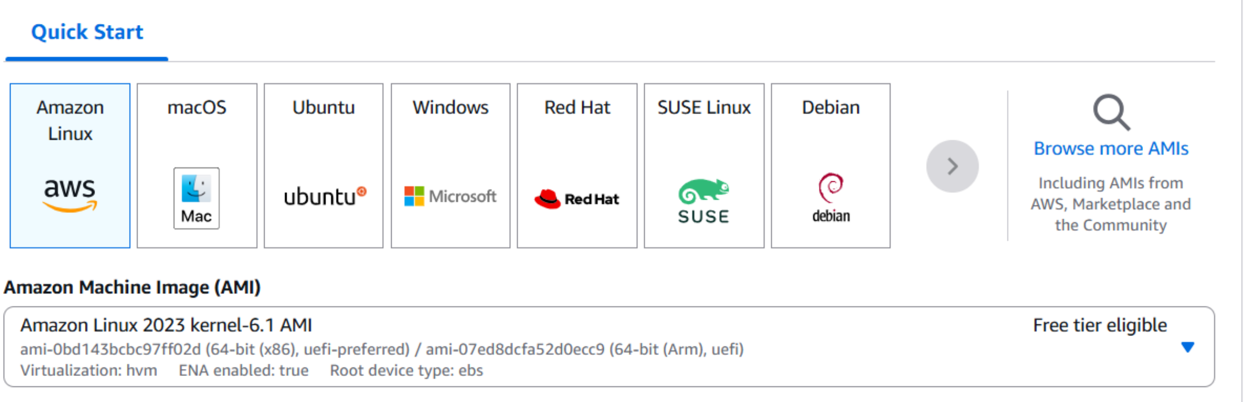


Name and tags Info

Name

myec2

- Choose the image (it is basically operating system which you are going to use in your virtual machine)

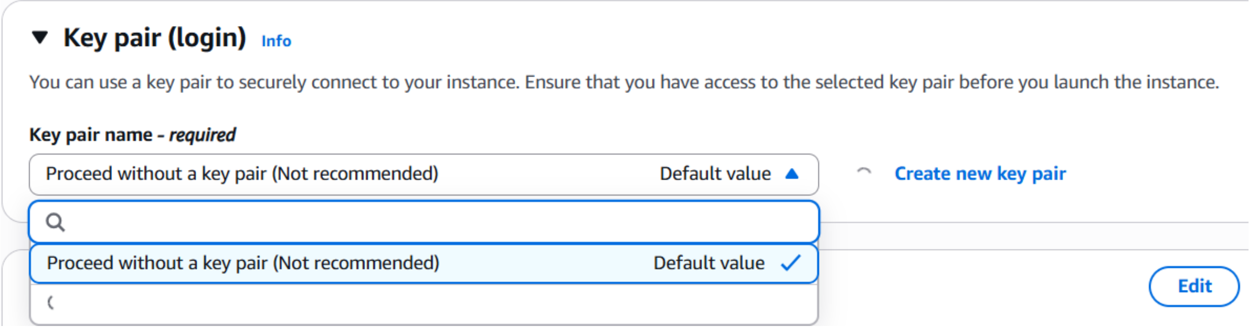


Amazon Machine Image (AMI)

Amazon Linux 2023 kernel-6.1 AMI — Free tier eligible
ami-0bd143bcbc97ff02d (64-bit (x86), uefi-preferred) / ami-07ed8dcfa52d0ecc9 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

- We can proceed without a key pair.

Key pairs allow you to access your EC2 instances without going through console.



▼ Key pair (login) Info

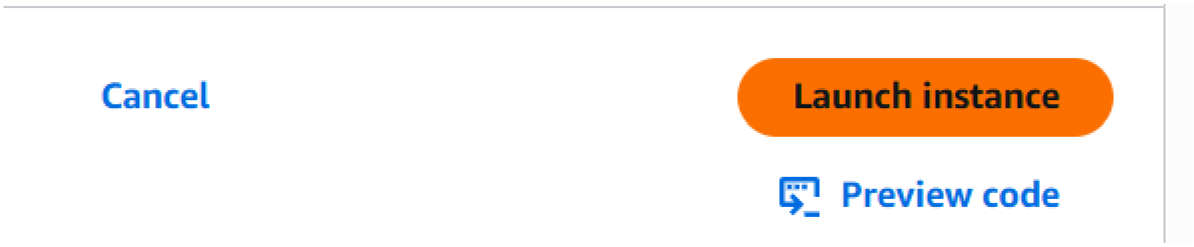
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

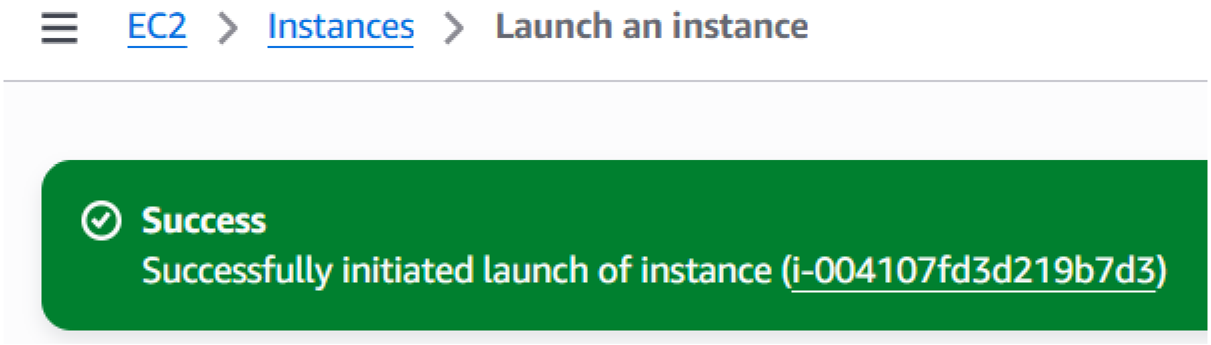
Proceed without a key pair (Not recommended) Default value ▲

Proceed without a key pair (Not recommended) Default value ✓

- Scroll down with default settings and click on Launch instance



- Our EC2 instance will be created.

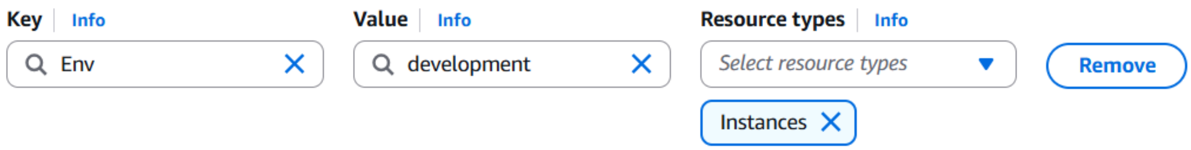


EC2 > Instances > Launch an instance

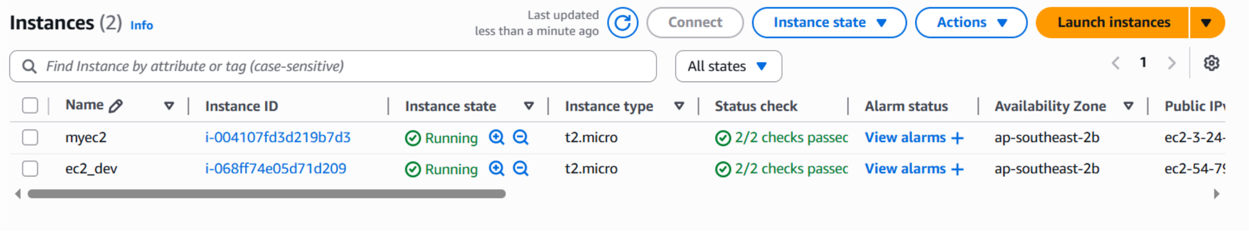
Success
Successfully initiated launch of instance (i-004107fd3d219b7d3)

- Create one more EC2 instance for the **development environment**.

Repeat the same flow, but this time using different tags



Key: Env
Value: development
Resource types: Instances



Instances (2)

Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
1

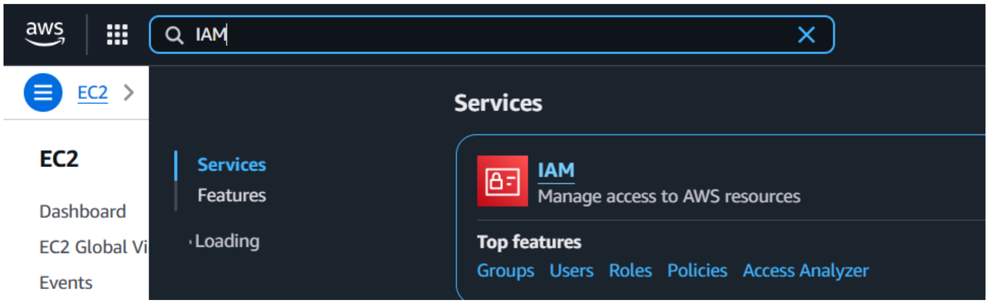
□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP\
□	myec2	i-004107fd3d219b7d3	Running	t2.micro	2/2 checks passec	View alarms +	ap-southeast-2b	ec2-3-24-
□	ec2_dev	i-068ff74e05d71d209	Running	t2.micro	2/2 checks passec	View alarms +	ap-southeast-2b	ec2-54-7!

IAM Policy:

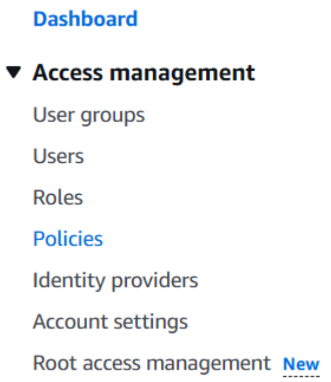
We'll set up some permission policies. We will set up a permission to access development EC2 instance but not the production instance.

IAM helps you managing access to your resources on the cloud.

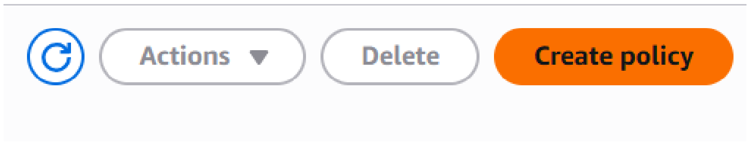
- Go to the IAM console



- On the left side, click on the Policies

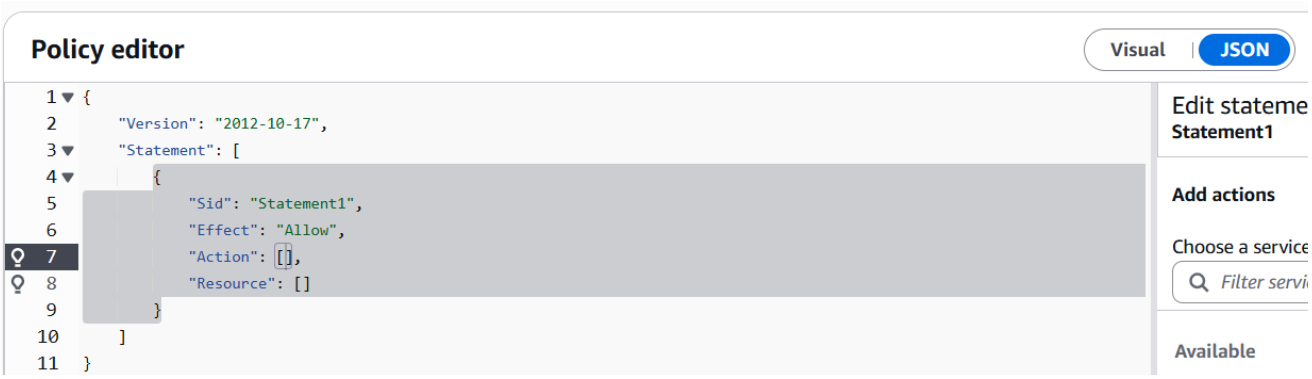


- Click on Create policy



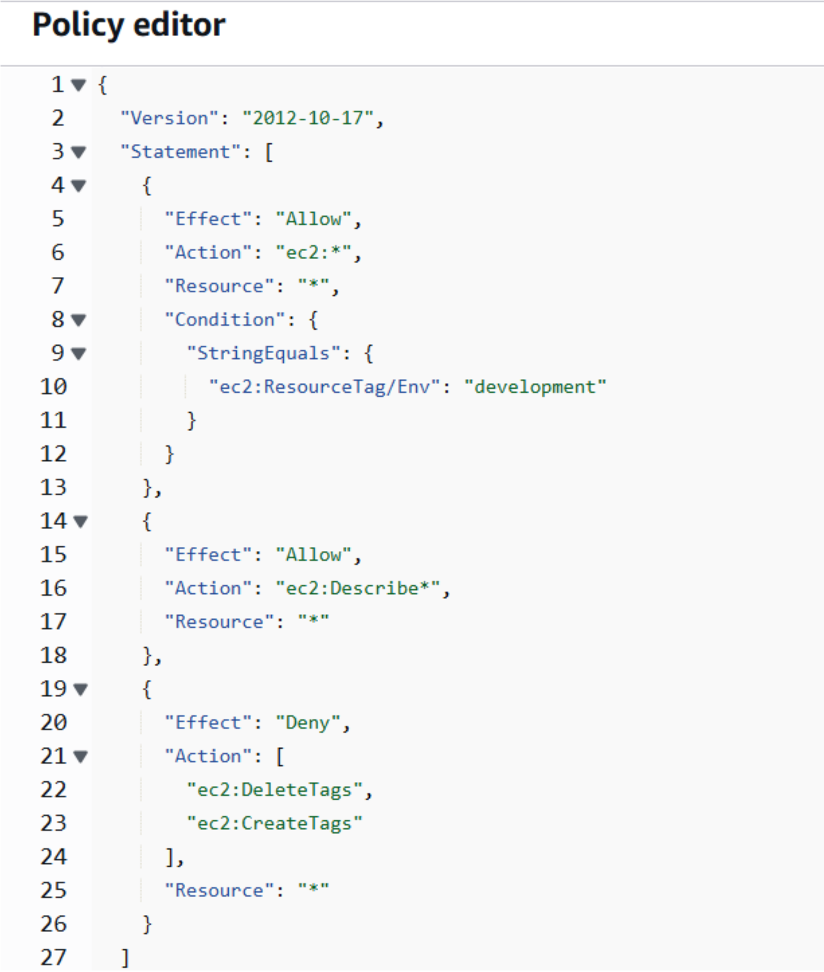
- Switch your Policy editor tab to JSON.

It is because we can create and edit AWS policies in the visual editor or JSON.



```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Action": [],
            "Resource": []
        }
    ]
}
```

- You can paste your policy in the policy editor.

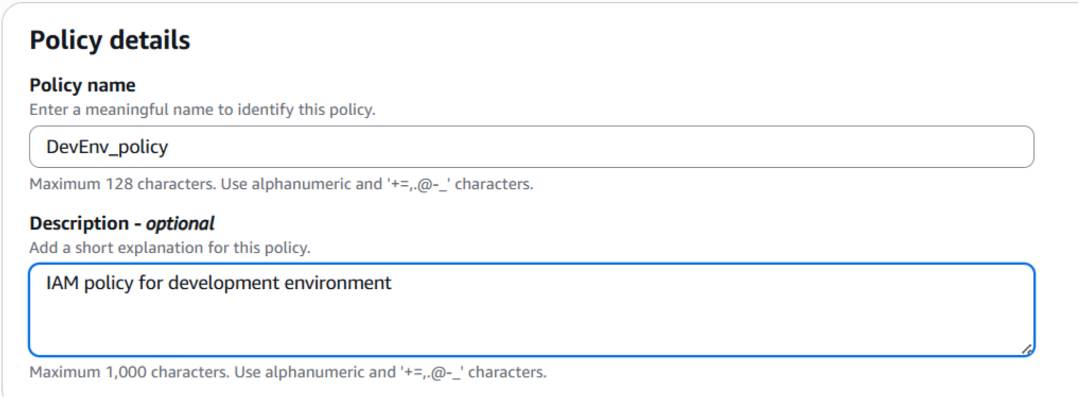


Policy editor

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Env": "development"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": [
        "ec2:DeleteTags",
        "ec2:CreateTags"
      ],
      "Resource": "*"
    }
  ]
```

This policy states that one can perform any action to EC2 instances which are in the development environment but he does not have permission to do anything outside development because this policy only allows to perform action to EC2 which are in the development environment. It doesn't say anything about production environment.

- Click Next and write the name of your policy.



Policy details

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

Description - *optional*
Add a short explanation for this policy.
IAM policy for development environment
Maximum 1,000 characters. Use alphanumeric and '+=,.@-_' characters.

AWS Account Alias:

An Account Alias is a friendly name for your AWS account that you can use instead of your account ID (which is usually a bunch of digits) to sign in to the AWS Management Console.

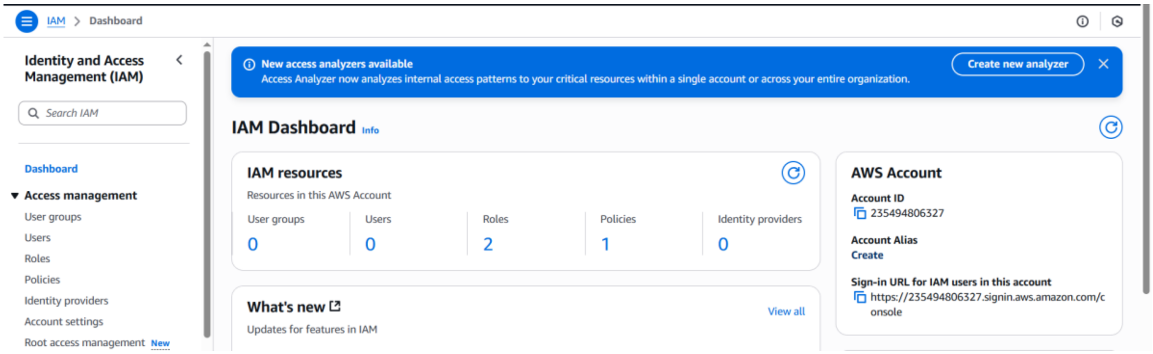
Your AWS account's sign-in page has this URL by default:
https://**Your_Account_ID**.signin.aws.amazon.com/console/

If you create an AWS account alias for your AWS account ID, your sign-in page URL looks more like: https://**Your_Account_Alias**.signin.aws.amazon.com/console/

You would create an alias to make it easier to remember and share your AWS console's login URL with others

Procedure:

- Still in your IAM console, select Dashboard from the left hand navigation panel.
- In the right-hand side of the dashboard, choose **Create** under **Account Alias**.



New access analyzers available
Access Analyzer now analyzes internal access patterns to your critical resources within a single account or across your entire organization.

IAM Dashboard Info

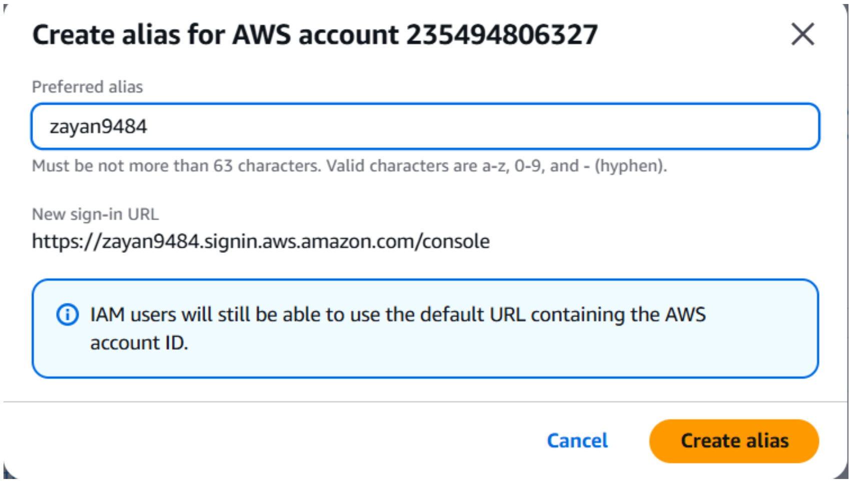
IAM resources
Resources in this AWS Account

User groups	Users	Roles	Policies	Identity providers
0	0	2	1	0

What's new
Updates for features in IAM

AWS Account
Account ID
235494806327
Account Alias
Create
Sign-in URL for IAM users in this account
https://235494806327.signin.aws.amazon.com/console

- Click on Create and write the name



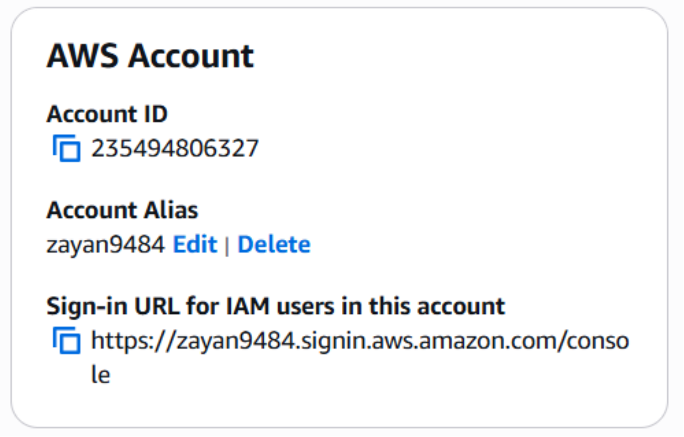
Create alias for AWS account 235494806327

Preferred alias
zayan9484
Must be not more than 63 characters. Valid characters are a-z, 0-9, and - (hyphen).

New sign-in URL
https://zayan9484.signin.aws.amazon.com/console

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

- Alias will be created.



AWS Account

Account ID

235494806327

Account Alias

zayan9484 Edit | Delete

Sign-in URL for IAM users in this account

https://zayan9484.signin.aws.amazon.com/console

Creating IAM User:

In this step, we will:

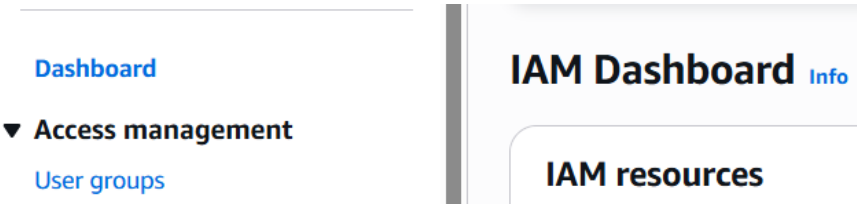
Set up a dedicated **IAM group** for all users, so we can manage all users permissions from one place. Set up a dedicated IAM **user**, so our user had a way to log in.

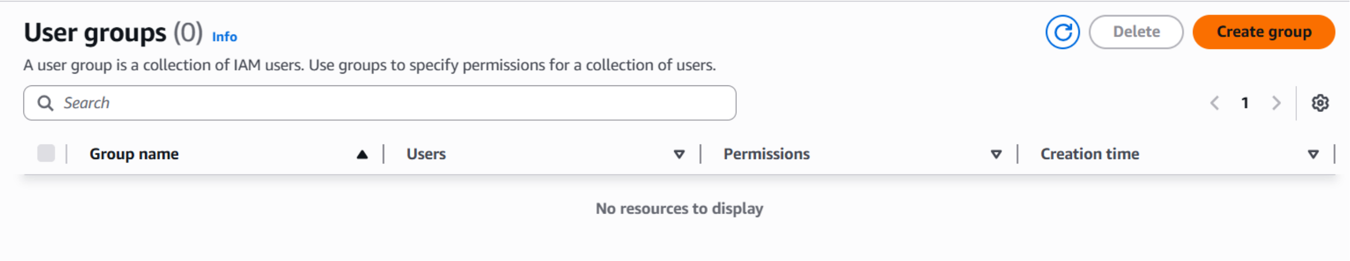
Procedure:

- Choose **User groups** in your left-hand navigation panel.

  An IAM user group is a collection/folder of IAM users. It allows you to manage permissions for all the users in your group at the same time by attaching policies to the group rather than individual users.

- Choose **Create group**.





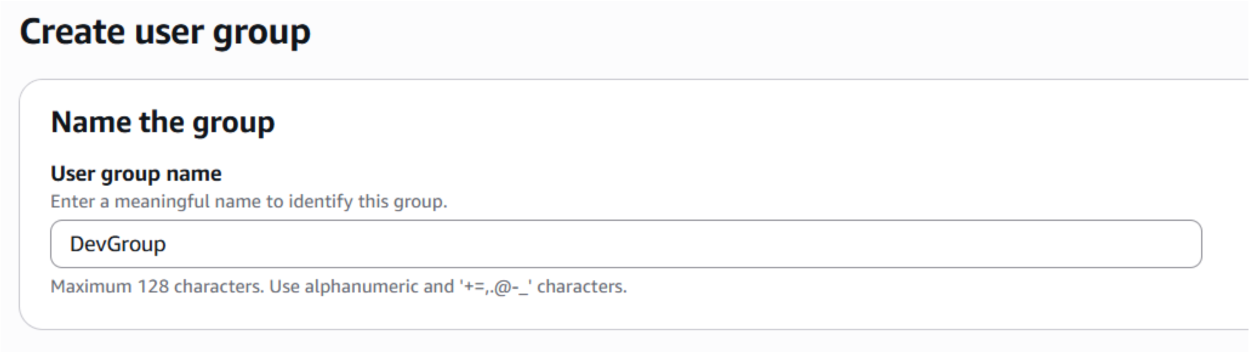
A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

Group name	Users	Permissions	Creation time

No resources to display

To set up your user group:

- Write the name of the group
- Attach permission policies (the one we created)



Create user group

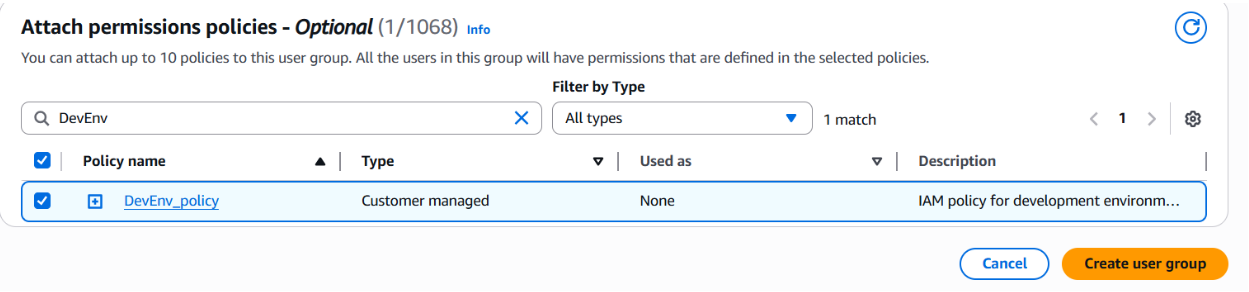
Name the group

User group name

Enter a meaningful name to identify this group.

DevGroup

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



Attach permissions policies - *Optional* (1/1068) 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.

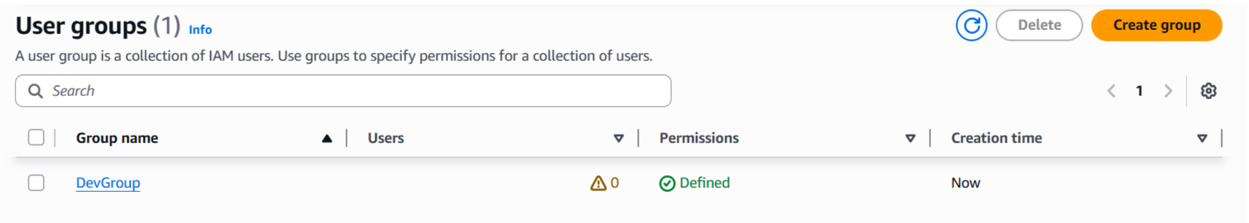
Filter by Type

DevEnv

All types

1 match

☑	Policy name ▲	Type ▽	Used as ▽	Description
☑	⊞ DevEnv_policy	Customer managed	None	IAM policy for development environm…



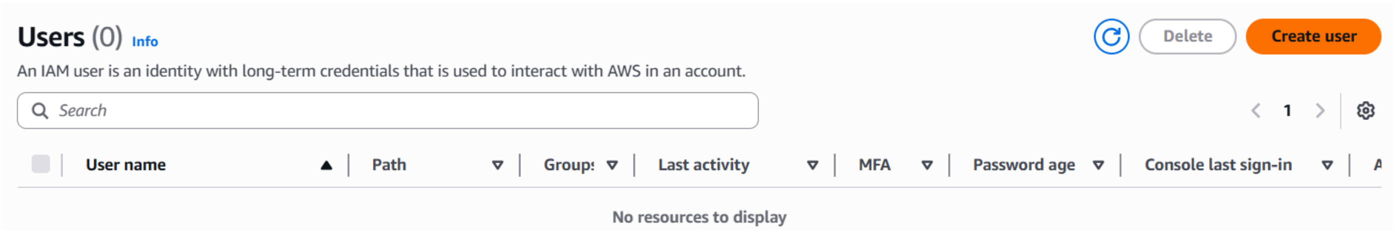
User groups (1) Info

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 ▽
□	DevGroup	⚠ 0	✓ Defined	Now

Now add the Users to the User Groups:

- Choose **Users** from the left-hand navigation panel.
- Choose **Create user**.

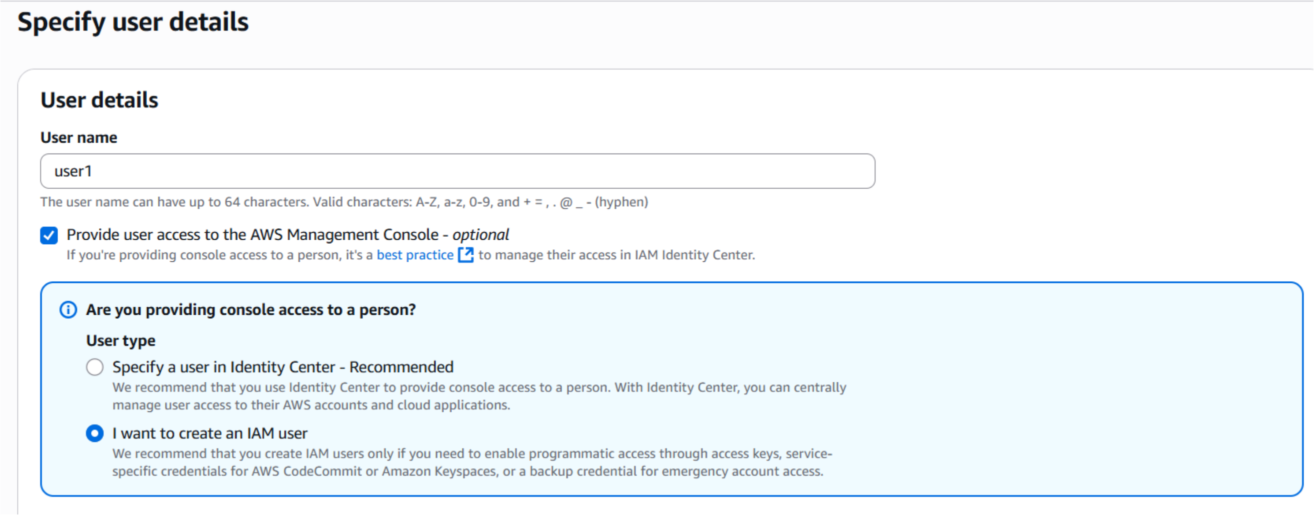


Users (0) Info

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 ▽	A

No resources to display



Specify user details

User details

User name

user1

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?

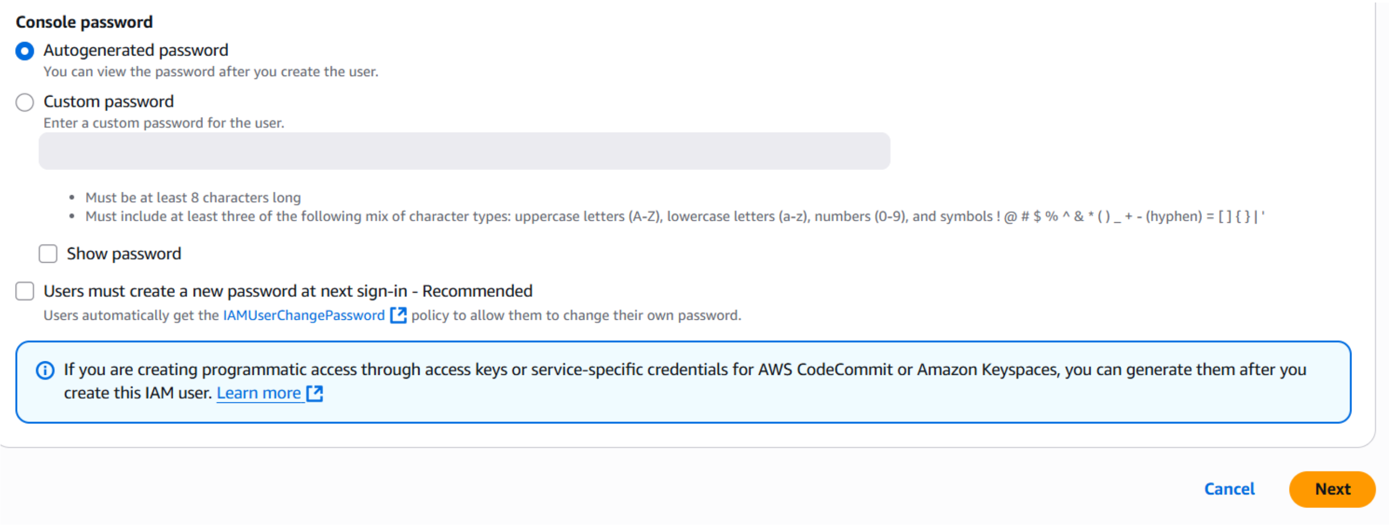
User type

○ 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.



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.

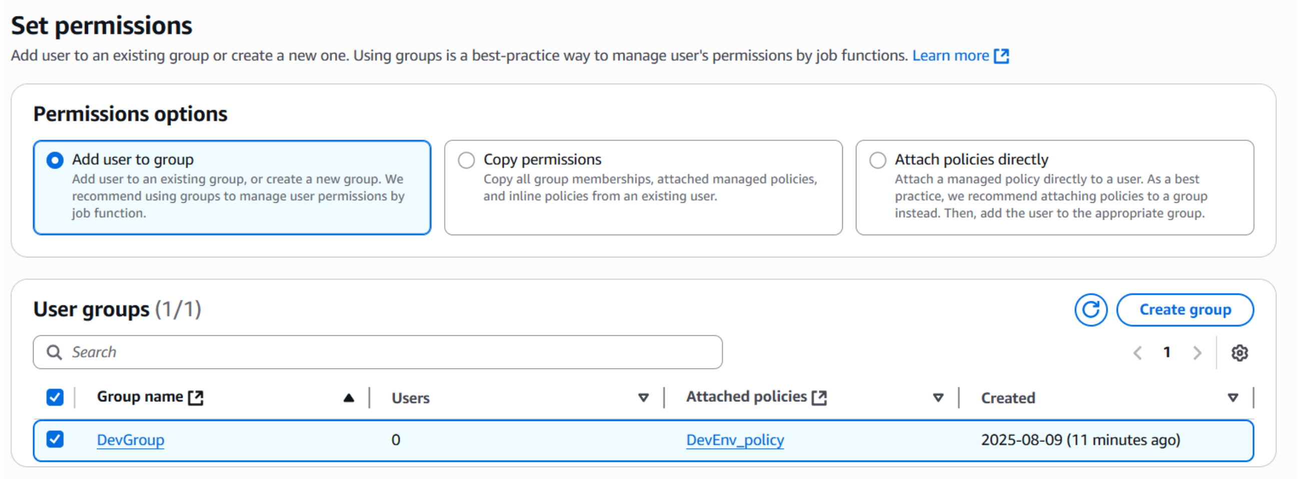
If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more

- Tick the checkbox for **Provide user access to the AWS Management Console**.

Why are we doing this?
If you don't tick this box, your new user won't get to sign in and access AWS services through the Console. They'll have to access AWS services through other, more advanced methods **e.g., AWS CLI**

You can check or uncheck the **Users must create a new password at next sign-in** box. I'm not checking this because it will then take extra time to change password.

- Click **Next**
- Select the checkbox of the User Group name **i.e., DevGroup** (in my case) to add the user into that group



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

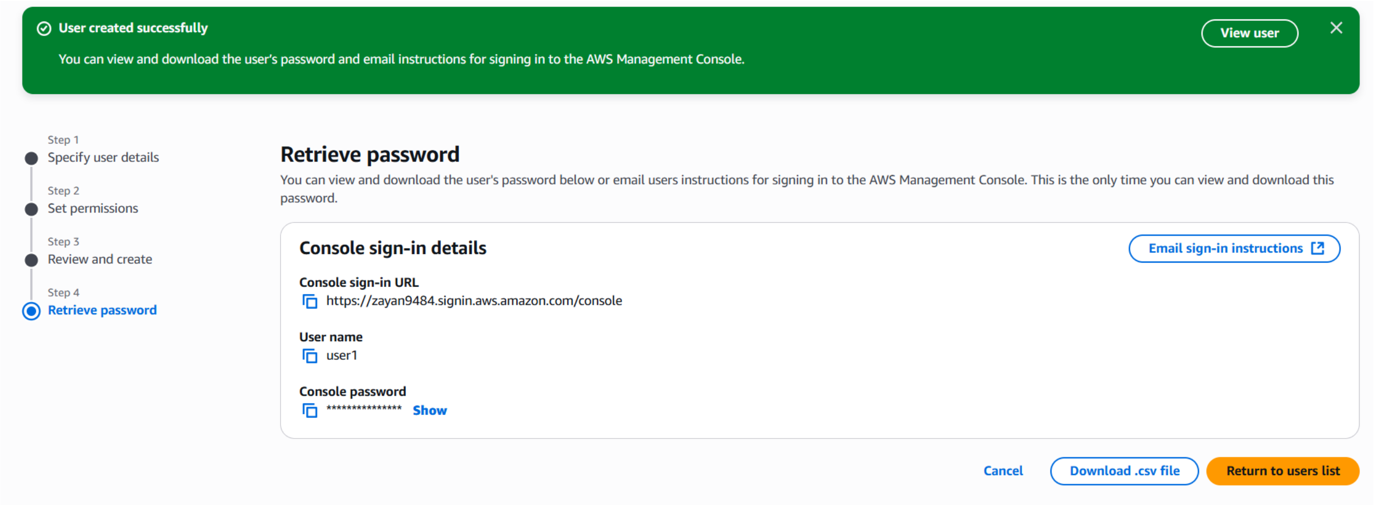
Permissions options

- ● 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
☑	DevGroup	0	DevEnv_policy	2025-08-09 (11 minutes ago)

- Click on **Next** and then **User** will be created



User created successfully

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

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://zayan9484.signin.aws.amazon.com/console

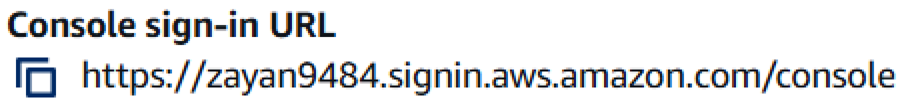
User name: user1

Console password: *************** Show

We can download the .csv file which contains the login credentials of the user.

Now let's test the User1 IAM User's access first. That way we can make sure that the user have the right access to our development instance (and not the production instance).

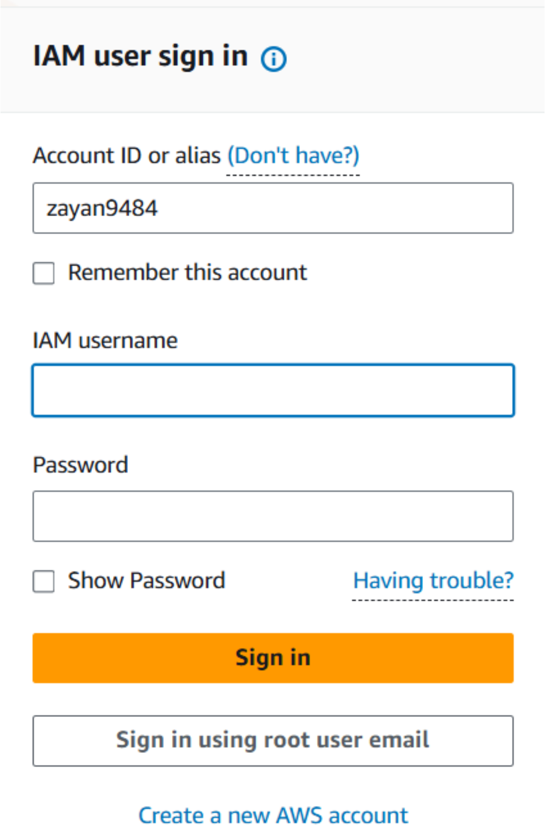
- Copy the **sign-in** URL and paste it in the browser



Console sign-in URL

https://zayan9484.signin.aws.amazon.com/console

We can see our alias already written here. Use the credentials which we've downloaded to login



IAM user sign in

Account ID or alias (Don't have?)

zayan9484

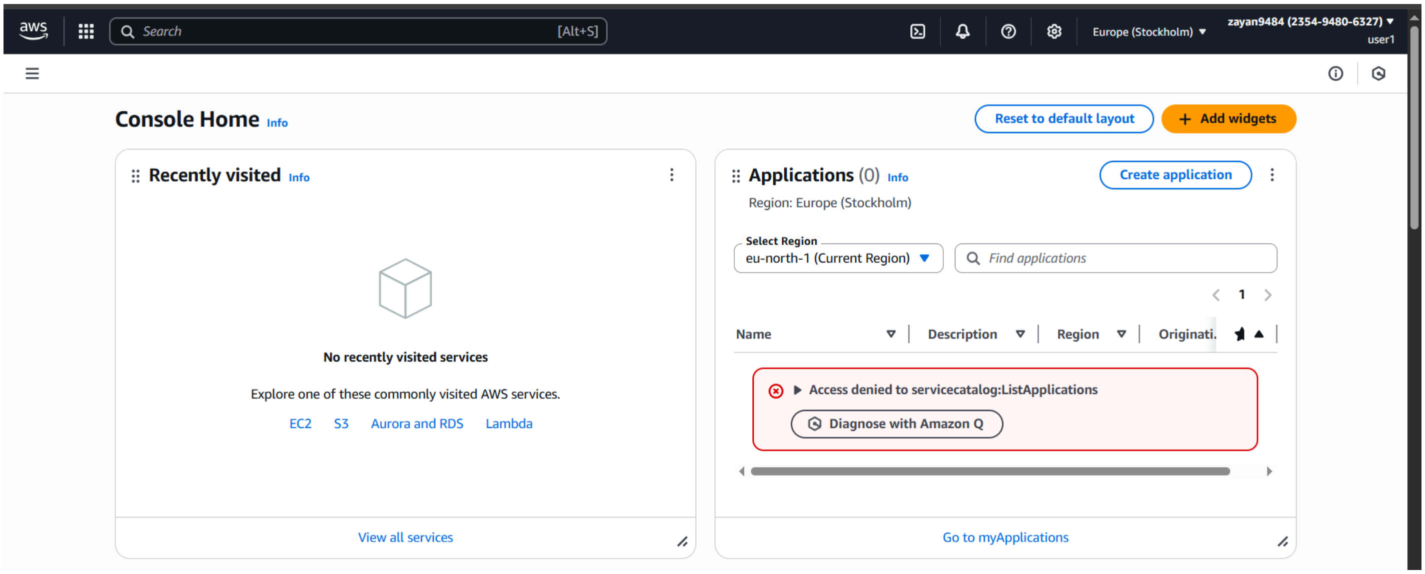
□ Remember this account

IAM username

Password

□ Show Password

- We can see on top right corner our alias name and we cannot access some features or things because of the policy we've created



zayan9484 (2354-9480-6327)
user1

Europe (Stockholm)

Console Home

Recently visited

No recently visited services

Explore one of these commonly visited AWS services.

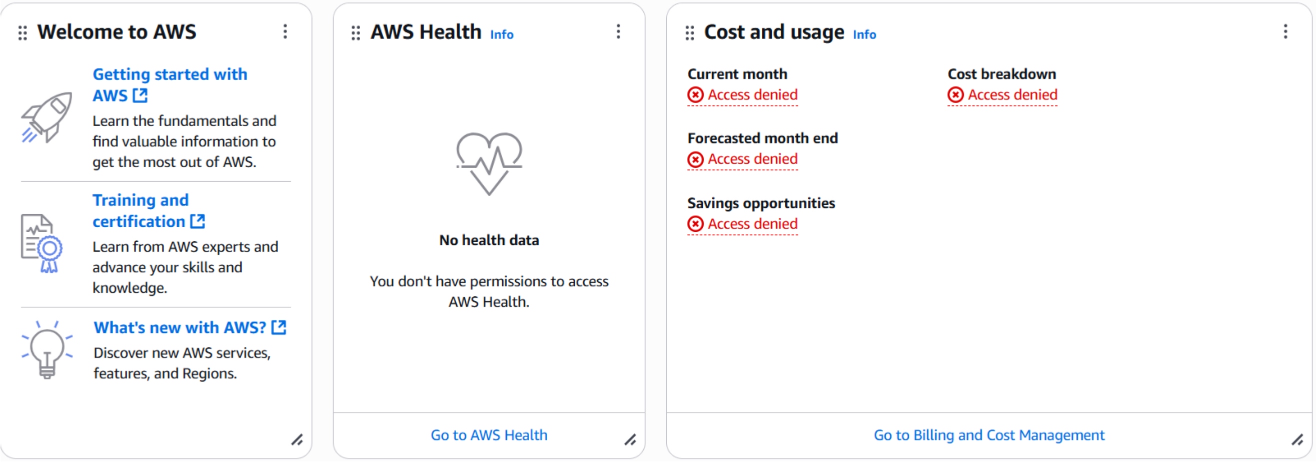
Applications (0)

Region: Europe (Stockholm)

Select Region: eu-north-1 (Current Region)

Name	Description	Region	Originati.

Access denied to servicecatalog:ListApplications



Welcome to AWS

Getting started with AWS
Learn the fundamentals and find valuable information to get the most out of AWS.

Training and certification
Learn from AWS experts and advance your skills and knowledge.

What's new with AWS?
Discover new AWS services, features, and Regions.

AWS Health Info

No health data

You don't have permissions to access AWS Health.

Go to AWS Health

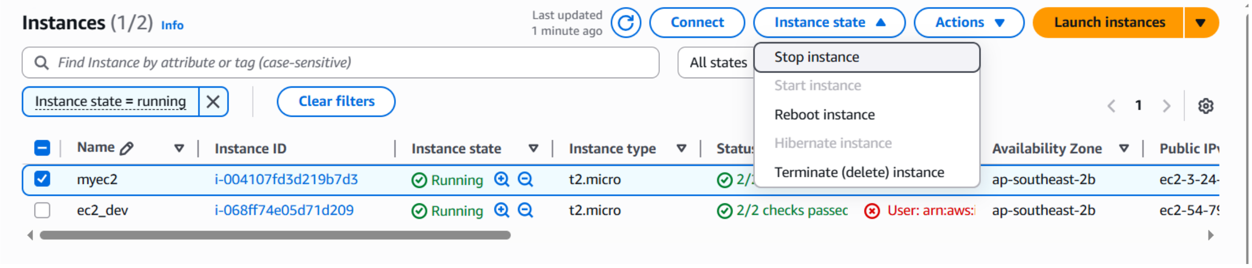
Cost and usage Info

Current month Access denied	**Cost breakdown** Access denied
Forecasted month end Access denied	
Savings opportunities Access denied	

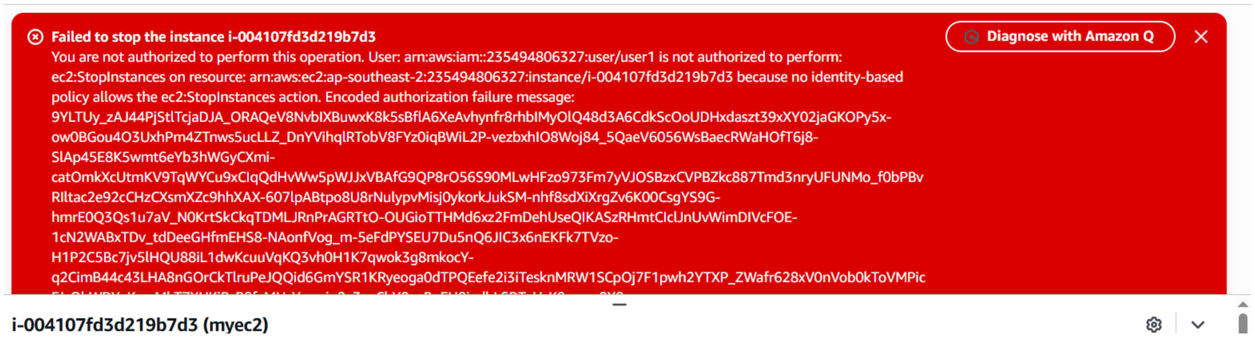
Go to Billing and Cost Management

- Head to your **EC2** console, and make sure you're in the same **Region** as the one where you deployed your two production and development instances.
- Head to **Instances**.
- Select your **production** instance, and in the **Instance state** dropdown, select **Stop instance**.

Let’s see if we can stop our production instance.



	Name	Instance ID	Instance state	Instance type	Status		Availability Zone	Public IP
☑	myec2	i-004107fd3d219b7d3	Running	t2.micro	2/2		ap-southeast-2b	ec2-3-24-
□	ec2_dev	i-068ff74e05d71d209	Running	t2.micro	2/2 checks passec	User: arn:aws:	ap-southeast-2b	ec2-54-7!



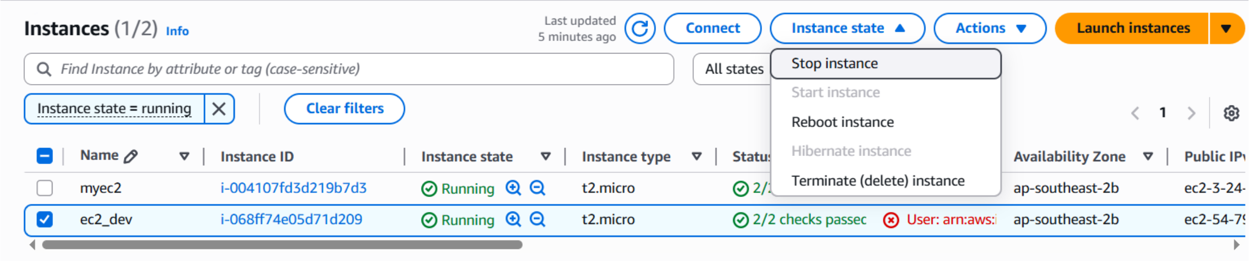
Failed to stop the instance i-004107fd3d219b7d3

```
You are not authorized to perform this operation. User: arn:aws:iam::235494806327:user/user1 is not authorized to perform:
ec2:StopInstances on resource: arn:aws:ec2:ap-southeast-2:235494806327:instance/i-004107fd3d219b7d3 because no identity-based
policy allows the ec2:StopInstances action. Encoded authorization failure message:
9YLTUy_zAJ44PjStlTcjaDJA_ORAQeV8NvbIXBuwxK8k5sBflA6XeAvhynfr8rhbIMyOlQ48d3A6CdkScOoUDHxdaszt39xXY02jaGKOPy5x-
ow0BGou4O3UxhPm4ZTnws5ucLLZ_DnYVihqlRTobV8FYz0iqBWiL2P-vezbxhIO8Woj84_5QaeV6056WsBaecRWaHOfT6j8-
SlAp45E8K5wmt6eYb3hWGyCXmi-
catOmkXcUtmKV9TqWYCu9xClqQdHvWw5pWJJxVBAfG9QP8rO56S90MLwHFzo973Fm7yVJOSBzxCVPBZkc887Tmd3nryUFUNMo_f0bPBv
Rlltac2e92cCHzCXsmXZc9hhXAX-607lpABtpo8U8rNulypvMisj0ykorkJukSM-nhf8sdXiXrgZv6K00CsgYS9G-
hmrE0Q3Qs1u7aV_N0KrtSkCkqTDMLJRnPrAGRTtO-OUGioTTHMd6xz2FmDehUseQIKASzRHmtClclJnUvWimDIVcFOE-
1cN2WABxTDv_tdDeeGHfmEHS8-NAonfVog_m-5eFdPYSEU7Du5nQ6JIC3x6nEKFk7TVzo-
H1P2C5Bc7jv5lHQU88iL1dwKcuuVqKQ3vh0H1K7qwok3g8mkocY-
q2CimB44c43LHA8nGOrCkTlruPeJQQid6GmYSR1KRyeoga0dTPQEefe2i3iTesknMRW1SCpOj7F1pwh2YTXP_ZWafr628xV0nVob0kToVMPic
```

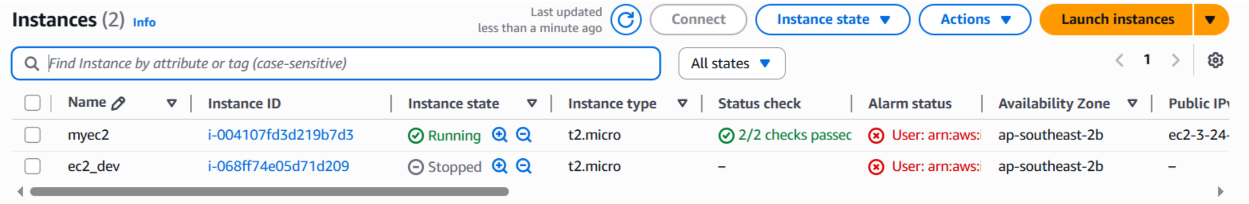
i-004107fd3d219b7d3 (myec2)

The banner is telling us this because we're not authorized! We don't have permission to stop any instance with the production tag.

- Let’s stop the **development instance** and see if we can stop it.



▬	Name	Instance ID	Instance state	Instance type	Statu		Availability Zone	Public IP
☐	myec2	i-004107fd3d219b7d3	✓ Running	t2.micro	✓ 2/		ap-southeast-2b	ec2-3-24-
☑	ec2_dev	i-068ff74e05d71d209	✓ Running	t2.micro	✓ 2/2 checks passec	☒ User: arn:aws:	ap-southeast-2b	ec2-54-79


Instances (2) Info
Last updated less than a minute ago
Connect
Instance state ▼
Actions ▼
Launch instances ▼
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 IP
☐	myec2	i-004107fd3d219b7d3	✓ Running	t2.micro	✓ 2/2 checks passec	☒ User: arn:aws:	ap-southeast-2b	ec2-3-24-
☐	ec2_dev	i-068ff74e05d71d209	⊖ Stopped	t2.micro	–	☒ User: arn:aws:	ap-southeast-2b	–

✓ Here we can see our development instance has been stopped.