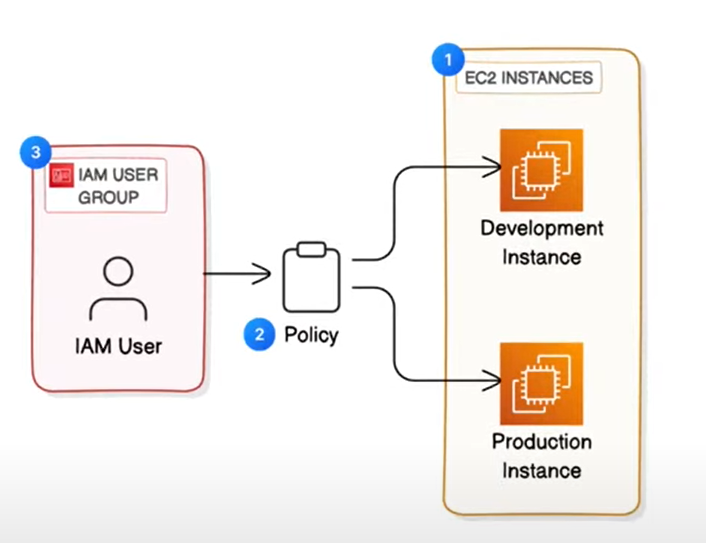
**Cloud Security with AWS IAM:**

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

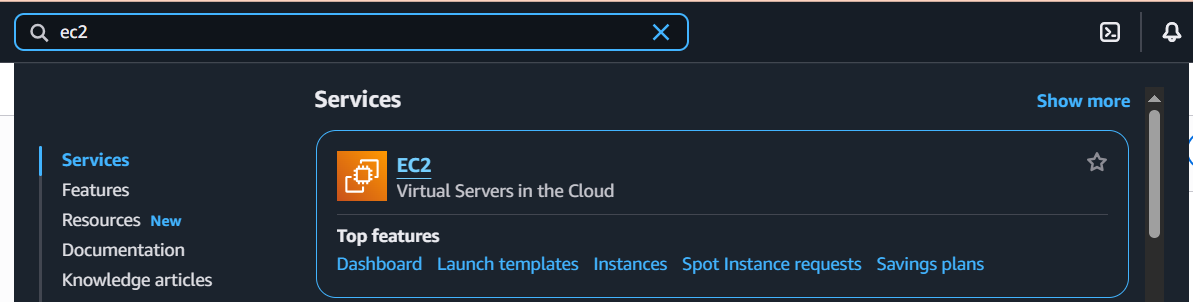
We will be using two EC2 instances (1 for the production environment & 1 for the development environment)

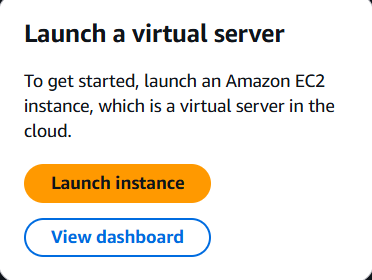
The **development** environment is where developers write, test, and debug code before it's deployed to **production**, which is the live environment that your end users can use.

* **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:**

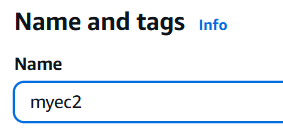
* [Log in to your AWS Management Console](https://console.aws.amazon.com/).
* Open your EC2 console - search for it at the search bar.



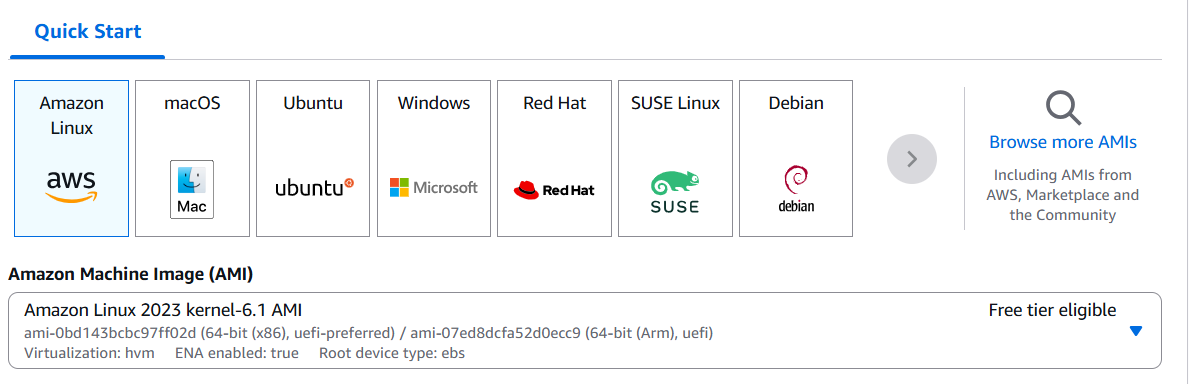


* Click on Launch instance.
* Give it a name like this:

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.

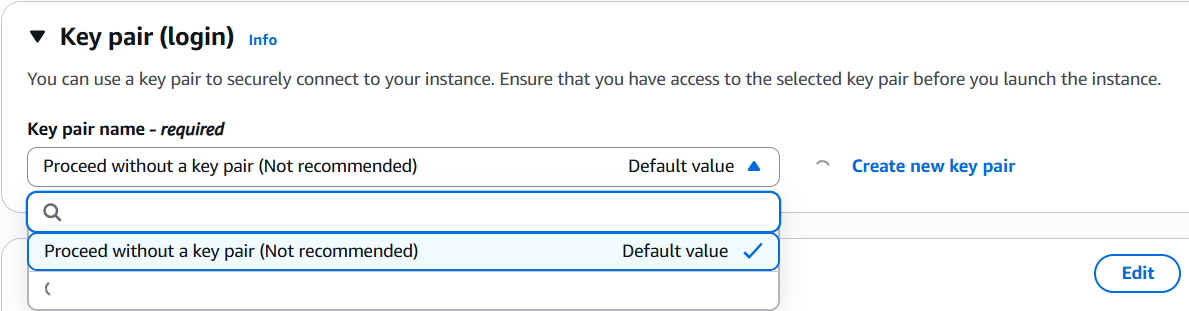


* Choose the image (it is basically operating system which you are going to use in your virtual machine)



* We can proceed without a key pair.

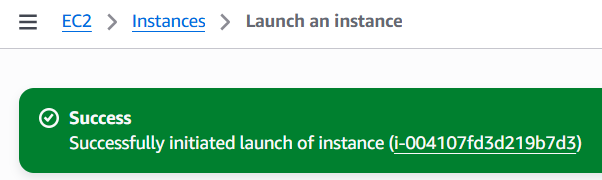
Key pairs allow you to access your EC2 instances without going through console.



* Scroll down with default settings and click on Launch instance

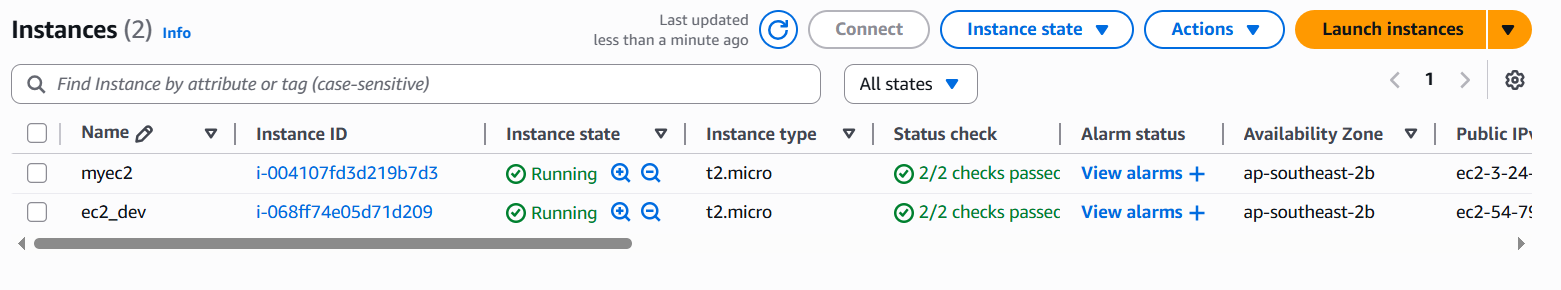
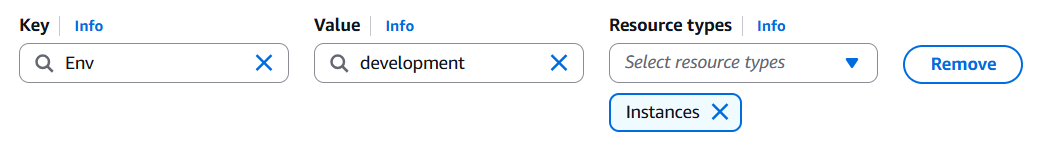


Our EC2 instance will be created.



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

Repeat the same flow, but this time using different tags

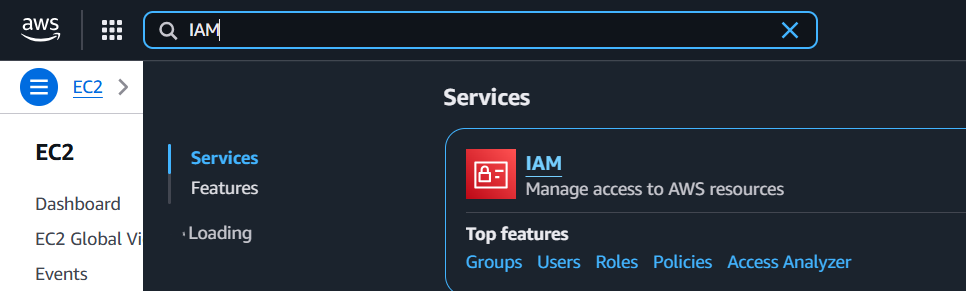


**IAM Policy:**

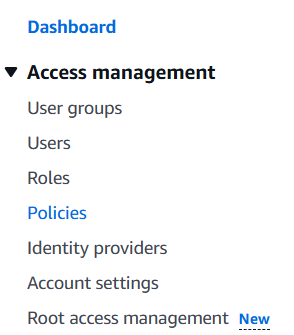
We’ll set up some permission policies. We will set up a permission to access development EC2 instance but not the production instance.

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

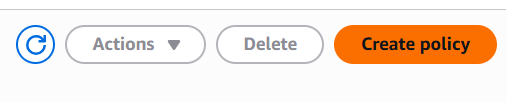
* Go to the IAM console



* On the left side, click on the Policies

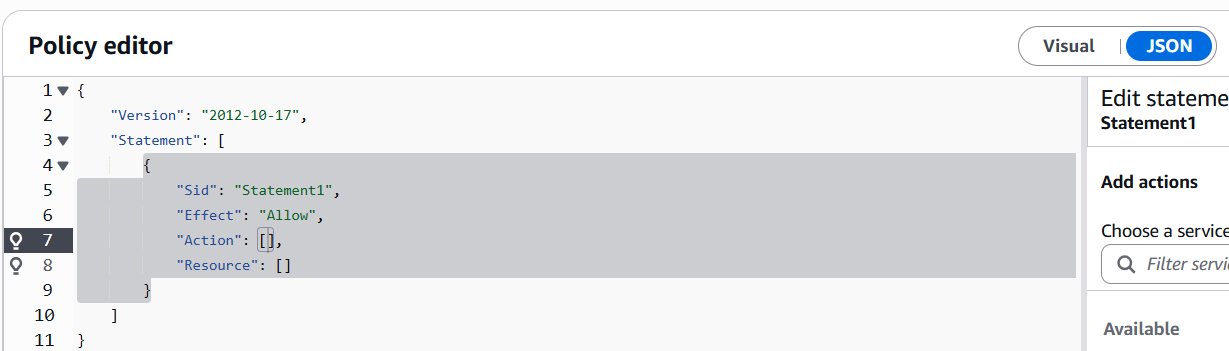


* Click on Create policy



* Switch your Policy editor tab to JSON.

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

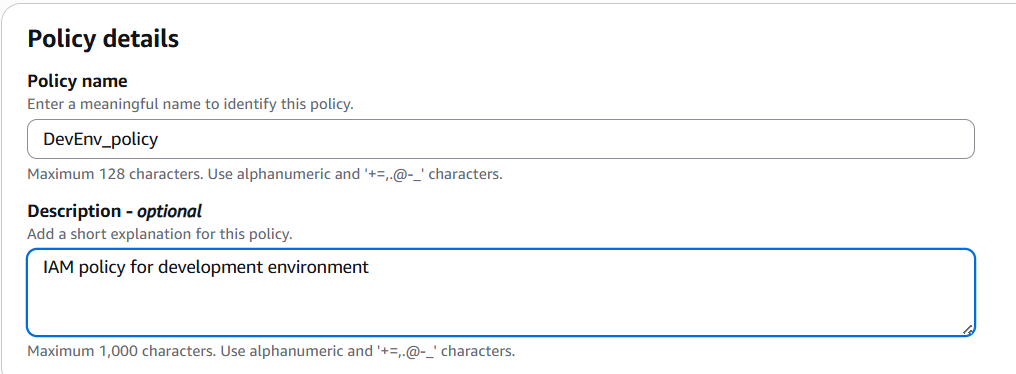


You can paste your policy in the policy editor.



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.



**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.

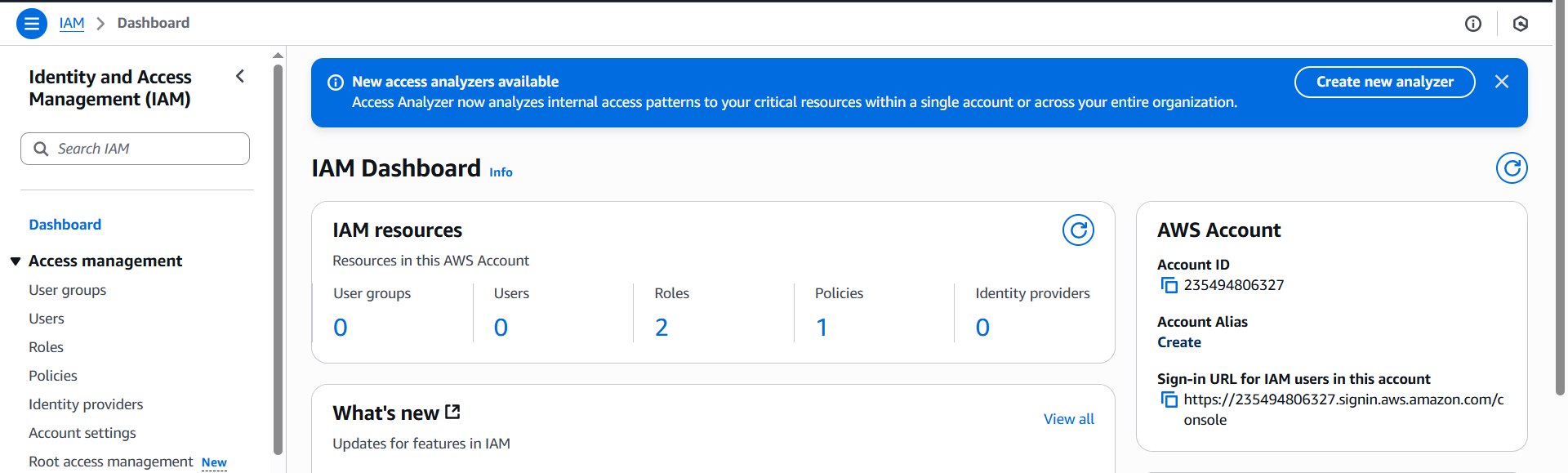
Your AWS account's sign-in page has this URL by default: https://**Your\_Account\_ID**.signin.aws.amazon.com/console/

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/

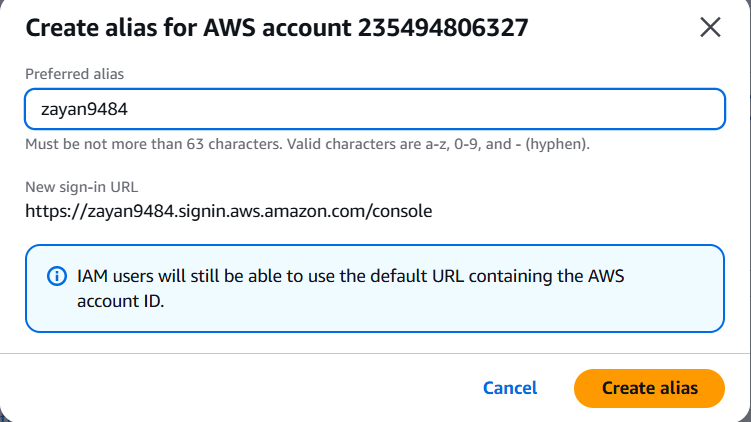
You would create an alias to make it easier to remember and share your AWS console's login URL with others

**Procedure:**

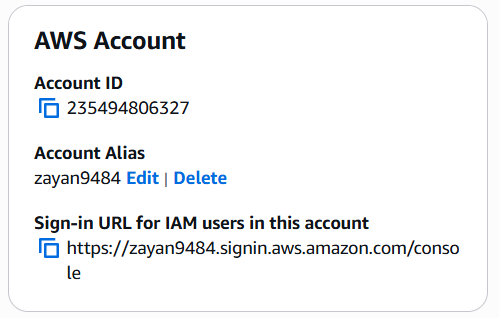
* Still in your IAM console, select Dashboard from the left hand navigation panel.
* In the right-hand side of the dashboard, choose **Create** under **Account Alias**.



* Click on Create and write the name



Alias will be created.



**Creating IAM User:**

**In this step, we will:**

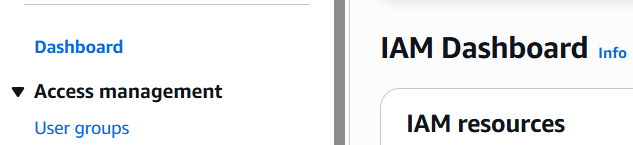
Set up a dedicated **IAM** **group** for all users, so we can manage all users permissions from one place. Set up a dedicated IAM **user**, so our user had a way to log in.

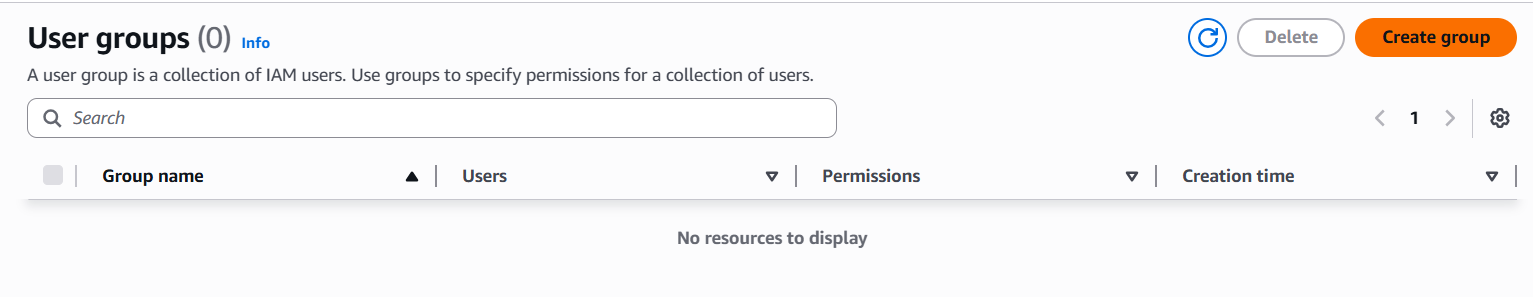
**Procedure:**

* 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.

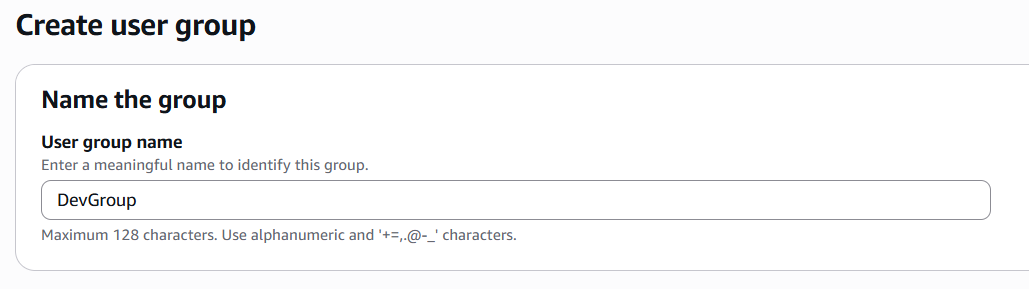
* Choose **Create group**.

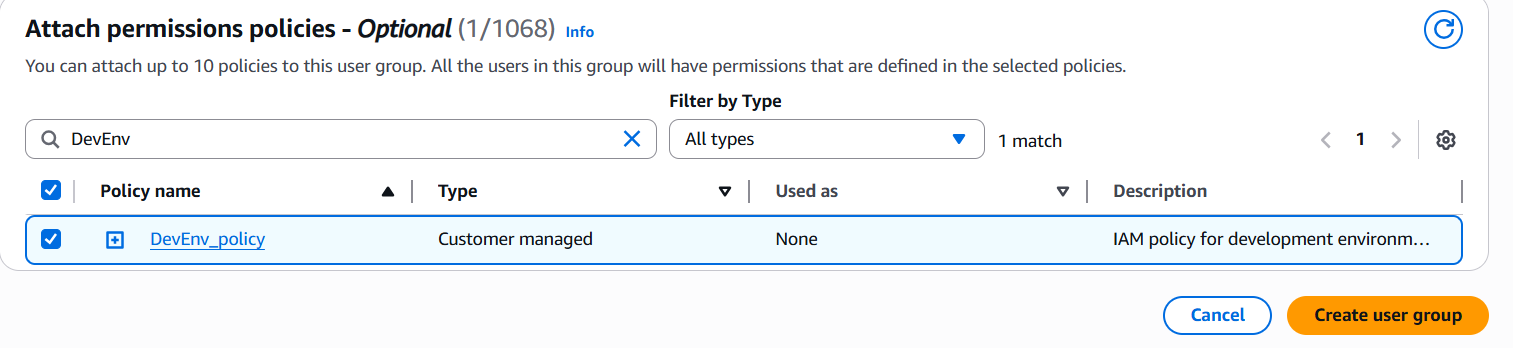


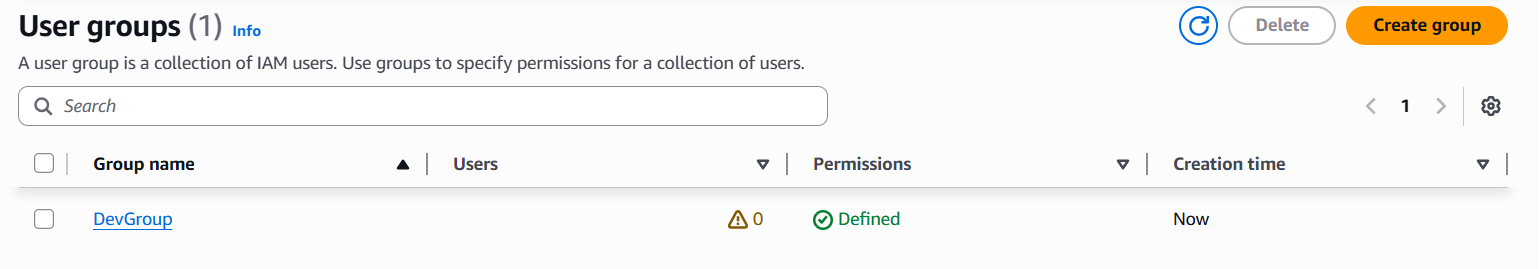


To set up your user group:

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

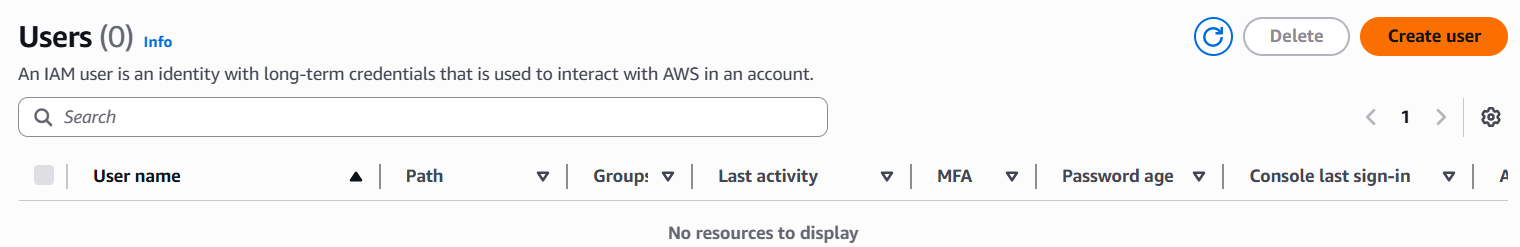


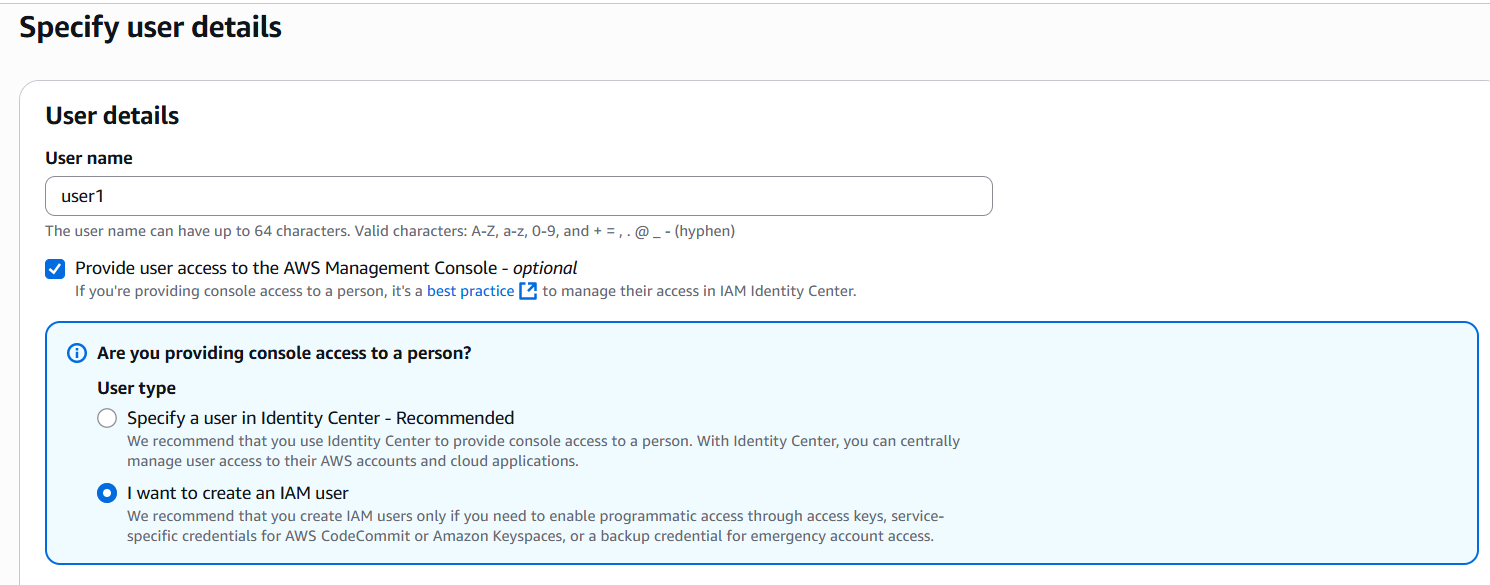


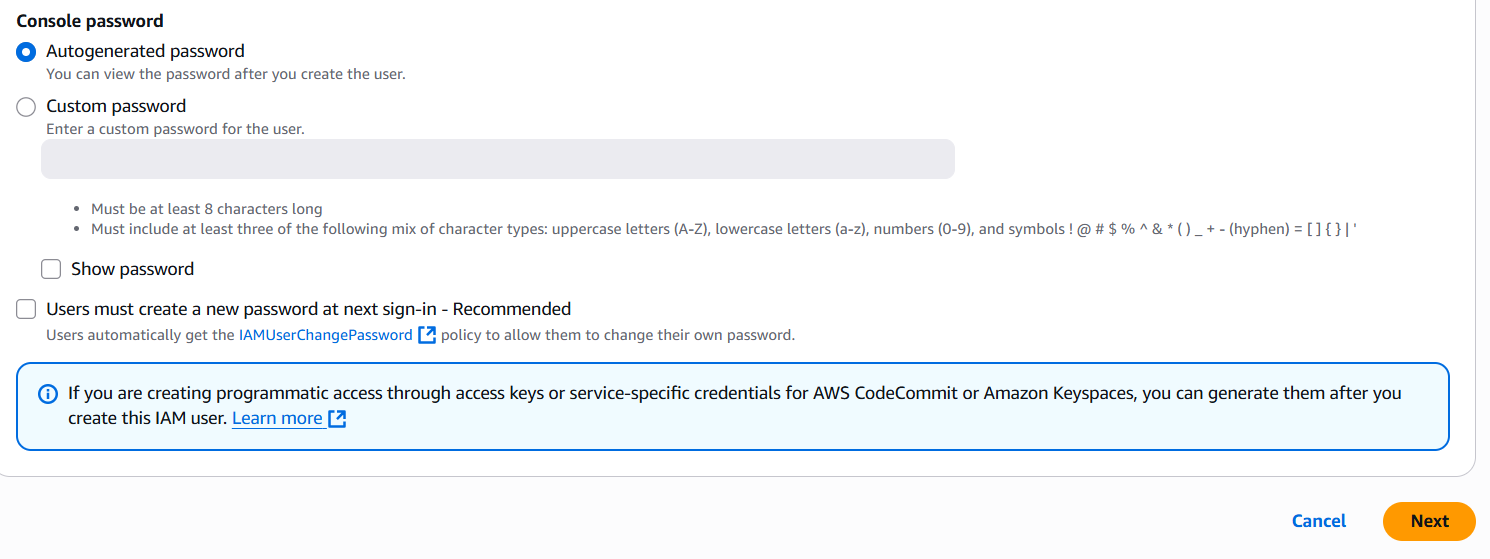


Now add the Users to the User Groups:

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





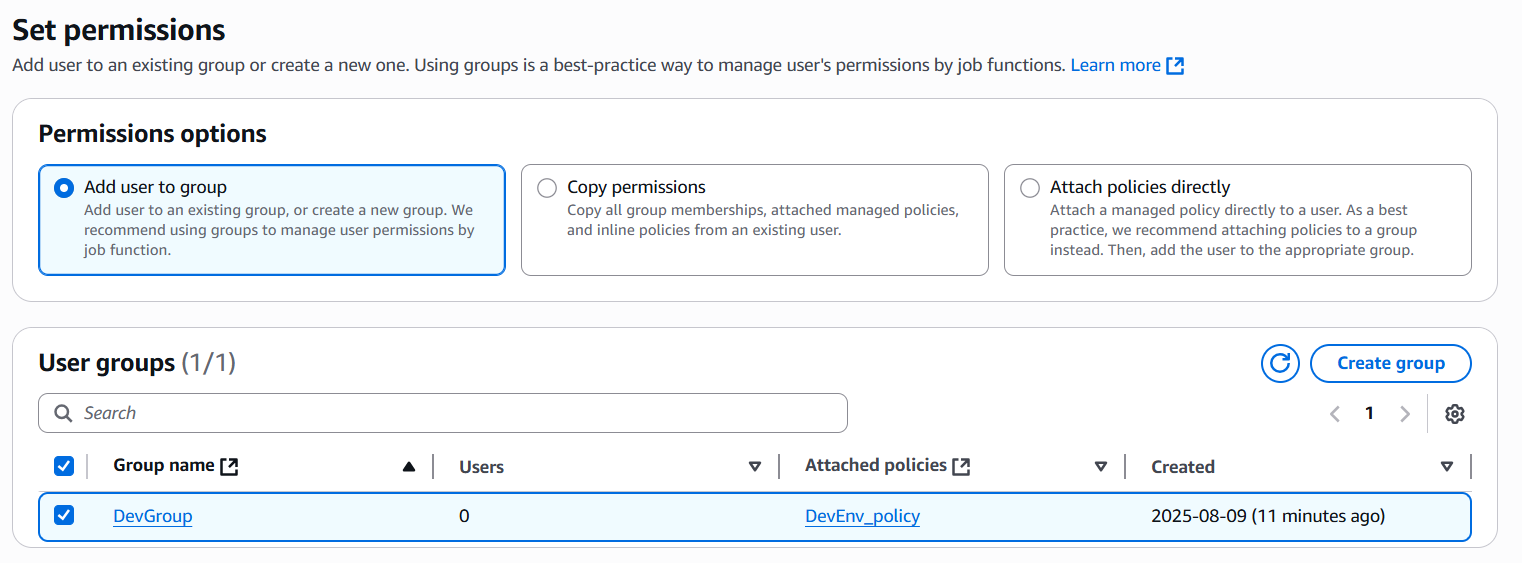


* Tick the checkbox for **Provide user access to the AWS Management Console**.

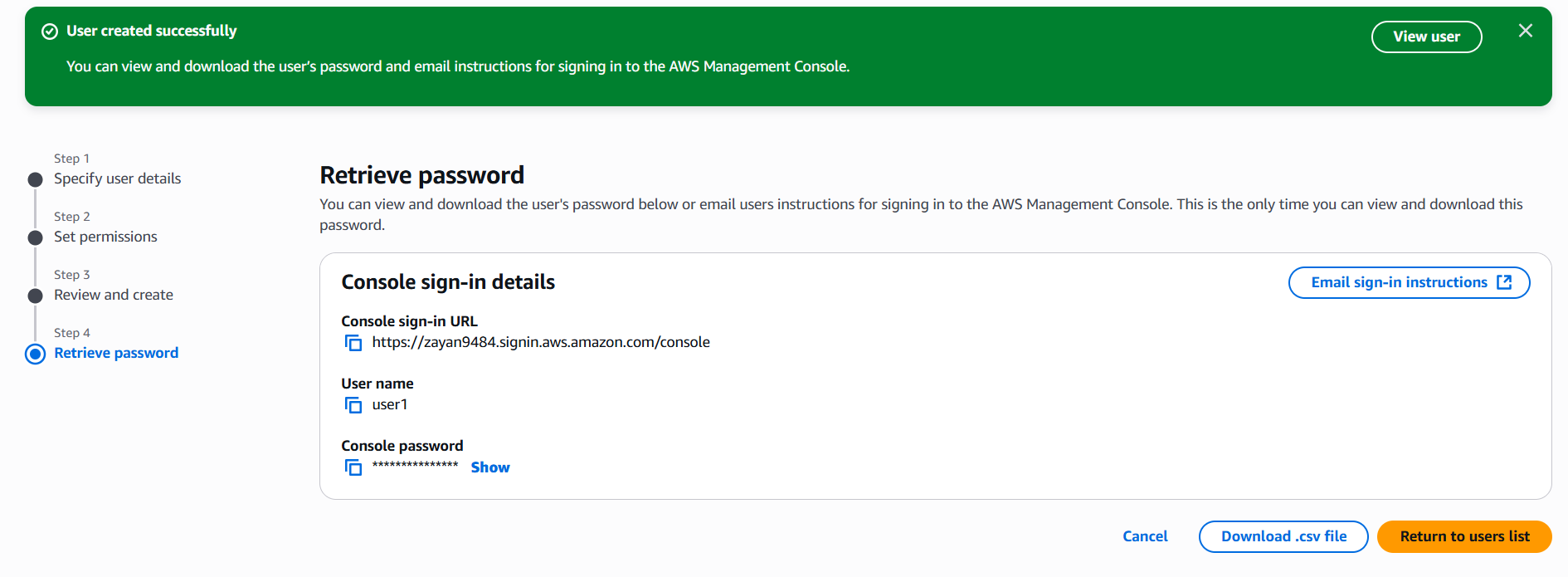
**Why are we doing this?**  
If you don't tick this box, your new user won't get to sign in and access AWS services through the Console. They'll have to access AWS services through other, more advanced methods **e.g., AWS CLI**

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



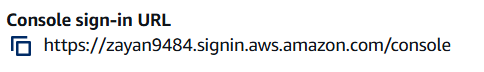
* Click on **Next** and then **User** will be created



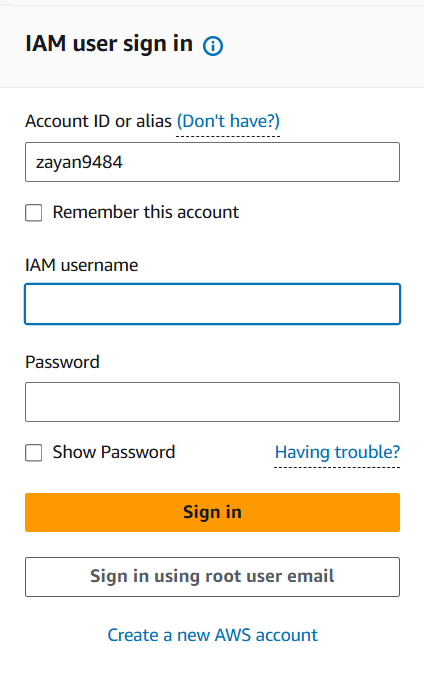
We can download the .csv file which contains the login credentials of the user.

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).

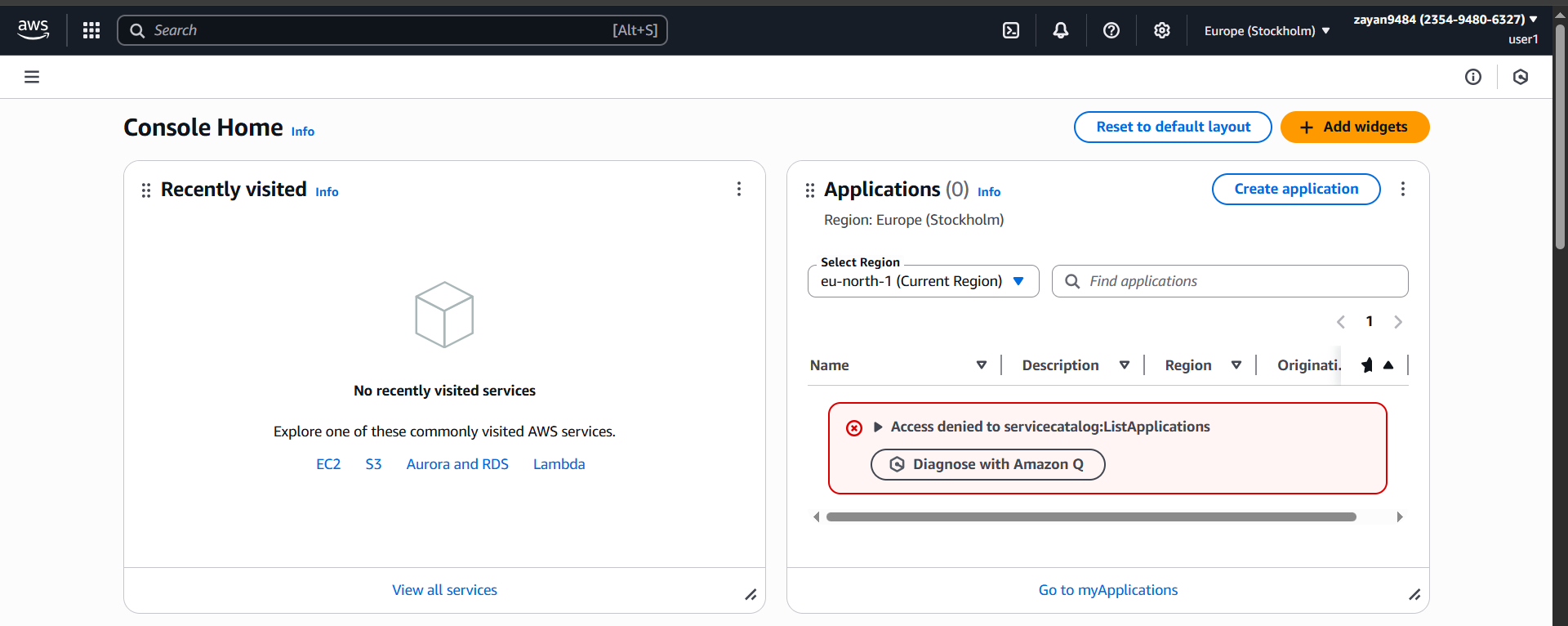
* Copy the **sign-in** URL and paste it in the browser

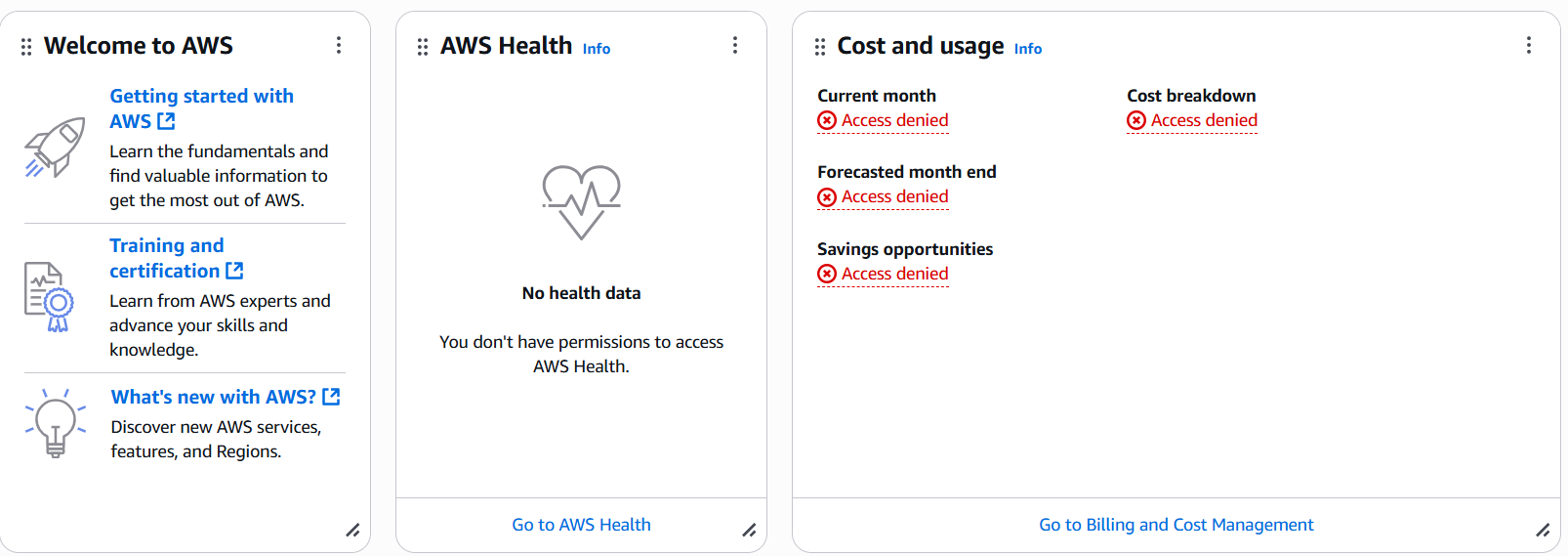


We can see our alias already written here. Use the credentials which we’ve downloaded to login



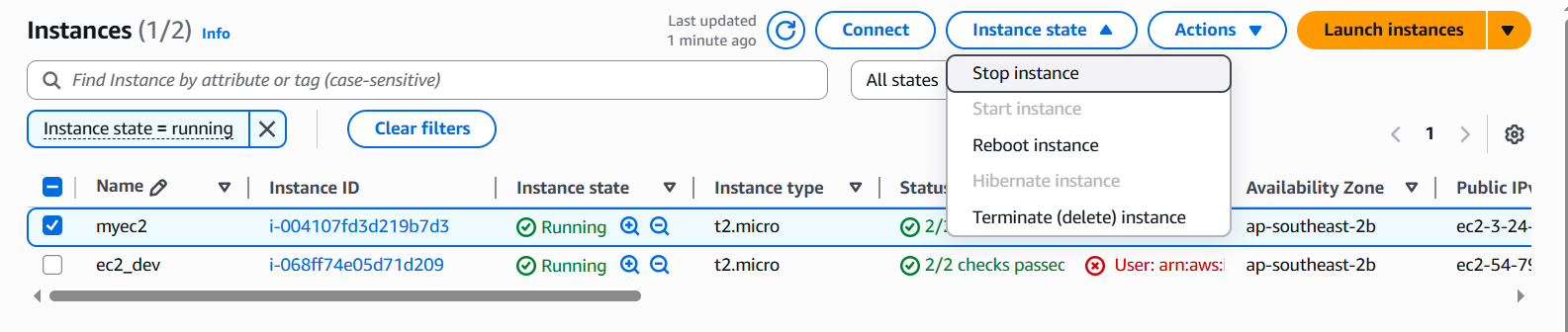
* We can see on top right corner our alias name and we cannot access some features or things because of the policy we’ve created





* 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**.

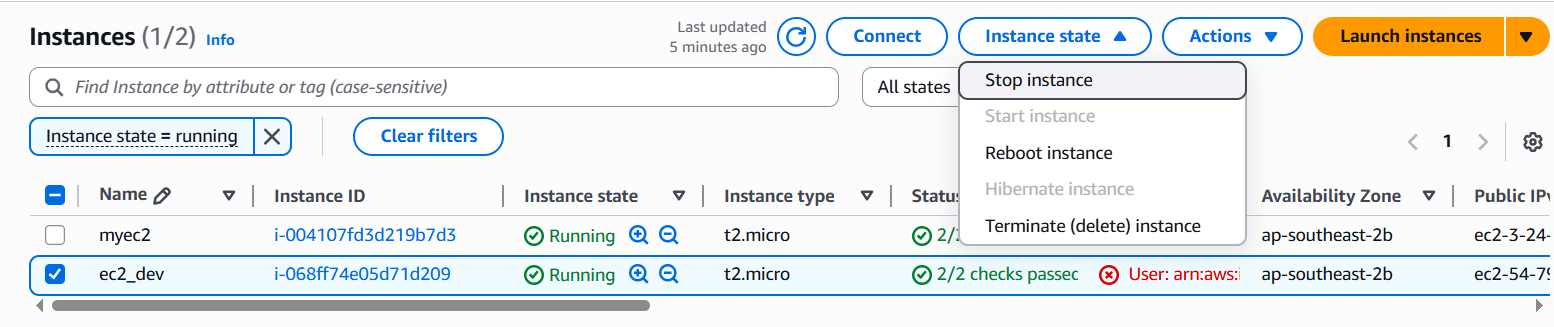
Let’s see if we can stop our production instance.

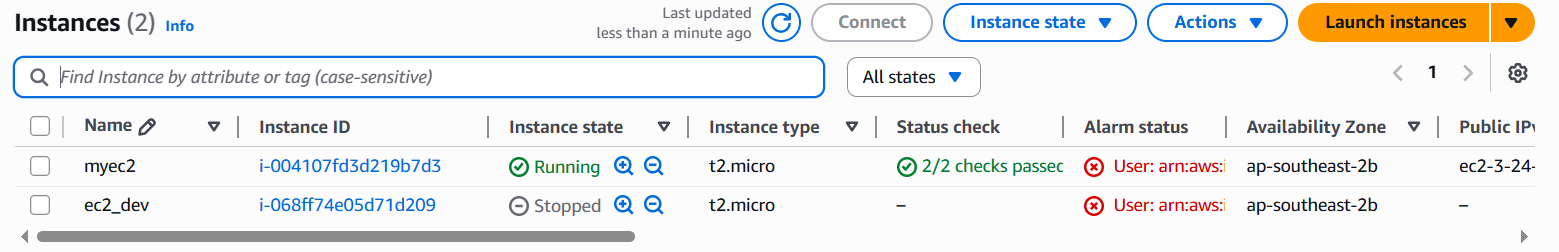




The banner is telling us this because we're not authorized! We don't have permission to stop any instance with the production tag.

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





Here we can see our development instance has been stopped.