Cloud Security with AWS IAM

**Cloud Security with AWS IAM**

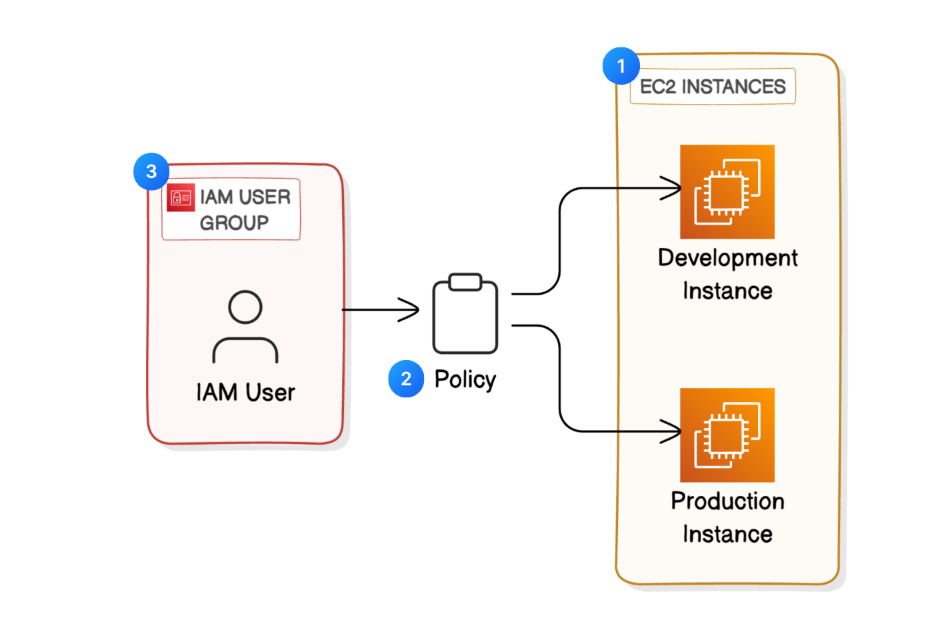
**30 second Summary**

In AWS, a user is a person or a computer that can do things on the cloud - just like you right now!

Today, in this Project I will be using the AWS Identity and Access Management (IAM) service to control who is authenticated (signed in) and authorized (has permissions) in your AWS console.

I will 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...

In this project, I will demonstrate how to use AWS IAM to control access and permission settings in my AWS account. I am doing this project to learn about cloud security from the absolute foundations- every company thinks about access permissions, and there are even entire jobs called 'IAM Engineers' focused on the skills we're about to build today



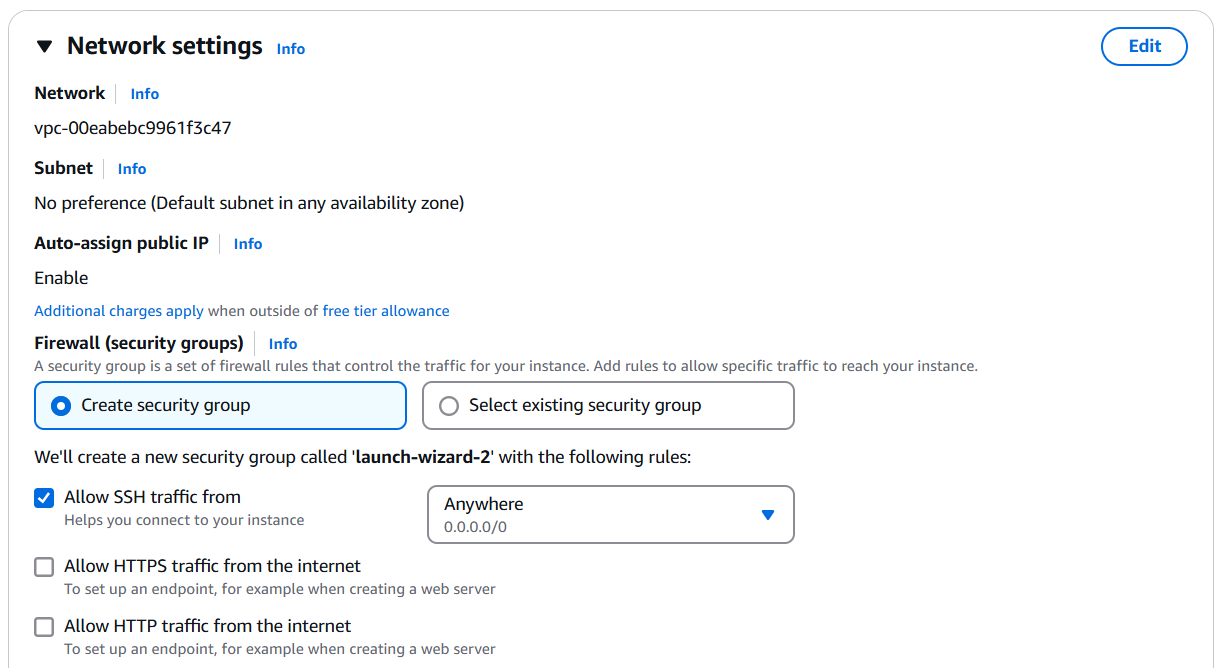
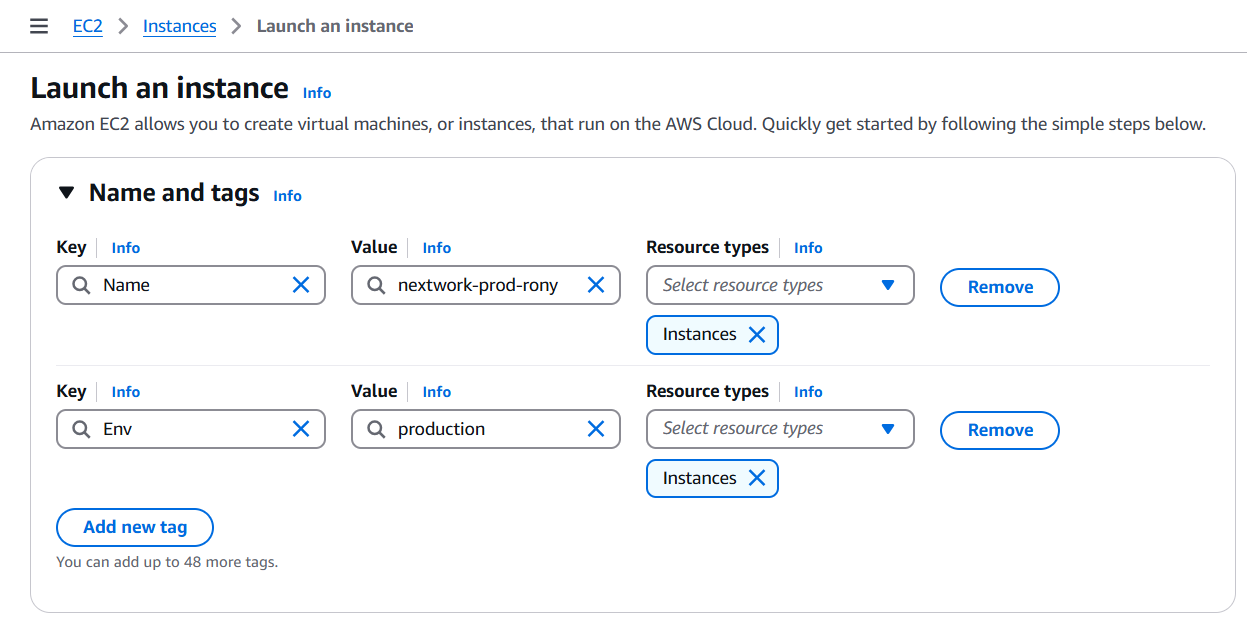
**Core concept of this project**

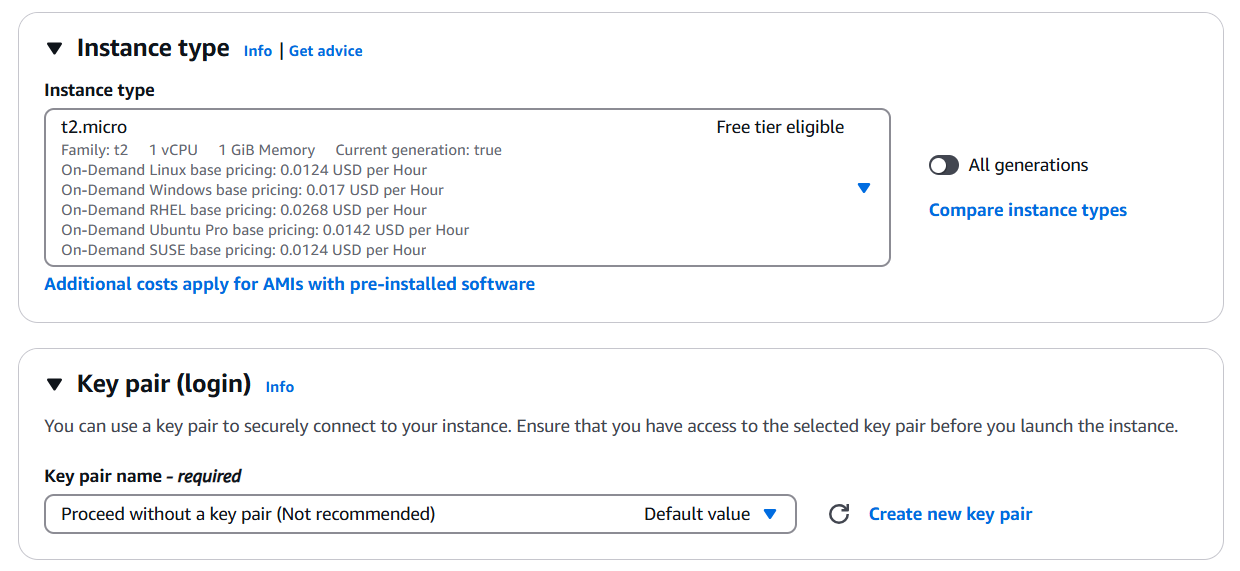
1. **EC2 instances**
2. **IAM Policies**
3. **IAM Users and User Groups**
4. **AWS Account Alias**

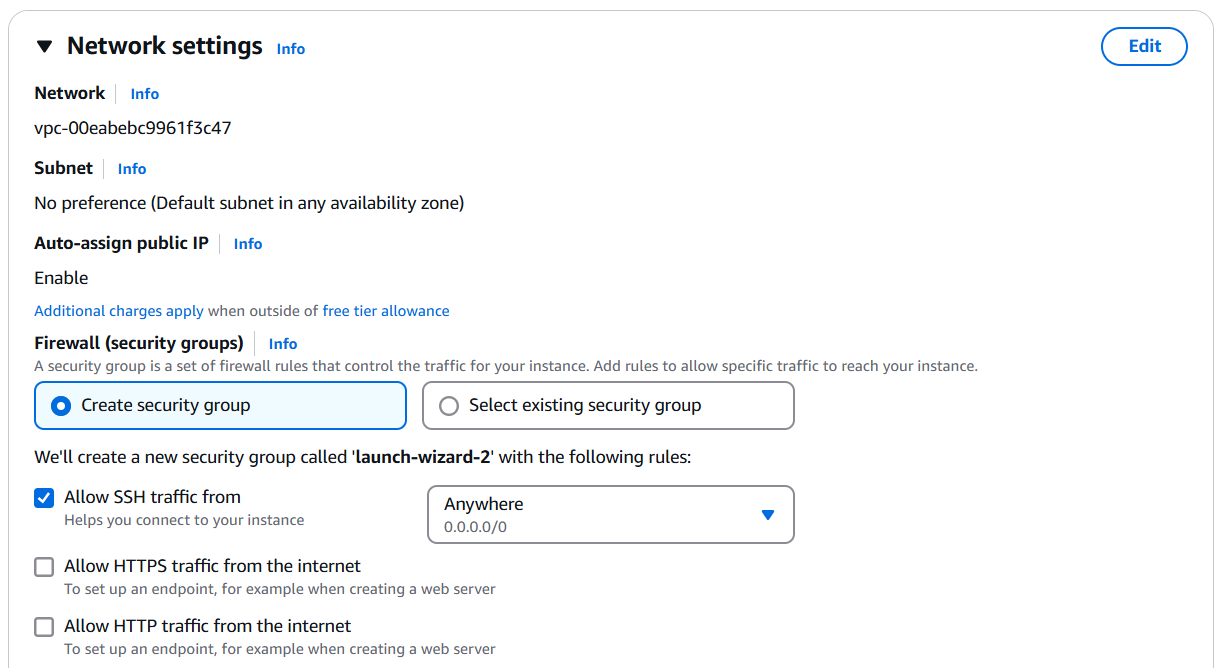
**Launch EC2 Instances**

In this step, we will launch two EC2 instances because we need to boost NextWork’s computing power we're expecting more users and traffic into our website over the summer break!

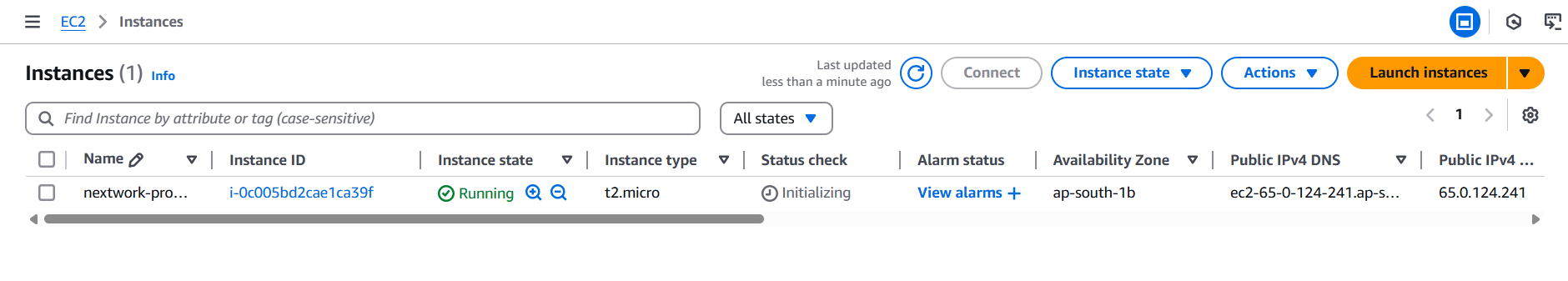
**Creation of EC2 Instance**

****

****

****

**Created EC2 Instance**

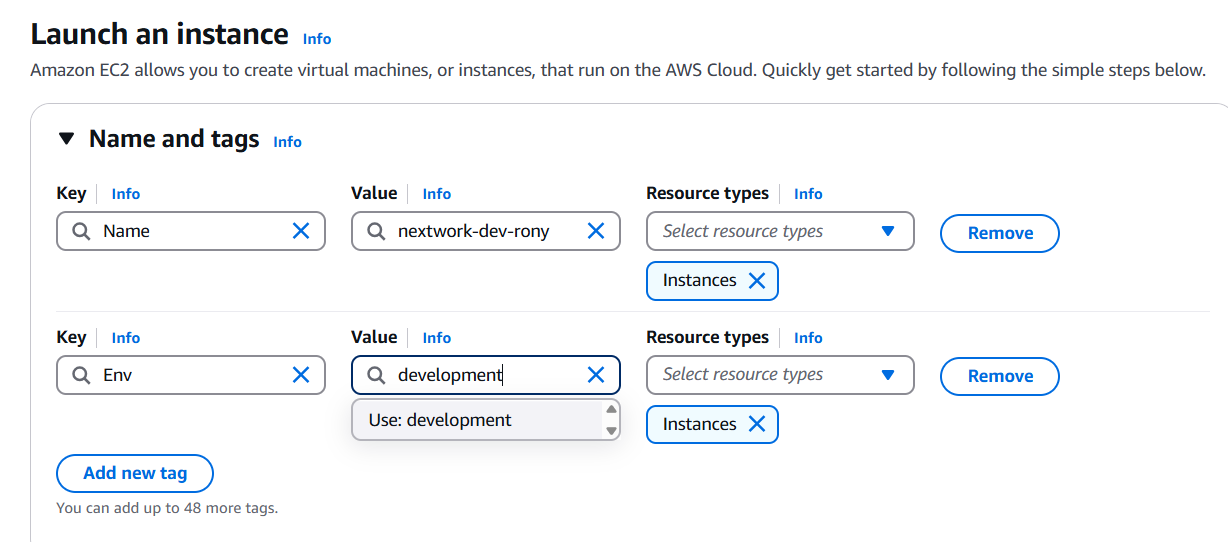
****

Now let's create one more EC2 instance for the **development environment**. To create the development environment the above same procedure

Steps to create EC2 instance:

* Network settings – create security group (default)
* Name and tags – Choose **Add additional tags**, which is right next to your **Name** field
* Add new tag – key: Env, Value: production
* Head on down to see your EC2 settings and make sure the **Amazon Machine Image (AMI)** is using a **Free tier eligible** option.
* For the instance type, also make sure you're using a **Free tier eligible** option!
* For **Key pair (login)**, select **Proceed without a key pair**.
* Click **Launch instance**.

**Second instance**



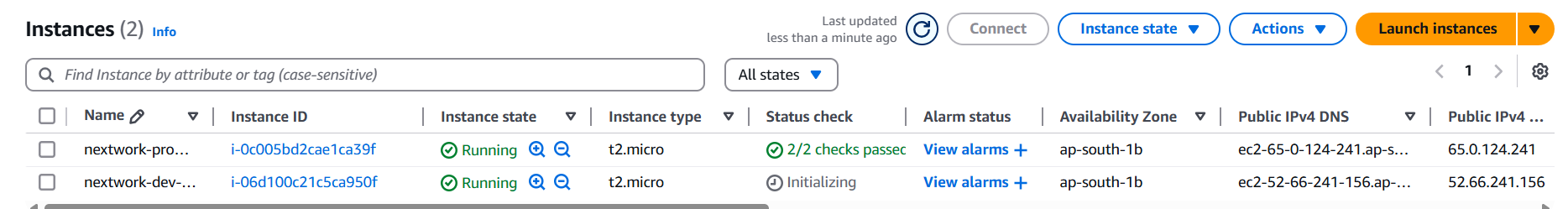
**What are tags? What are they useful for?**

Tags are organisational tools that lets us label our resources. They are helpful for grouping resources, cost allocation and applying policies for all resources with the same tag

**What are the tags and values you've assigned to your two EC2 instances?**

The tag I've used on my EC2 instances is called Env, which stands for environment. The value we've assigned for our instances are production and development.

**The 2 created instance**

****

**Creation of IAM Policy**

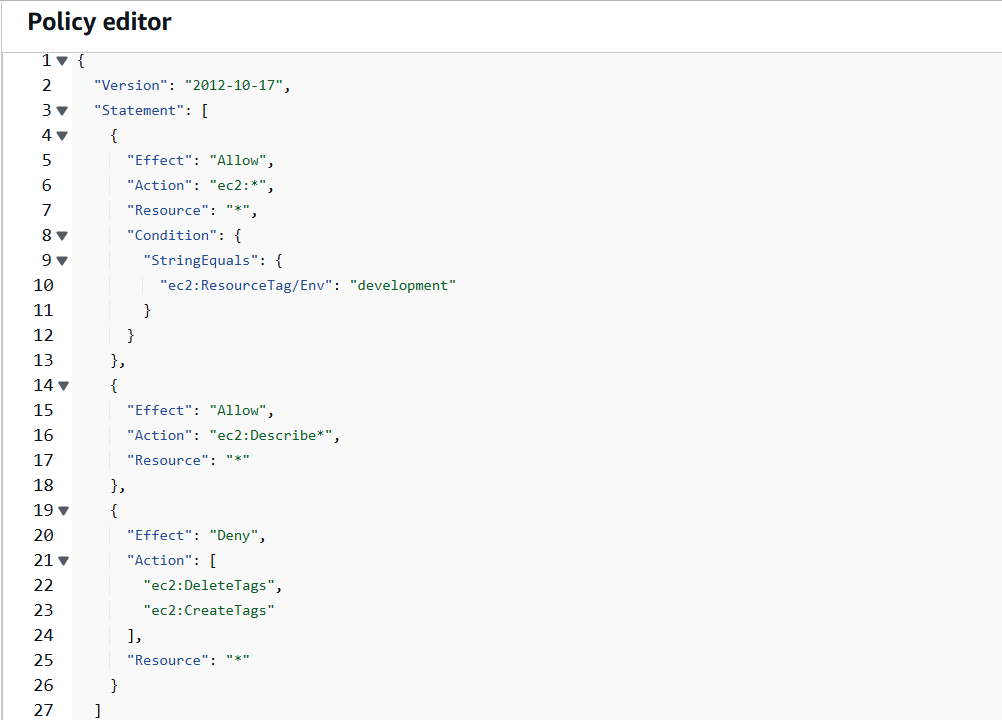
My intern should have permission to the development EC2 instance but not the production instance. I don't want them to accidentally shut down the platform or push their changes to the production environment while they're just testing things!

To start this task, I will use AWS IAM to give my intern access to the development instance first.

**What is IAM?**  
IAM stands for Identity and Access Management. You'll use AWS IAM to manage the access level that other users and services have to your resources.

**What do IAM Policies do?**

IAM Policies are like rules that determine who can do what in our AWS account. I am using policies today to control who has access to our production/environment instance.

****

**Extra for Experts: how are JSON policies structured?**  
***Version***  
‍This means 2012-10-17 is the date of the latest policy version. This tells you whether the policy is up to date with the latest standards and practices.

***‍Statement***  
‍The main part of the policy structure and defines a list of permissions.

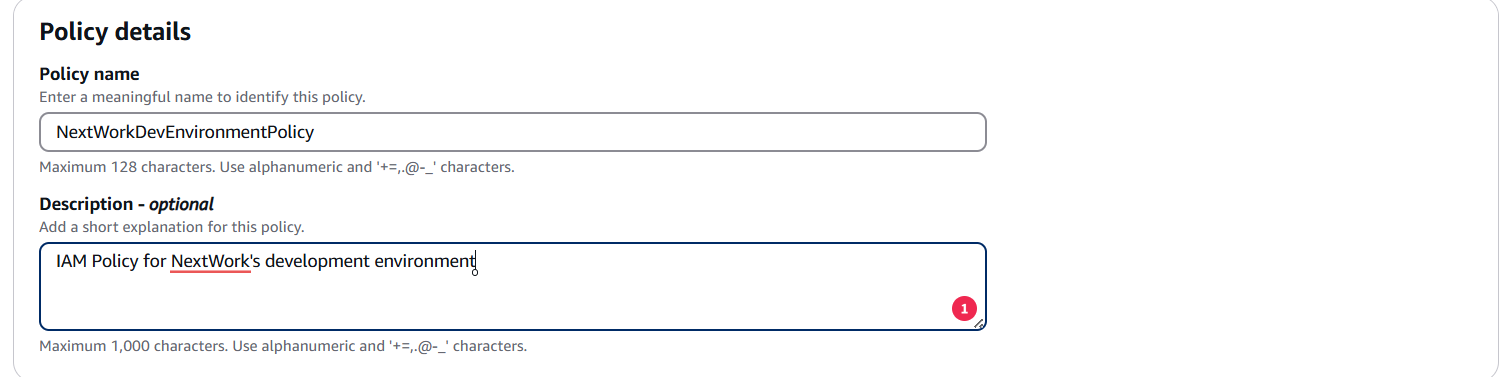
***‍Effect***  
‍This can have two values - either **Allow** or **Deny** - to indicate whether the policy allows or denies a certain action. **Deny** has priority. Looking at the first statement, "Effect": "Allow" means this statement is trying to allow for an action.

***‍Action***  
‍A list of the actions that the policy allows or denies. In this case, "Action": "ec2:\*" means all actions that you could possibly take on EC2 instances are allowed. Woohoo!

***‍Resource***  
‍Which resources does this policy apply to? Specifying "\*" means all resources within the defined scope (see the next point).

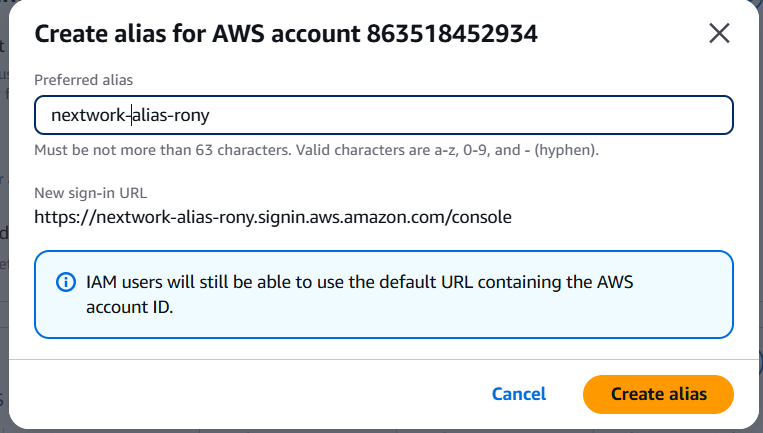
***Condition Block*** *(optional)*  
‍The circumstances under which the policy is in action. In this case, the condition is that the resource is tagged **Env - development**. This means specifying "Resource": "\*" in the line above means all resources with the **Env - development** tag is impacted by your statement.

**Creation of policies**

****

**Create an AWS Account Alias**

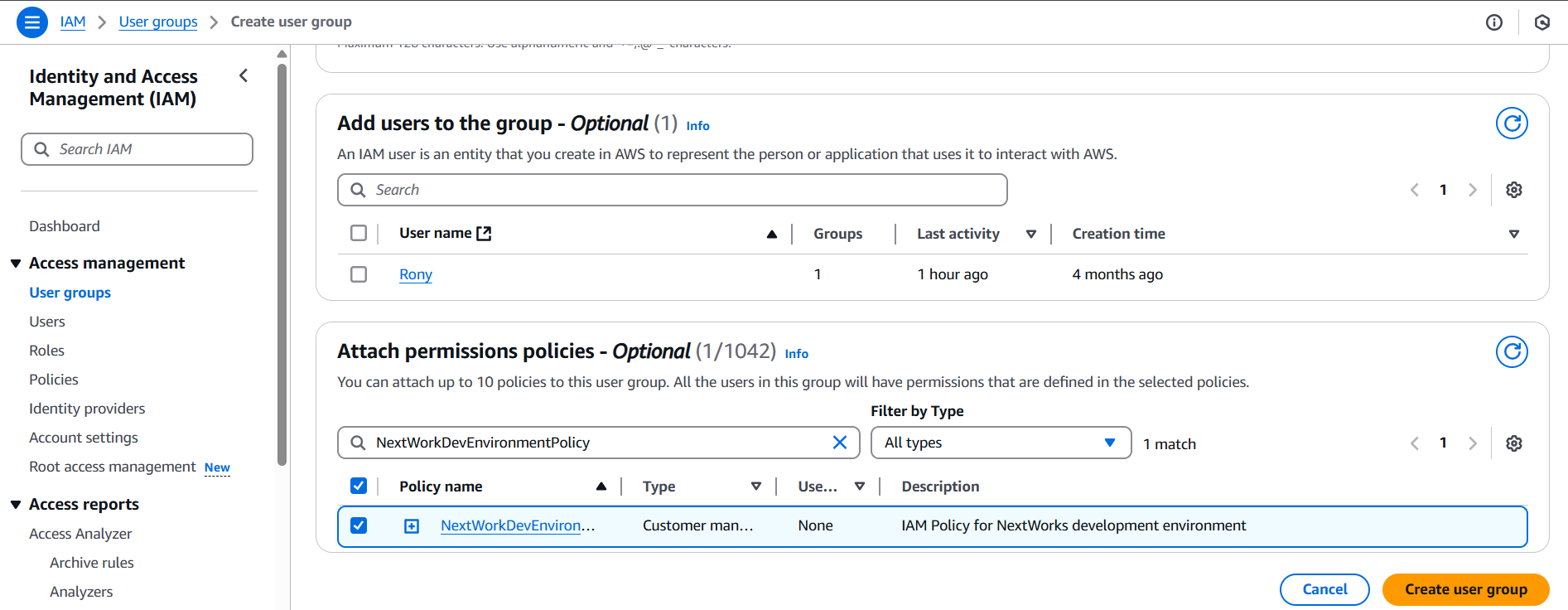
In this step, I will set up an Account Alias, which is like a nickname for i AWS account's console login. This is because an account alias makes it simpler for our users to login!



**What does an Account Alias mean?**

An account alias is simply a nickname for our AWS account! Instead of a long account ID, we can now reference our account alias instead!

**Create IAM Users and User Groups**

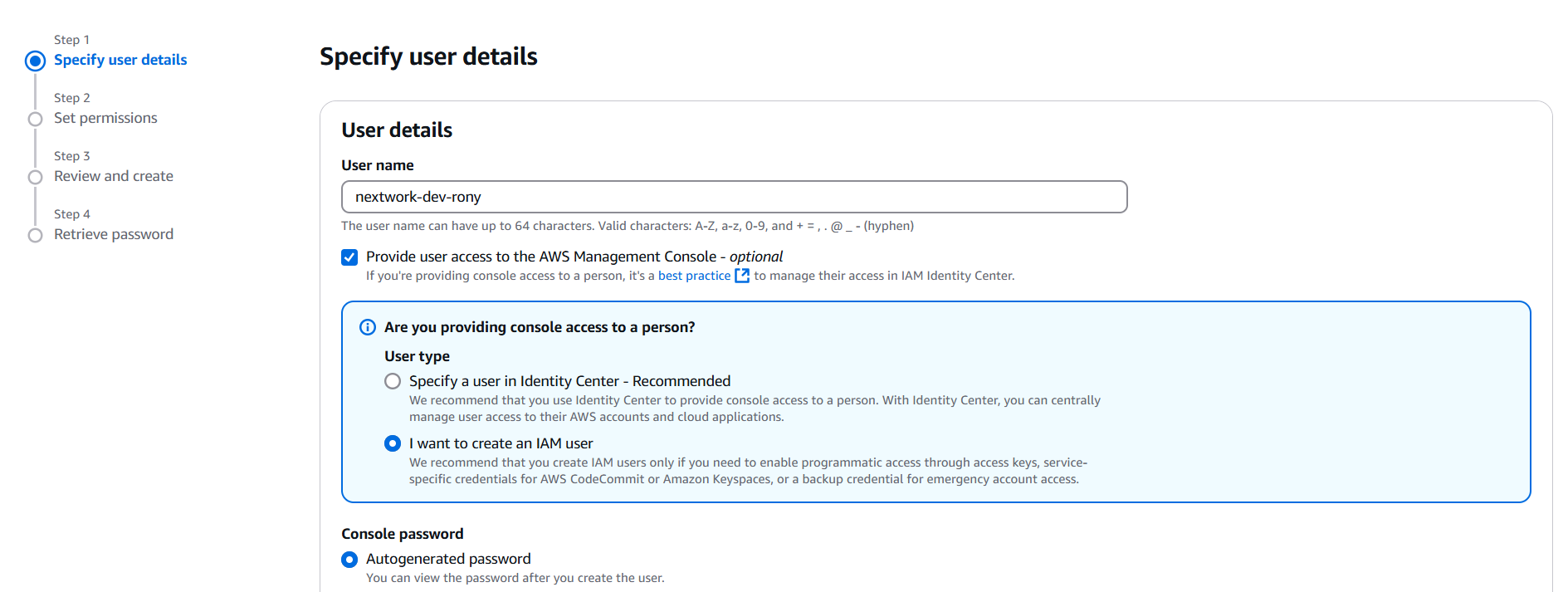
****

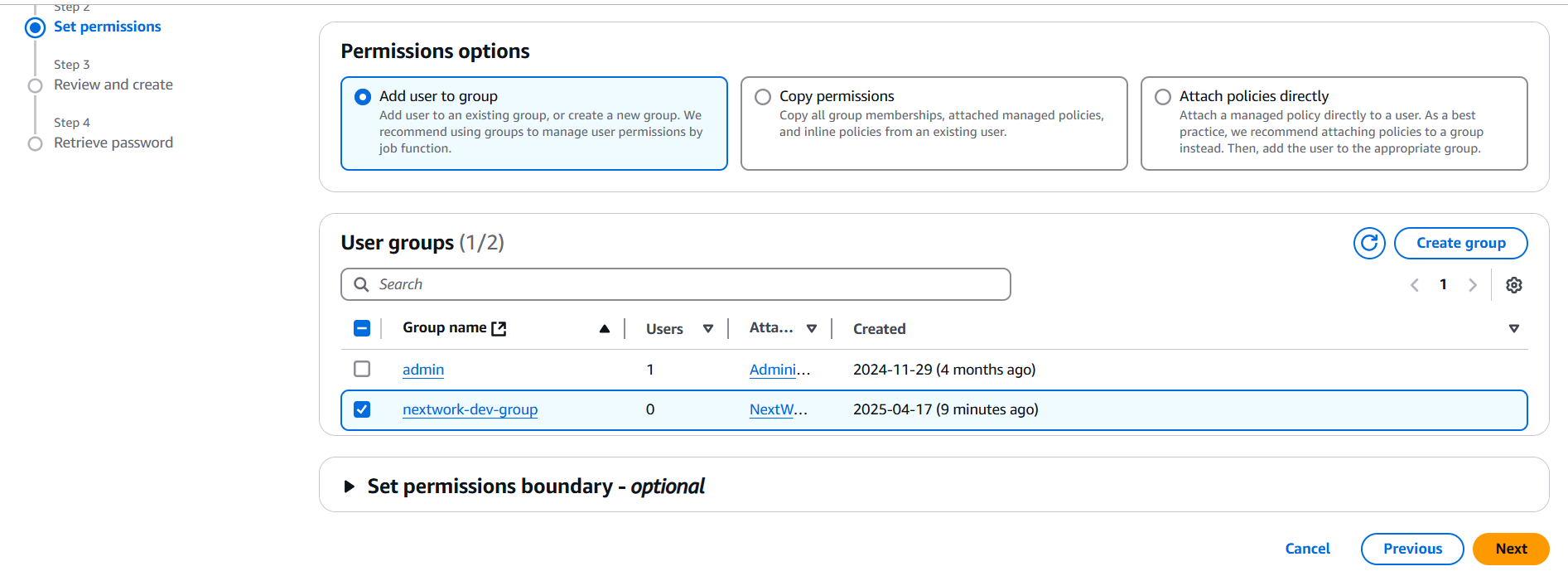
**What are IAM user groups?**

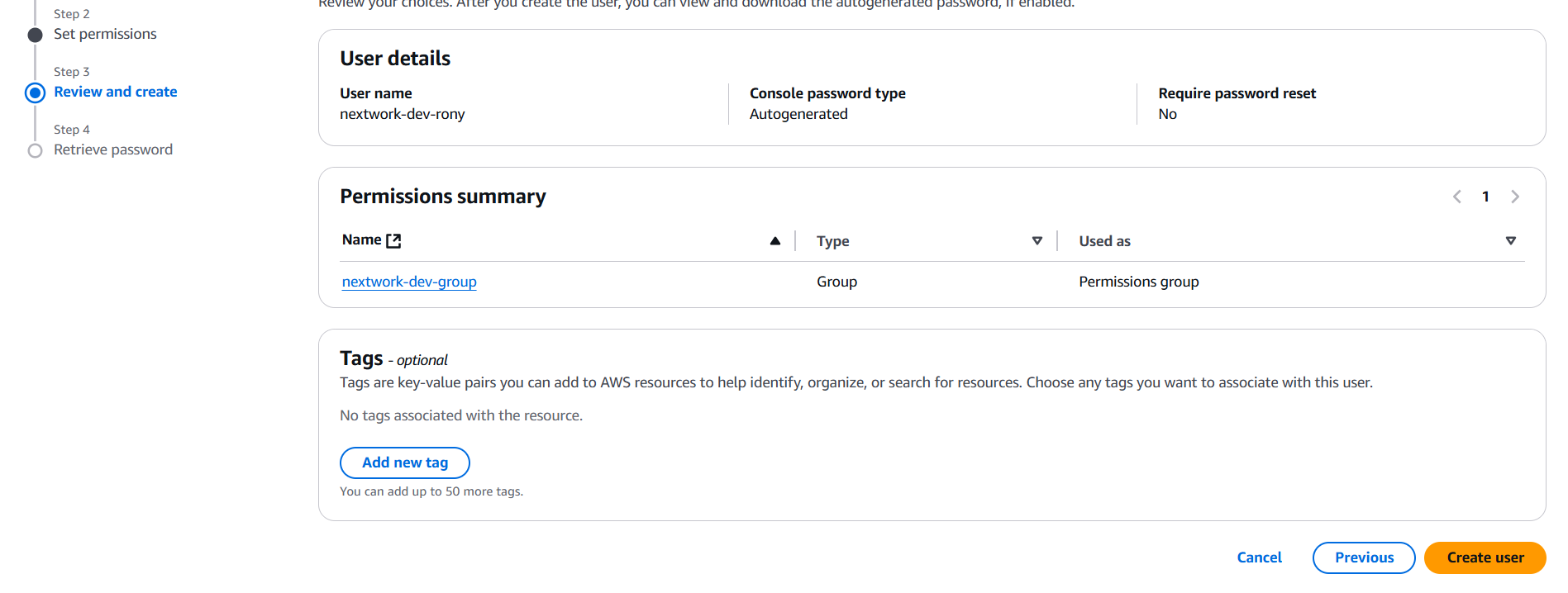
IAM user groups are like folders that collect IAM users so that you can apply permission settings at the group level

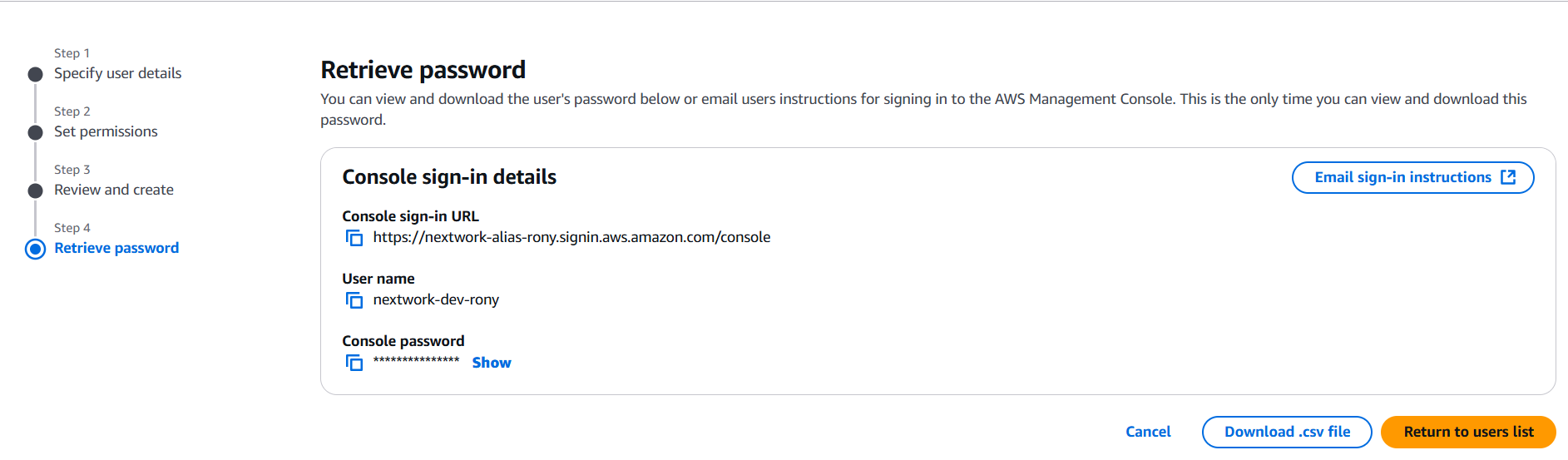
**What is the effect of attaching policy to user group?**

I attached the policy we created to this user group, which means. any user created inside this group will 874 automatically get the permissions attached to our NextWork Dev Environment Policy IAM policy.



****



****

**What are IAM users?**

IAM users are people or entities that have access/can login to our AWS account.

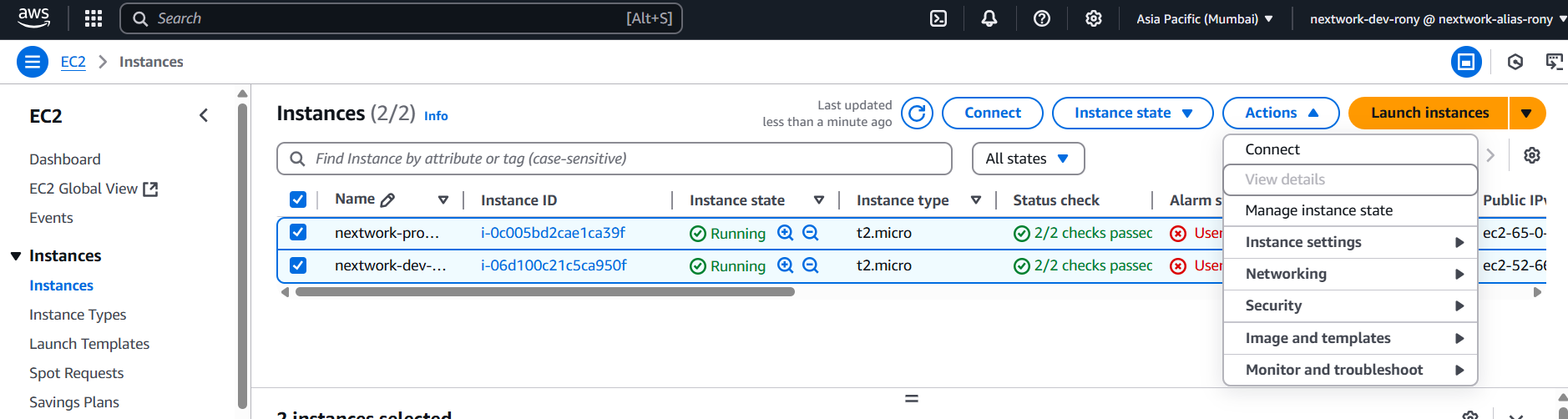
**What are the two ways i could share a new User's sign-in details?**

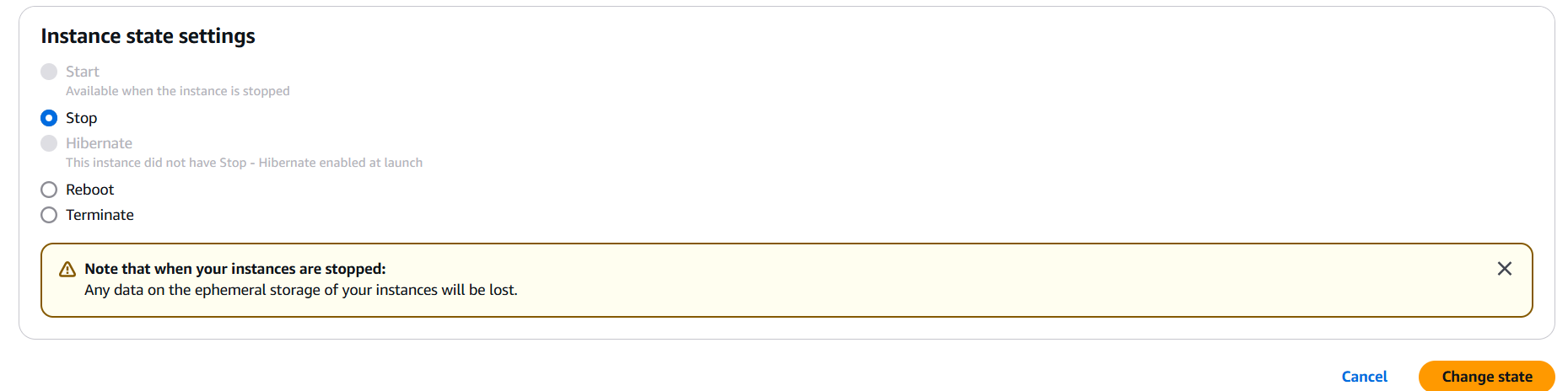
The first way is to email sign-in instructions to the user, while the second way is to download a .csv file with the sign in details inside.

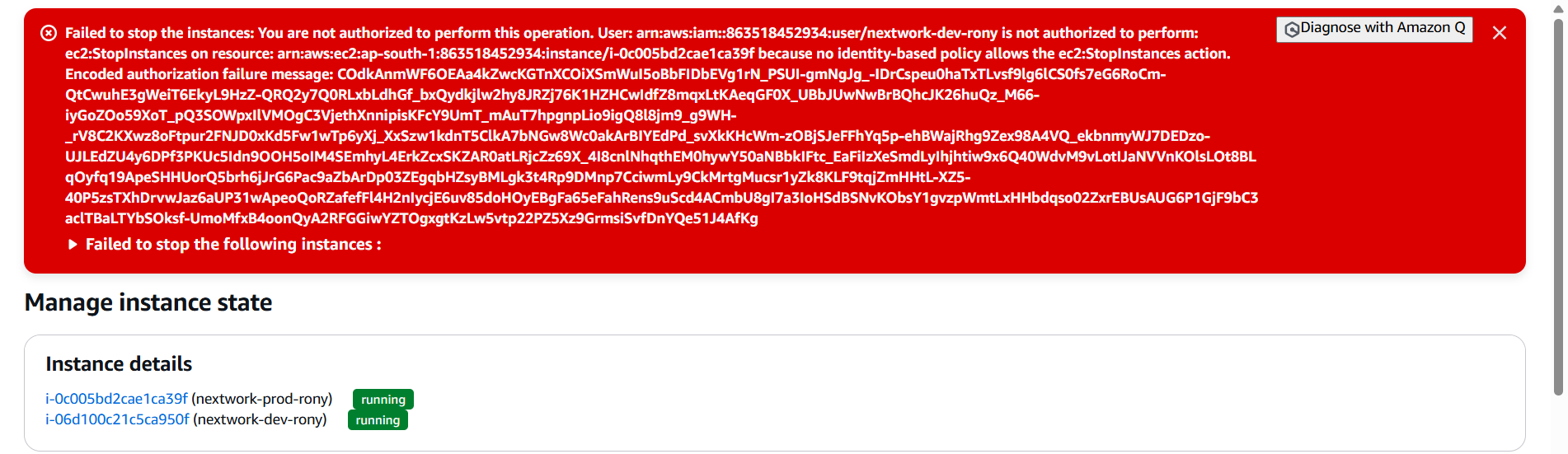
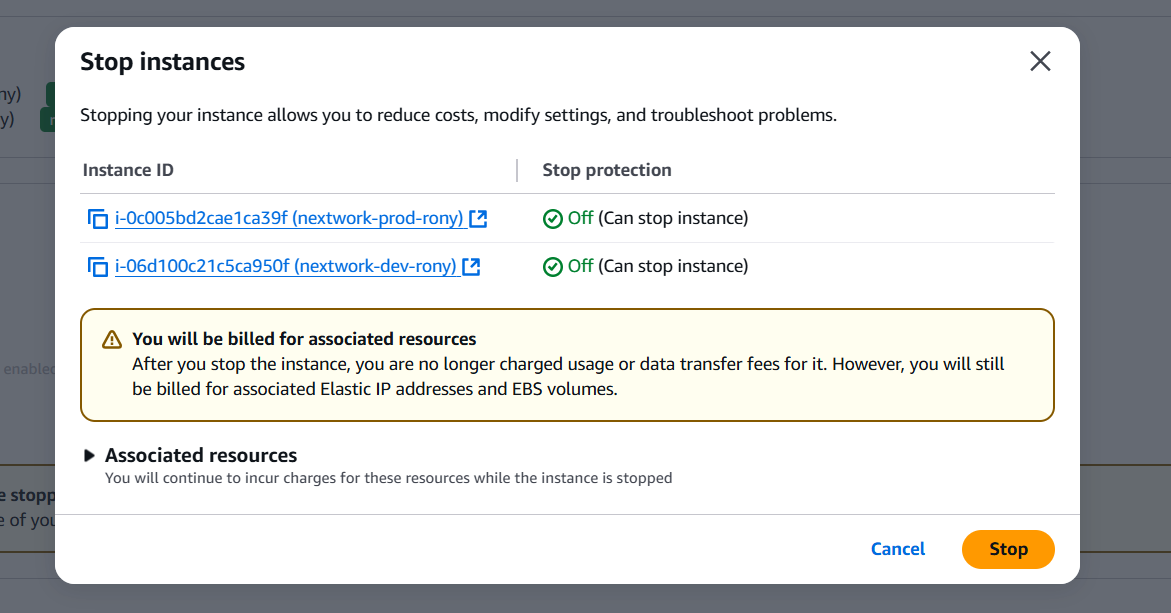
**What did i observe in new IAM user's AWS dashboard?**

Once i logged in as our IAM user, i notice that my user is already denied access to panels on the main AWS console dashboard. This was because my only set up permissions to our development EC2 instance, so my intern wouldn't have access to even see anything else

**When I tried to stop the production instance**

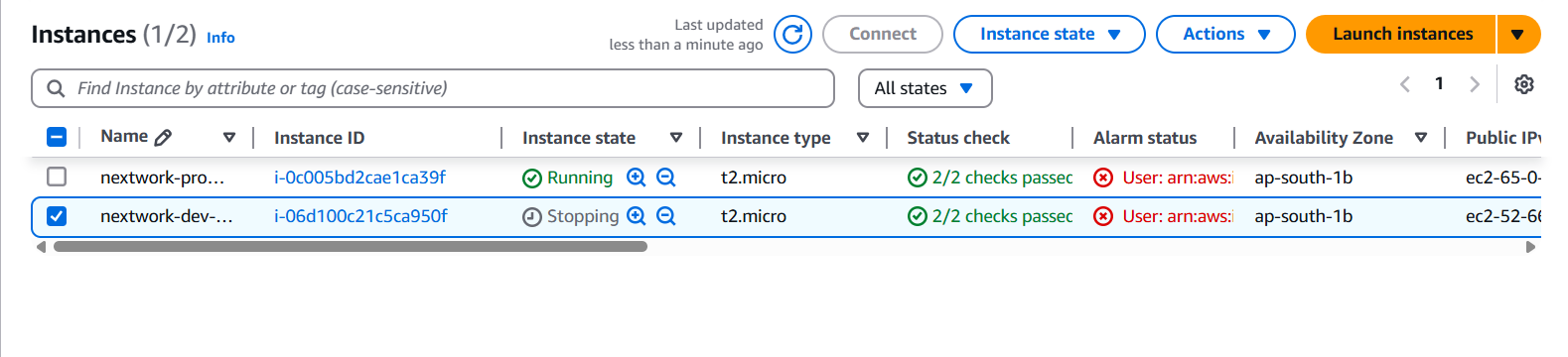






**What happened when I tried to stop the production instance?**

When i tried to stop the production instance, i were met with an error! This was because our production instance is tagged with the 'production' label, which is outside of the scope of my permission policy - interns are only allowed to do things to development instances.



**What was the action i performed on two EC2 instances?**

I tested our JSON IAM policy by attempting to stop both the development and instances.

**What happened when i tried to stop the development instance?**

when i tried to stop the development instance, i successfully saw the instance state change to Stopping and then Stopped. This was because my permission policy allows intern. users next work.

**IAM Policy Simulator**

In the last step, you ended up shutting down the development instance to test your intern's access. Shutting down EC2 instances could get pretty disruptive for your other engineers and your users, so it's best practice to run these tests in another way.



**Why would i use the IAM Policy Simulator?**

The IAM Policy Simulator is a tool that lets us simulate actions and test permission settings by defining a specific user/group/role and the action i want to test for. It's useful for saving time when testing permission settings! No more logging into another user or actually stopping resources

**What were the simulation results for the development instance?**

I set up a simulation for whether my dev group has permission to Stop instances or Delete Tags. The results were denied for both we had to adjust the scope of the EC2 Instances to ones that are tagged with "development". Once I applied that tag, permission was allowed.

**What were the key services and concepts i learnt in this project?**

Services I used today were Amazon EC2 and AWS IAM!

Key concepts I learnt include IAM users, policies, user groups and account aliases.

I also learn how to use the Policy Simulator and how JSON policies work. How to launch an instance, how to tag an instance, how to log in as another user.

**How long did I take to complete this project?**

This project took me approximately 1.5eurs today including project demo time! The most challenging part was understanding the IAM policy since it was written in JSON and it contained multiple statements. It was most rewarding to see permission denied when our intern tried to delete our production instance my IAM access management worked!

Thanks

**Rony Joseph Thomas**