NEXTWORK

MANAGING PERMISSIONS WITH AWS IAM

Kanika Mathur
@github.com/KanikaGenesis

WHAT IS AWS IAM?

What it does:

- **AWS Identity and Access Management (IAM) is a web service that helps you securely control different users' access to AWS resources.**

Why it's useful:

- You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use your account's resources.

How I'm using it in today's project:

- Today, I am going to create two EC2 instances, tagging one as development and the other as production. I will develop specific policies for these instances to manage their actions based on their tags. I will then set up a user group and attach these tailored policies. Finally, I will create a user, granting them permission to start and stop the development instance while restricting their ability to stop the production instance. This setup will ensure secure and controlled management of EC2 instances, tailored to the needs of different environments.

Kanika Mathur
@github.com/KanikaGenesis

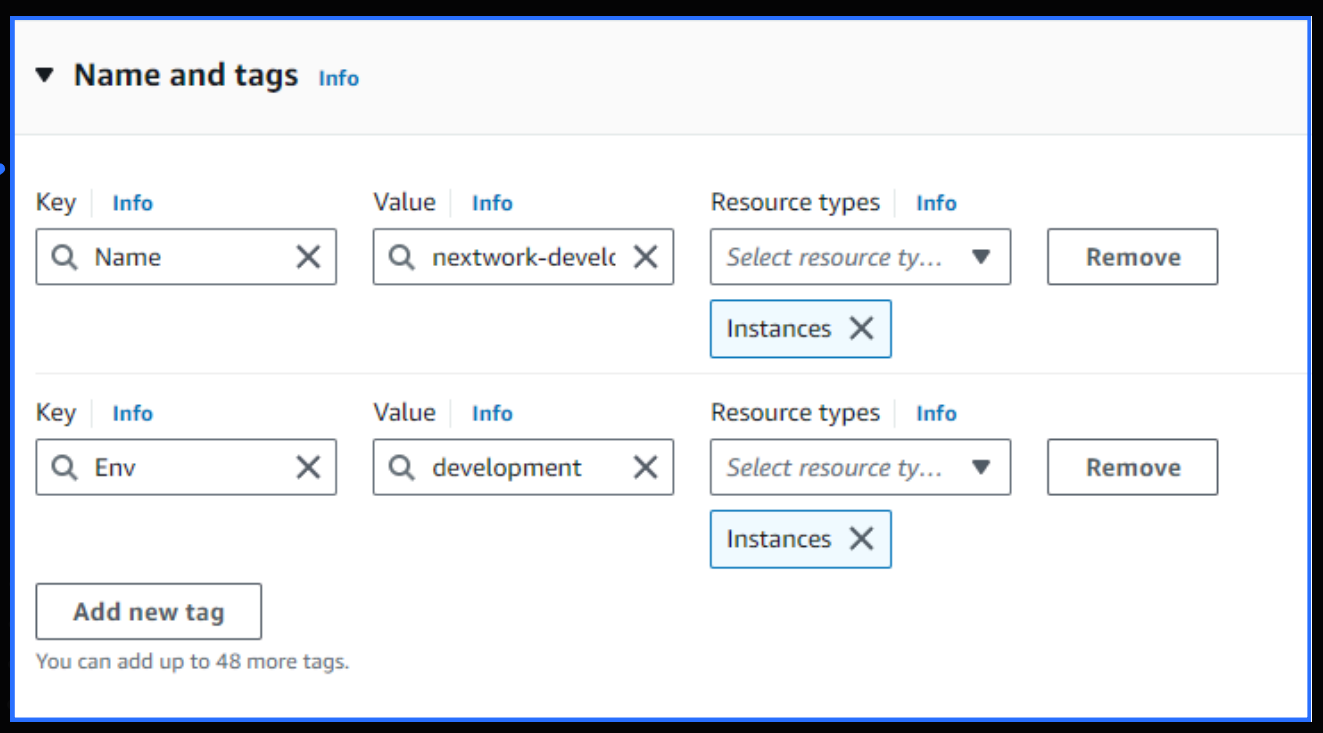
SETTING UP TAGS

- I've set up two EC2 instances to test the effectiveness of the permission settings I'll set up in AWS IAM. I've used **tags** to label them.

- Tags are labels to help AWS Account users identify and manage their resources. Tags are useful for grouping, mass management and applying security policies.

- The tag I've used on my EC2 instances is called Env. The value I've assigned for my instances are production, and development. This represents the two different environments that we use to build and release the app.

How the tags are set up for my EC2 instances



Kanika Mathur

@github.com/KanikaGenesis

IAM POLICIES

- **IAM Policies** are rules that help to applow/deny users'/ resources' permissions to perform certain actions to my AWS Account's resources.

- For this project, I've set up a policy using the JSON editor.

- I've created a Policy that allows all EC2-related actions to all EC2 instances that have the Environment ("Env") tag "development". But, it also denies creating and deleting tags for all EC2 instances.

- When writing JSON Policy statements, you have to specify the:
  - Effect: i,e. Allow or Deny
  - Action: i,e. the specific action that we are wanting to allow or deny.
  - Resource: the specific resources/group of resources in my AWS Account that this policy will take effect on.

The policy I've set up in the IAM Policies page!

Kanika Mathur
@github.com/KanikaGenesis

AWS ACCOUNT ALIAS

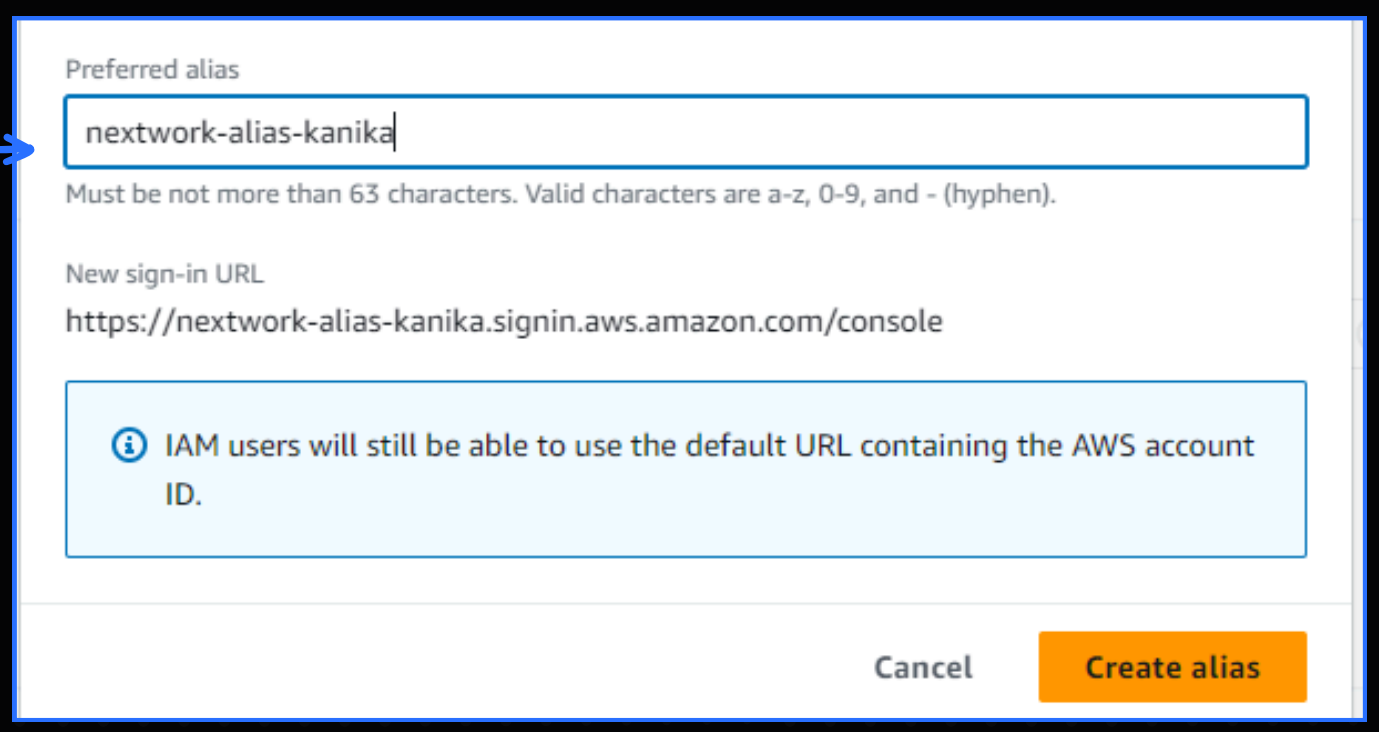
- When new users get onboarded onto my AWS account, they get access by signing into a unique URL created for my account's Account ID.

- An account alias is a custom name that I can assign to my AWS Account. This custom name would replace my Account ID in my Account's log-in URL.

- Creating an account alias took me less than a minute - super fast!

- Now, my new AWS console sign-in URL is https://**nextwork-alias-kanika**.signin.aws.amazon.com/console

You get to set up your own account alias name!



Kanika Mathur
@github.com/KanikaGenesis

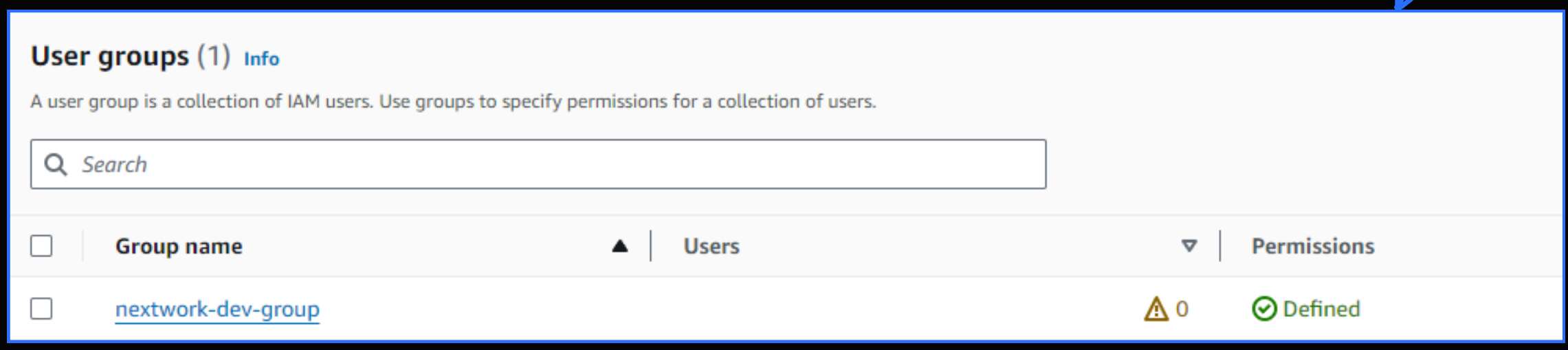
IAM USERS + USER GROUPS

- **IAM Users** are other log-ins/ people who have access to my AWS accounts and these users are created by myself using the AWS IAM service! I can designate my IAM users access to my AWS Account's resources/ services.

- I also created a **User Group**. User Groups are useful for grouping and managing users' permissions at a group level. They act similarly to folders when it comes to mass assigning permissions/ policies.

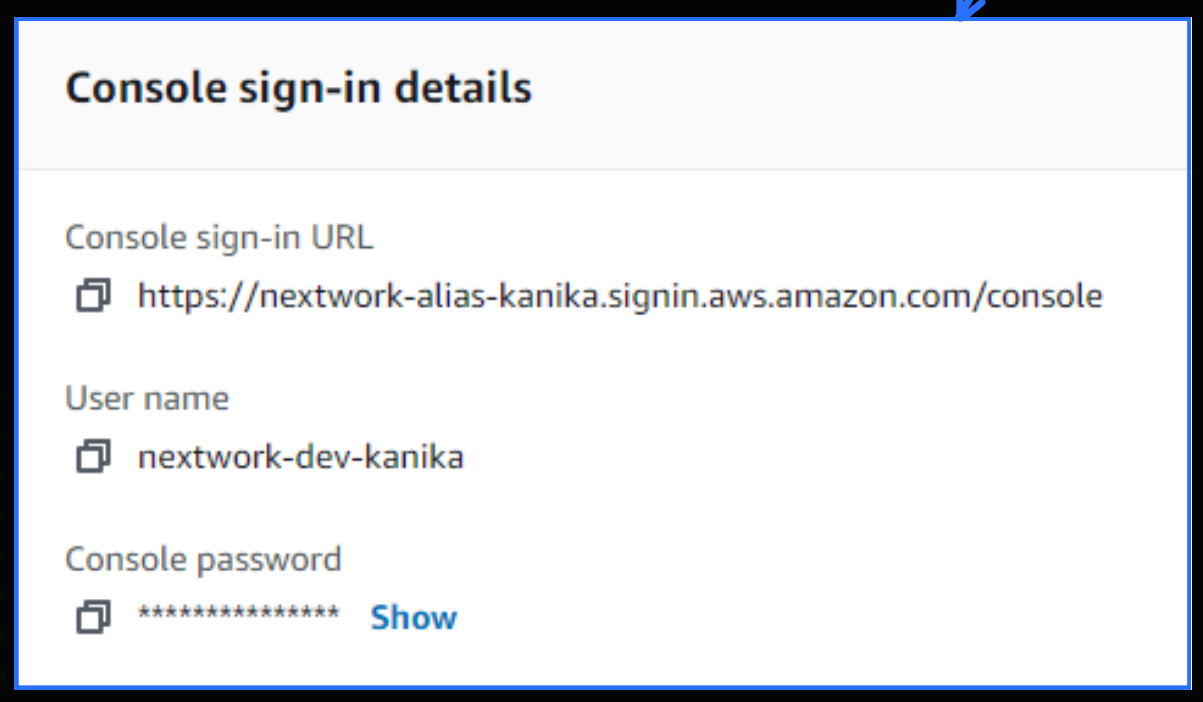
- My User Group is called... . I attached the Policy I created to this User Group, which means all users that are added to that user group will automatically inherit the user group's access permissions.

My User Group!



sign-in details!

- When I created a new User, I had to tick a checkbox that allowed the user access to the AWS Management Console.



- Once my new user was set up, there were two ways I could share its sign-in details: firstly, emailing sign-in instructions; secondly, downloading a .csv file.

- My new user had a unique sign-in URL - this is my Account Alias at work!

Kanika Mathur
@github.com/KanikaGenesis

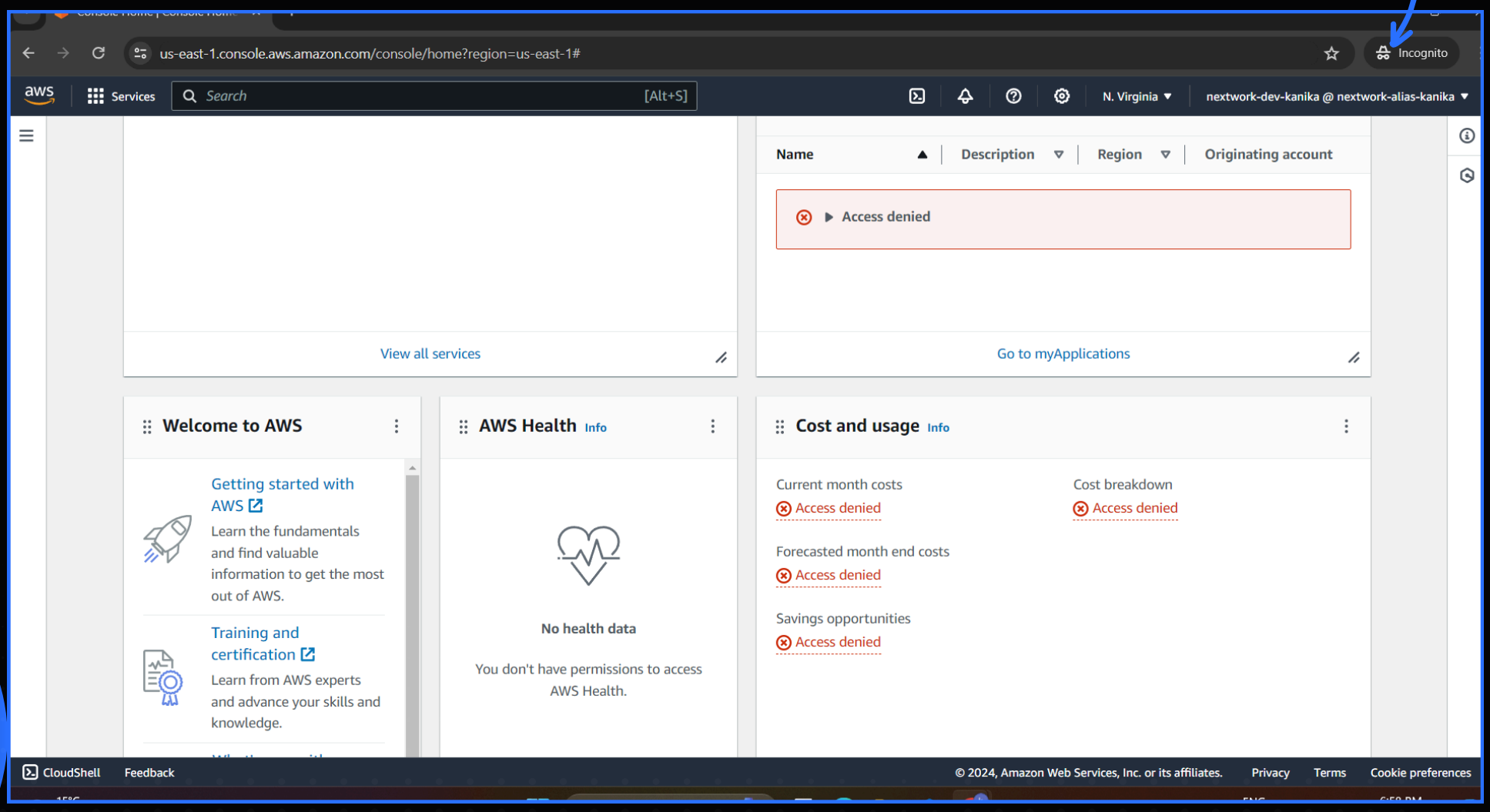
IAM USER IN ACTION

- Now with my IAM Policy, IAM User Group and IAM User all set up, let's put it all together! To do this, I logged into my AWS account as the new user.

- To log in as my IAM User, I had to access he console log-in URL that was given to me as I set up the new user.

- Once I logged in, I noticed that a lot of panels displayed "Access denied". This was a clear difference to the dashboard I usually see in my AWS Account (where I ahd unrestricted access to resources and wasn't denied access to anything.)

Some of my dashboard's panels showed access denied!



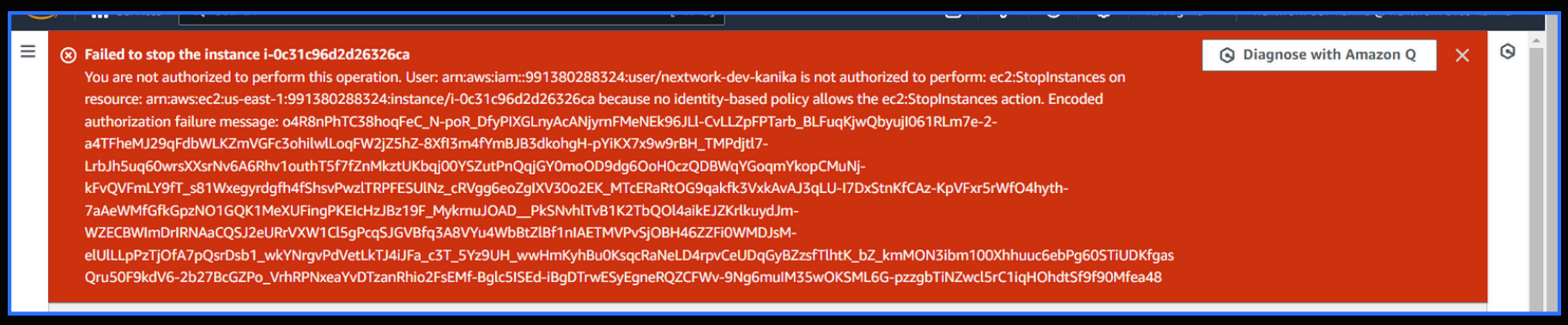
Kanika Mathur
@github.com/KanikaGenesis

IAM POLICIES IN ACTION

- Then, I tested the JSON IAM policy I set up by trying to Stop the development and production instances i,e. triggering the StopInstances action.

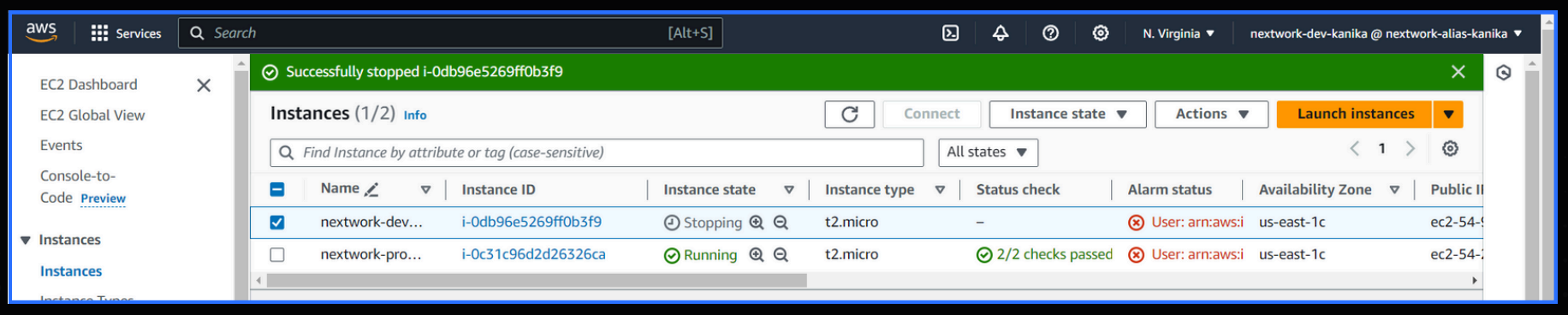
- When I tried to stop the production instance, an error message stopped me and explained that I am not authorised to stop the production instance.

Woah! An red fail banner pops up if I stop the production instance



- Next, when I tried to stop the development instance,the development instance could be stopped. This was because the policy I created (and attached to the User Group that my user is a part of) allowed all EC2 related actions to all EC2 instances/resources with the Env tag development.

Phew! An green success banner pops up if I stop the development instance



Kanika Mathur

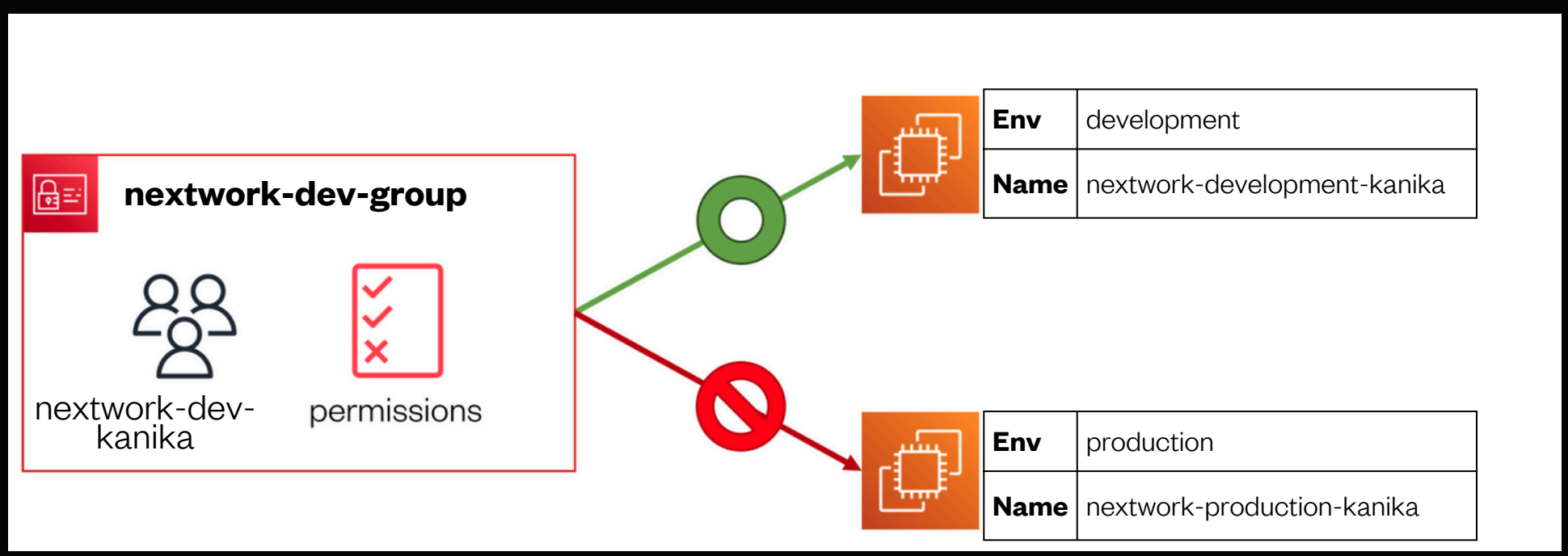
@github.com/KanikaGenesis

TO SUMMARISE

I created:

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

- My user could Stop the development instance, but could not stop the production instance. This is by design using my custom Policy.



Kanika Mathur
@github.com/KanikaGenesis

My Key Learnings

01

What are IAM Policies?

IAM Policies are JSON documents in AWS that define permissions for actions on AWS resources. They specify what actions are allowed or denied, on which resources, and under what conditions.

02

What are IAM Users? Why would you create one?

IAM Users are entities in AWS representing individuals or services that interact with AWS resources. They have unique credentials and permissions. You create IAM users to manage access securely, giving specific permissions to different users based on their roles and needs

03

What are IAM User Groups? Why would you create one?

IAM User Groups are collections of IAM users that simplify permissions management. By assigning permissions to a group, all users in that group inherit those permissions. This helps efficiently manage access for multiple users with similar roles.

04

What is an AWS Account Alias?

An AWS Account Alias is a user-friendly name that you can set for your AWS account to replace the default account ID in the AWS Management Console URL. It makes the URL easier to remember and recognize.

05

In making a custom policy you can specify what are the action that are allowed or denied and in which tags and values which give more specification for the policies.

Kanika Mathur

@github.com/KanikaGenesis

Final Thoughts...

- This project took me approximately one hour to complete.
- Delete EVERYTHING at the end! Let's keep this project free :)

- One thing I didn't expect was how deep you can go with the polices which is useful for security.

- Now that I know how IAM could be used to enhance security and permissions in my AWS account, some real-world use cases of what I've learnt are:
    - Creating different IAM Users for my personal AWS account to enhance security when I do personal projects that will enable public access to my account's resources.
    - Creating user groups for different company departments e.g. marketing, finance, development
    - Using an AWS Account Alias to create a user friendly console log in URL for a company's AWS account.

Kanika Mathur
@github.com/KanikaGenesis

Find this helpful?

- Like this post
- Leave a comment
- Save for later
- Let's connect!

Kanika Mathur
@github.com/KanikaGenesis

Thanks NextWork for the free project guide!

NEXTWORK