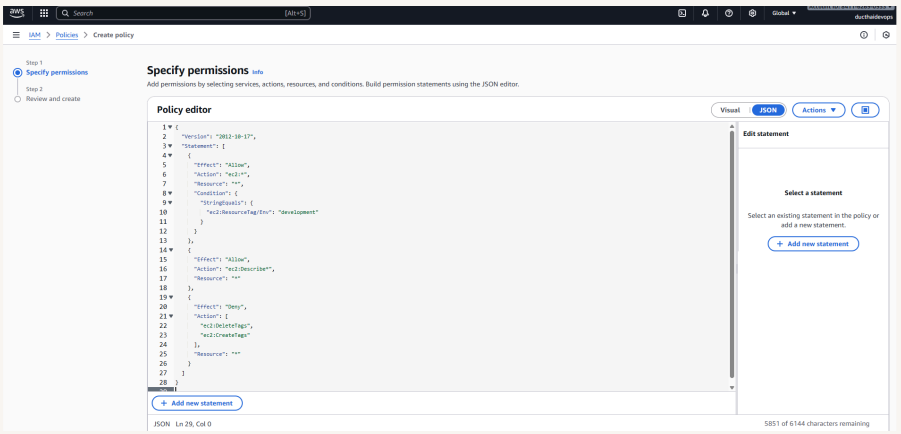
nextwork.org

Cloud Security with AWS IAM

D Duc Thai



Introducing Today's Project!

Today I will be using the AWS Identity and Access Management (IAM) service to control who is authenticated (signed in) and authorized (has permissions) in my AWS account.

Tools and concepts

Services I used were IAM policy: JSON rules that allow or deny actions on resources. Account alias: a friendly account name used in the sign-in URL. IAM users: individual identities with credentials.

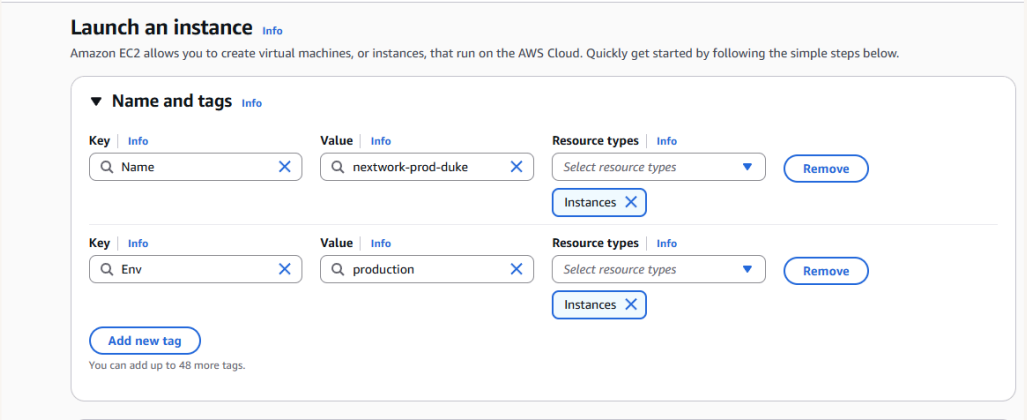
Project reflection

This project took me approximately 1 hours. The most challenging part was lauch EC2 instance, it need a subnet. It was most rewarding to create a subnet and know a little bit about it.

Tags

Tags are labels that you can attach to EC2 instance. 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.

The tag I’ve used on my EC2 instances is calle Name and Env. The value I’ve assigned for my instances are nextwork-prod-duke and production for production instacne and nextword-dev-duke and dev for development instance.



IAM Policies

IAM Policies are rules for who can do what with your AWS resourses.It's all about giving permissions to IAM users, groups, or roles, saying what they can or can't do on certain resources, and when those rules kick in.

The policy I set up

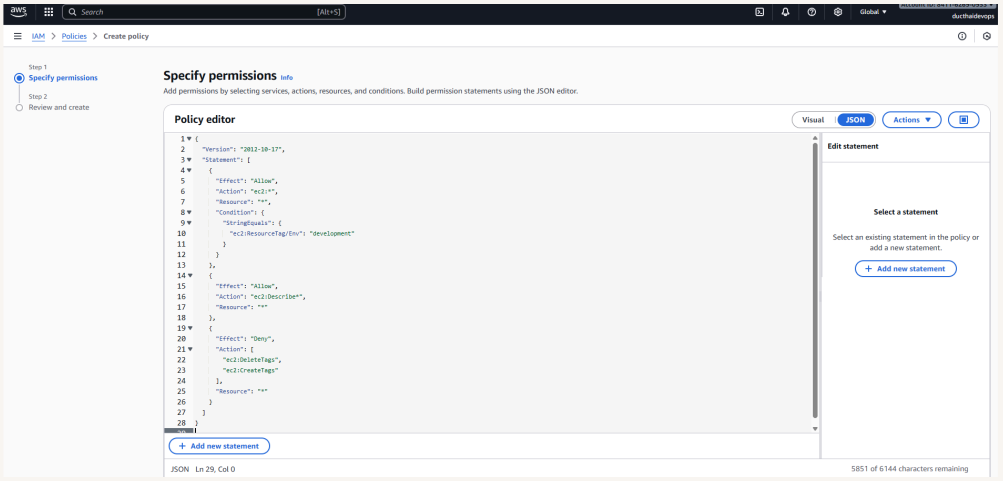
For this project, I've set up a policy using JSON.

I've created a policy that allows all actions that take on EC2 instance that has the Env - development tags. Allow the describe action and all ec2 instances and deny all the delete and create actions to all the ec2 instances.

When creating a JSON policy, you have to define its Effect, Action and Resource.

Effect: whether the policy Allow(s) or Deny(s) the request. Action: which operations or API calls (e.g., ec2:StartInstances) the statement covers. Resource: which objects (ARNs or "*"for all) the rule applies to.

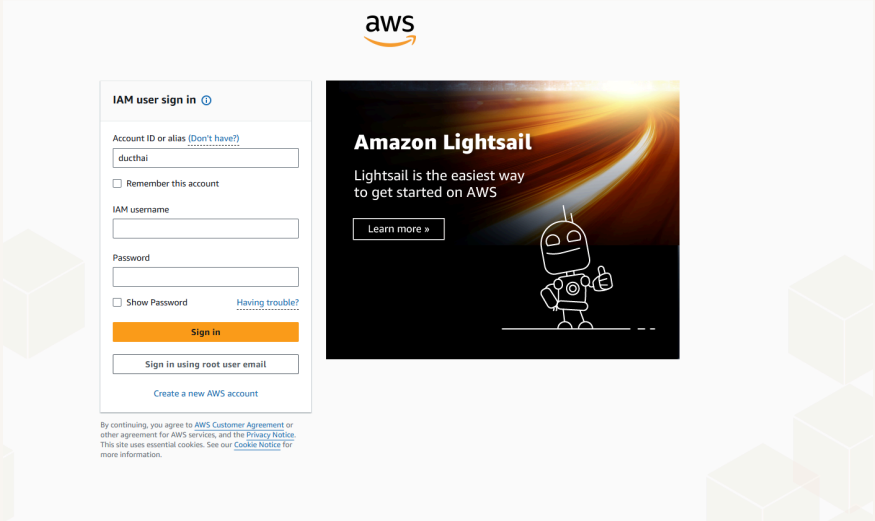
My JSON Policy



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.

Creating an account alias took me one minute Now, my new AWS console sign-in URL is https://ducthai.signin.aws.amazon.com/console



IAM Users and User Groups

Users

IAM users are AWS identities for people or services. Each user can have login credentials and is granted permissions via policies or groups to control which AWS resoures.

User Groups

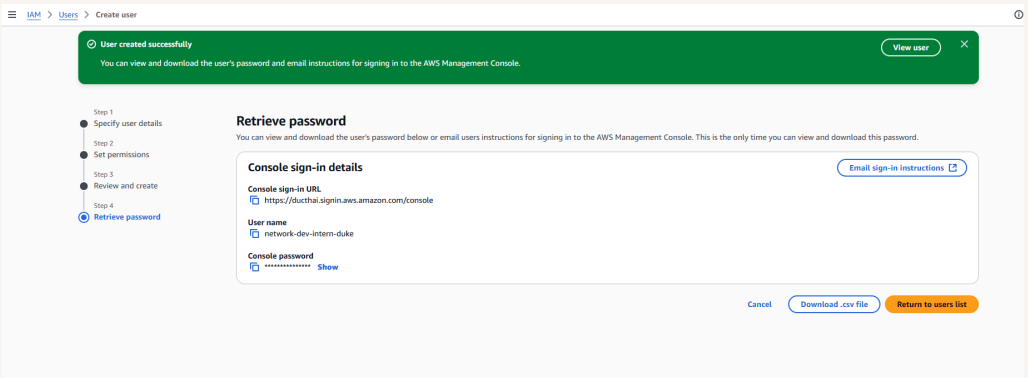
An IAM user group is a collection/folder of IAM users. It allows you to manage permissions for all the users in your group at the same time by attaching policies to the group rather than individual users.

I attached the policy I created to this user group, which means all user in the group have the similar access to AWS resourses.

Logging in as an IAM User

The first way is thorugh console with sign in url and password. Other way is programmactically through CLI/SDK use.

Once I logged in as my IAM user, I noticed my history, application, cost management is disabled. This was because AWS treat me as new user and I do not have access to all the resources.



Testing IAM Policies

I tested my JSON IAM policy by try stopping noth prod and dev instances. I failed to stop the prod instance but I was able to stop the development one.

Stopping the production instance

When I tried to stop the production instance an angry-looking banner tells us we've failed to stop this instance and I am not authorized This was because my IAM pocilies does not allow me to stop any instance at all.

Testing IAM Policies

Stopping the development instance

Next, when I tried to stop the development instance I was sucessful. This was because the IAM pocilies what attched to me allows me to any action to development instance.

The place to learn & showcase your skills

Check out nextwork.org for more projects