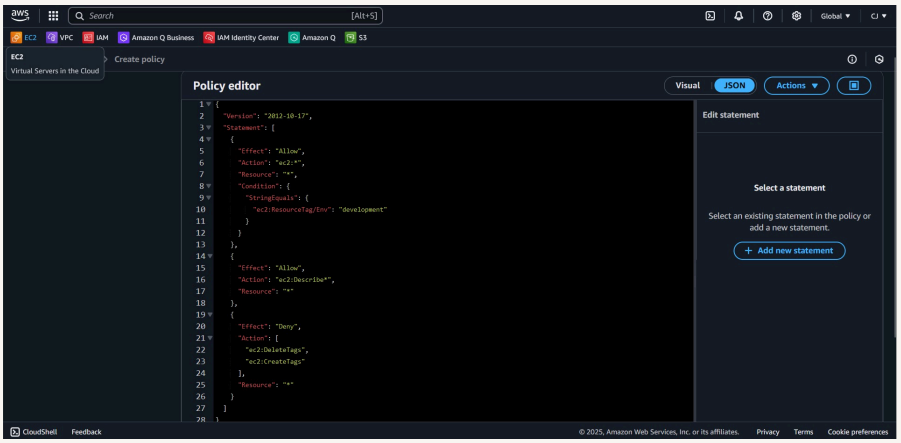
nextwork.org

Cloud Security with AWS IAM

Chrispinus Jacob



Chrispinus Jacob
NextWork Student

Introducing Today's Project!

In this project, I will demonstrate how to use AWS Identity and Access Management (IAM) to control who is authenticated (signed in) and authorized (has the necessary permissions) in an AWS account. The goal of this project is to learn how to securely manage access to AWS resources by creating and managing users, groups, roles, and policies.

Tools and concepts

Got it — here's your **wrap-up in one clear line**: **Services I used were IAM, EC2, and the IAM Policy Simulator, and key concepts I learnt include creating users and groups, writing JSON policies with Effect, Action, and Resource, using tags and conditions to control access, and applying the principle of least privilege to protect production resources.** □ Short, neat, and ready for your report!

Project reflection

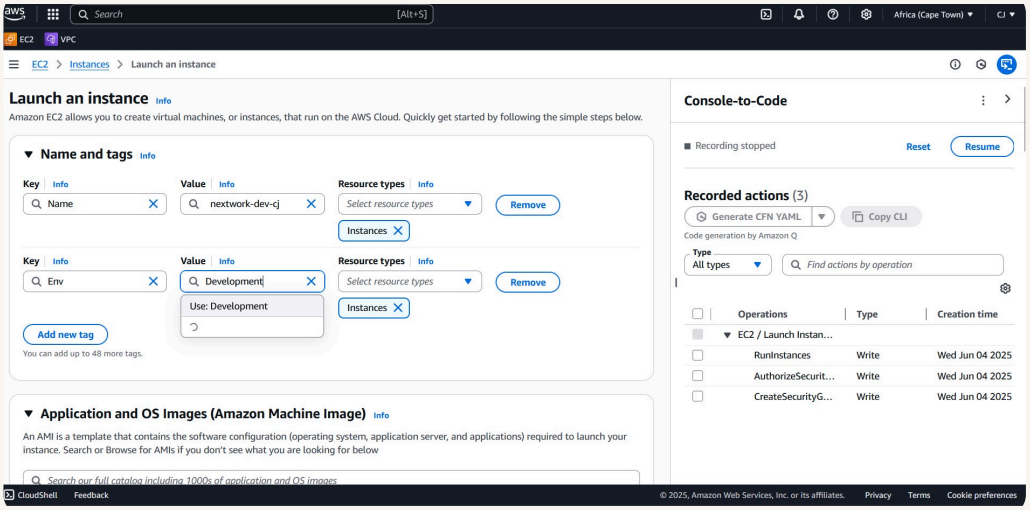
This project took me approximately 1 hour to complete. The most challenging part was making sure the IAM policy conditions were correct so that the intern could only access development instances and not production. It was most rewarding to see the policy work as expected in the IAM Policy Simulator and to learn how to safely test permissions without risking real resources.

Chrispinus Jacob
NextWork Student

Tags

Tags are like labels you can attach to AWS resources for organization. tagging helps us with identifying all resources with the same tag at once (they are useful filters when you're searching for something)

The tag I've used on my EC2 instances is called ENV The value I've assigned for my instances are Production and Development



Chrispinus Jacob
NextWork Student

IAM Policies

IAM policies are a set of rules that define who can do what in an AWS account. They specify which actions are allowed or denied on which AWS resources. By attaching policies to users, groups, or roles, you control what permissions those identities have, ensuring secure and organized access management.

The policy I set up

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

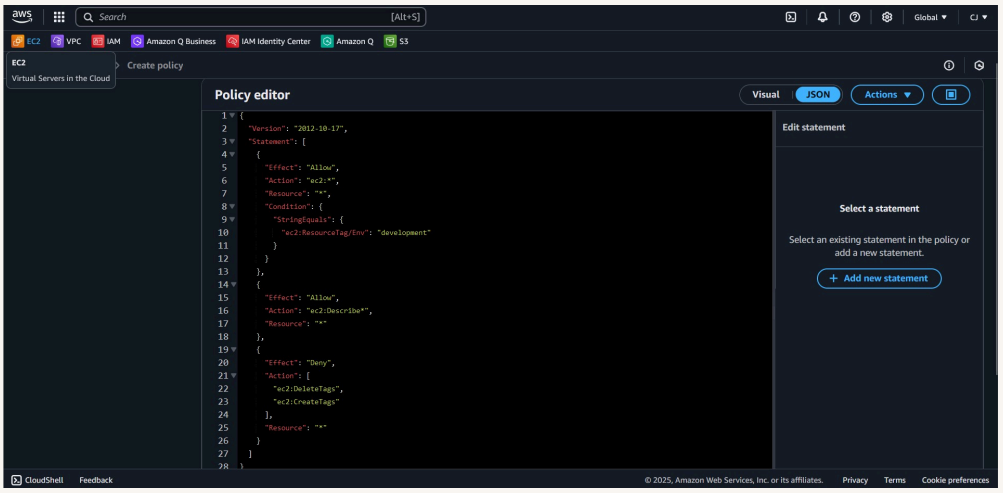
I have created an IAM policy that allows the policy holder (for example, the Nextwork intern) to perform any action on EC2 instances that are tagged with Environment = Development. This means they can start, stop, reboot, or modify development instances freely. For all other EC2 instances (for example, production instances), they can only view information (such as describing instances) but cannot perform any operations like starting, stopping, or modifying them.

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

The Effect, Action, and Resource attributes of a JSON policy means...whether or not the policy is allowing/denying the action. (effect) , what policy holder can /not do . (Action) . AWS Resources that the policy holder relates to. (Resource)

Chrispinus Jacob
NextWork Student

My JSON Policy

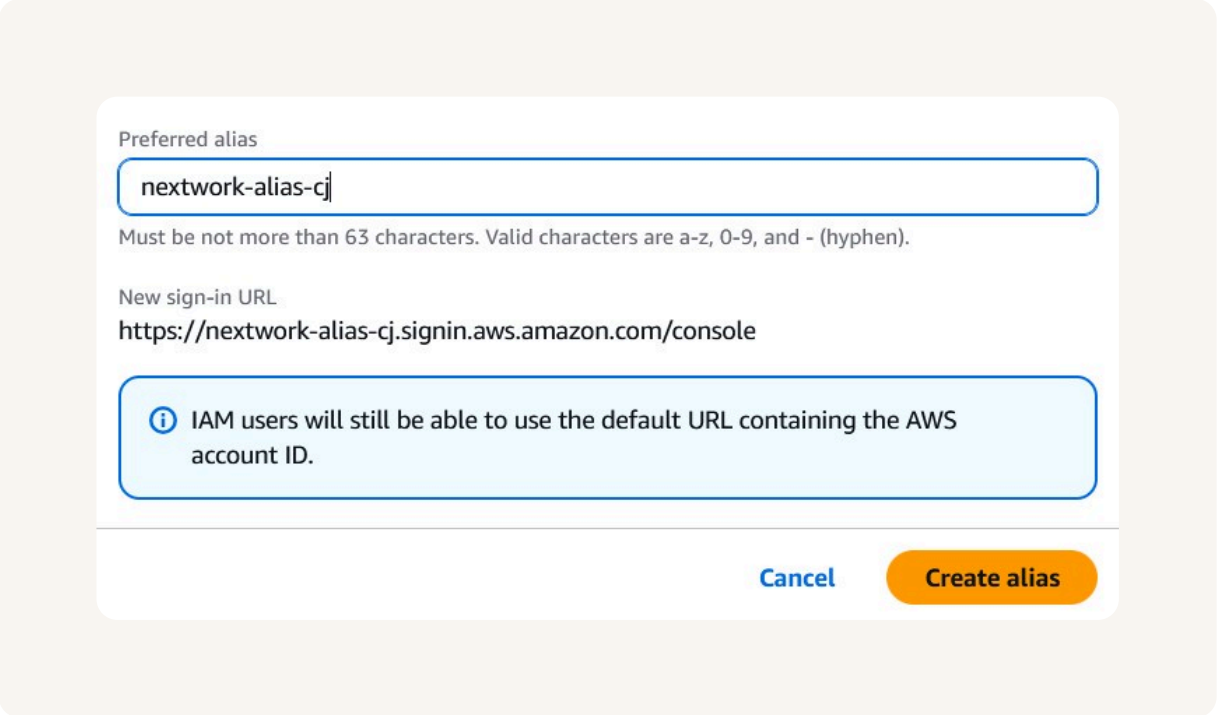


Chrispinus Jacob
NextWork Student

Account Alias

An account alias is simply nickname for aws account instead of the long name

Creating an account alias took me... 30 secs Now, my new AWS console sign-in URL is... uses alias instead of my account id



Chrispinus Jacob
NextWork Student

IAM Users and User Groups

Users

IAM users are are individual identities created within AWS IAM to represent a person or application that interacts with AWS resources.

User Groups

IAM user groups are like folders that collect multiple IAM users together. This allows you to apply permissions at the group level instead of managing permissions for each user one by one. Any permissions you attach to a group automatically apply to all users in that group.

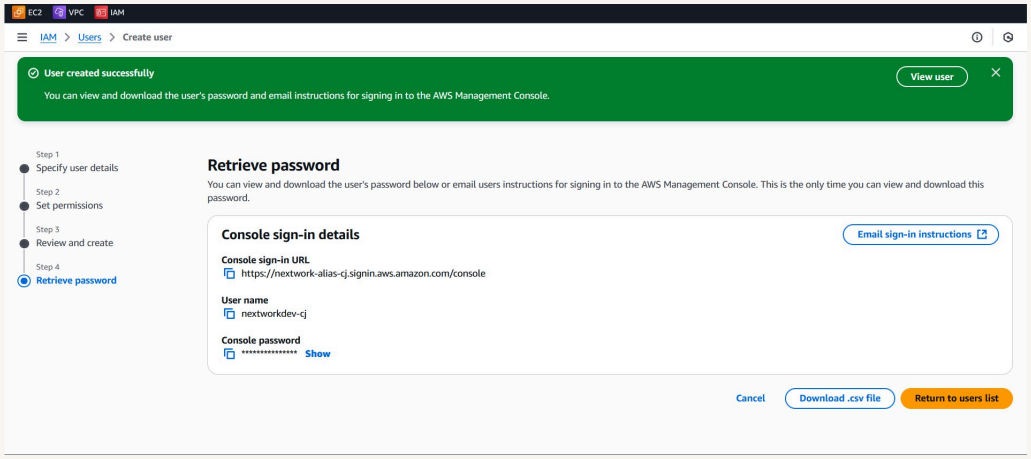
I attached the IAM policy I created (which allows full access to development instances and read-only access to others) to the Dev user group. This means any IAM user added to the Dev group will automatically get the same level of access defined by the policy. So, all developers in this group can work on development resources but cannot make changes in the production environment.

Chrispinus Jacob
NextWork Student

Logging in as an IAM User

The first way is...AWS provides user sign-in link, username, and (optionally) a temporary password after creation. You can manually copy and send

Once I logged in as the intern IAM user, I noticed that the user could not see some items or panels in the AWS Management Console. This is because we only granted permissions to manage development instances. The IAM user does not have permissions for other AWS services or resources, so those parts of the console are hidden or inaccessible.



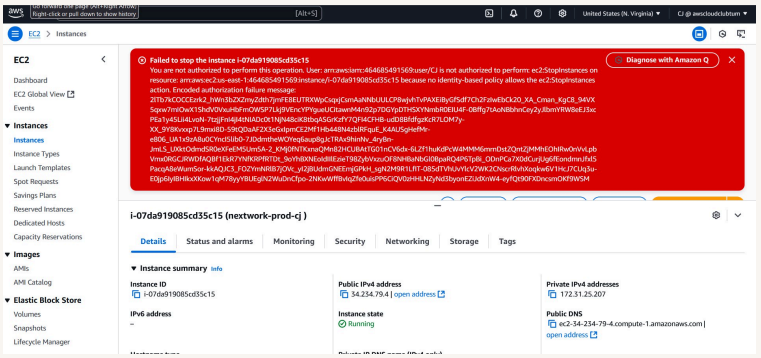
Chrispinus Jacob
NextWork Student

Testing IAM Policies

I tested my JSON IAM policy by logging in as the intern IAM user and trying to perform actions on both development and production EC2 instances. I confirmed that the intern could start, stop, and manage instances tagged as Development but could only view production instances and could not stop or modify them. When I attempted to stop a production instance, I received an error message as expected — proving that the policy is correctly blocking unauthorized actions.

Stopping the production instance

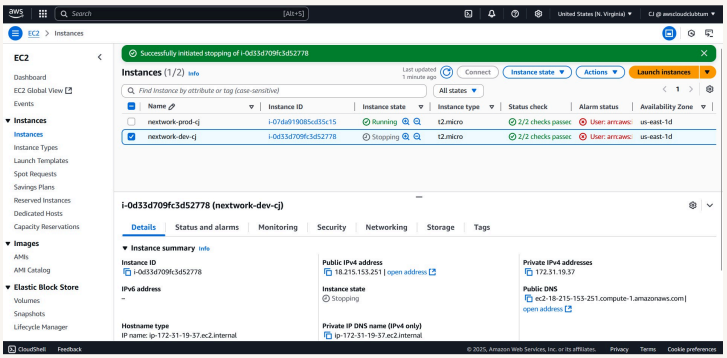
When I tried to stop the production instance while logged in as the intern IAM user, I received an error message. This happened because the IAM policy attached to the intern explicitly denies any actions on resources tagged as production. The intern is only allowed to manage development instances and has no permissions to perform actions (like starting, stopping, or modifying) on production resources.



Testing IAM Policies

Stopping the development instance

Next, when I tried to stop the development instance while logged in as the intern IAM user, the action succeeded as expected. This was because the IAM policy explicitly allows full control (ec2:*) for instances tagged with Environment = Development. This confirms that the user has the correct permissions to manage development resources, as intended.



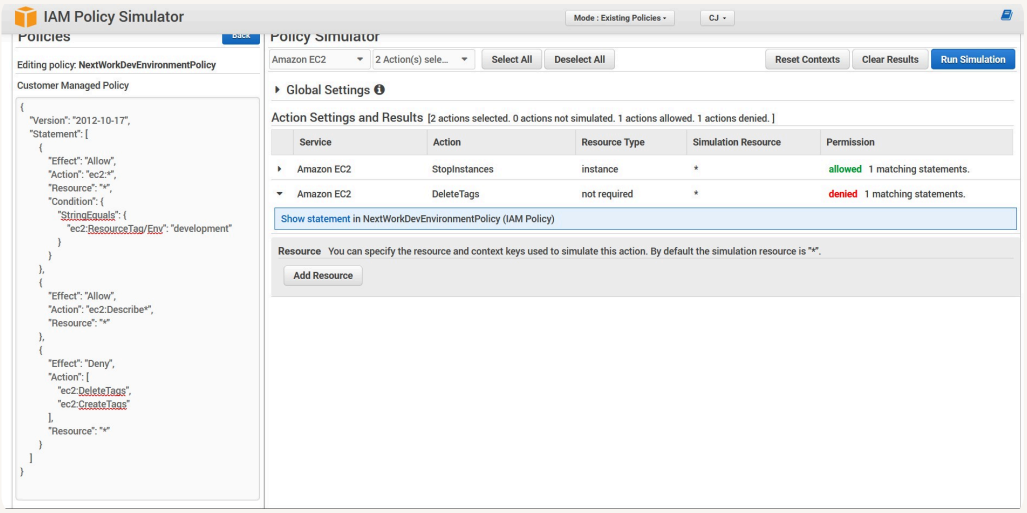
Chrispinus Jacob
NextWork Student

The IAM Policy Simulator

The IAM Policy Simulator is a tool provided by AWS that lets you test and troubleshoot IAM policies without actually performing actions on real AWS resources.

How I used the simulator

I set up a simulation for my intern IAM user to test whether they could delete tags and stop instances for both development and production EC2 instances, and the results showed that the user was allowed to stop instances in the development environment but denied from stopping production instances or deleting tags on either environment.



The place to learn & showcase your skills

Check out nextwork.org for more projects