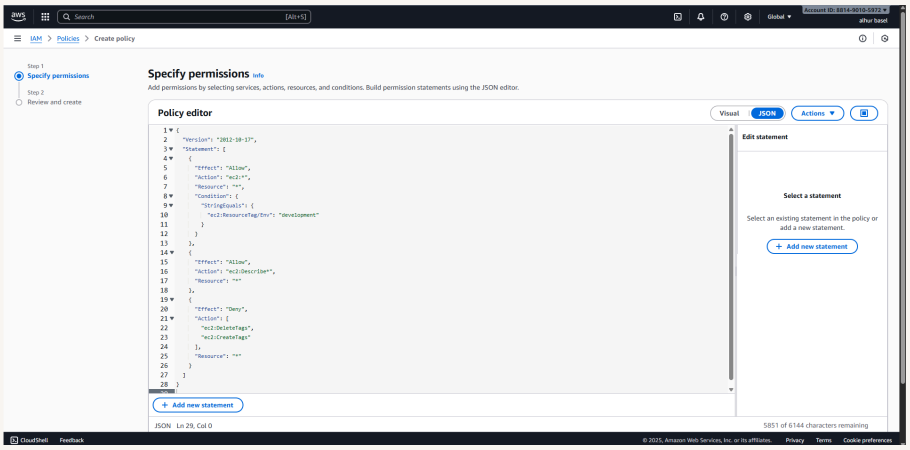
nextwork.org

Cloud Security with AWS IAM

al7r



Introducing Today's Project!

In this project, I will demonstrate how to create and manage AWS Identity and Access Management (IAM) users, groups, and policies, and how to launch an EC2 instance. I'm doing this project to learn how to control authentication and authorization in AWS, manage permissions securely, and gain hands-on experience with core AWS services that are essential for cloud computing, security, and DevOps.

Tools and concepts

I learnt EC2 for launching virtual servers, IAM for creating users, groups, and policies, tags to organize resources, account aliases for easier logins, and how to control permissions to safely separate development and production environments.

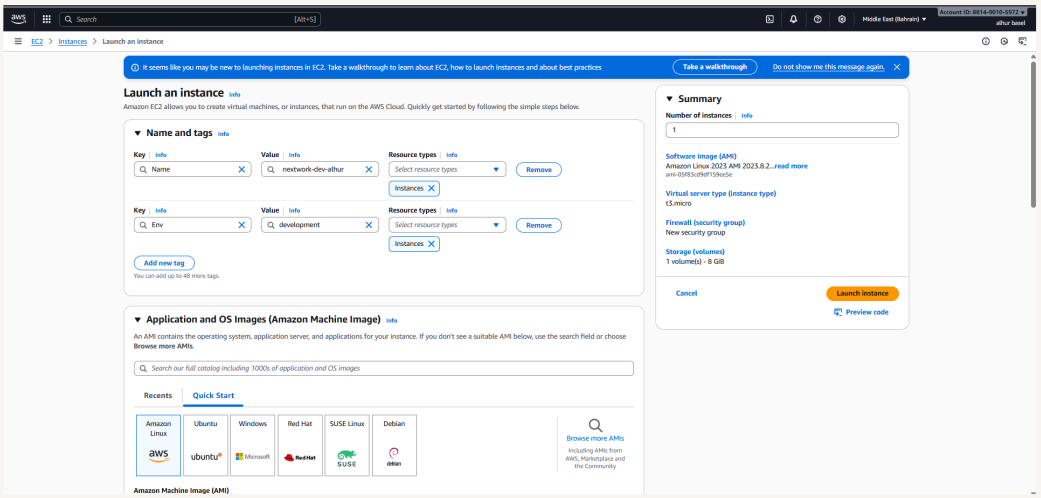
Project reflection

This project took me approximately 1 hour. The most challenging part was setting up the IAM policy correctly to restrict access to development while denying production. It was most rewarding to test the intern's access and see that the policy worked exactly as intended.

Tags

Tags are labels attached to AWS resources. They help organize, filter, and identify resources, track costs, and apply policies based on environment or purpose.

The tag I've used on my EC2 instances is called Env. The value I've assigned for my instances is production for the first instance and for the second instance.



IAM Policies

IAM Policies are rules that define who can access AWS resources and what actions they can perform. They control permissions for users, groups, or roles to ensure resources are used securely and appropriately.

The policy I set up

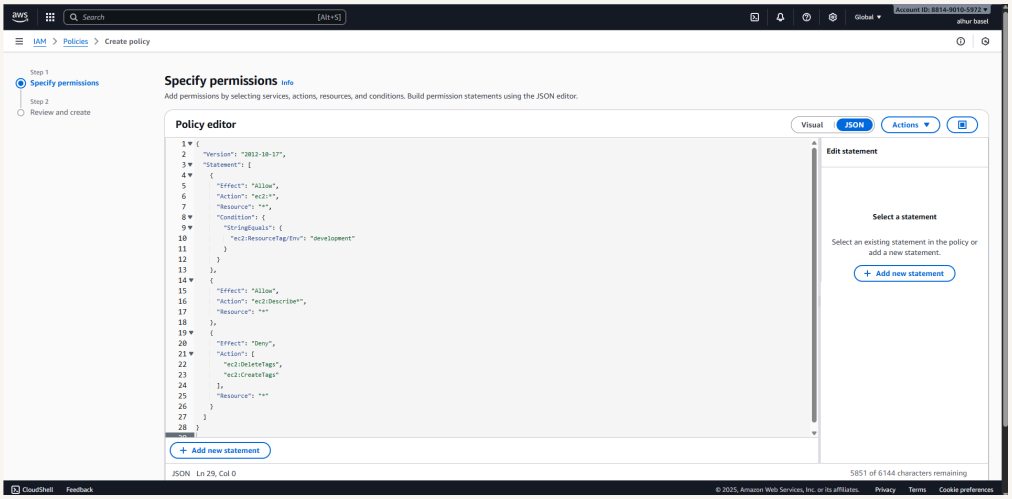
I set up the policy using JSON to define precise permissions for the access to the development EC2 instance.

The effect of my policy is to allow the intern to manage only development EC2 instances while denying any actions that could affect production instances or tags, keeping resources secure.

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

Effect specifies whether the policy allows or denies an action. Action lists the operations the policy controls, like starting or stopping instances. Resource defines which AWS resources the policy applies to, such as specific EC2 instances or all resources.

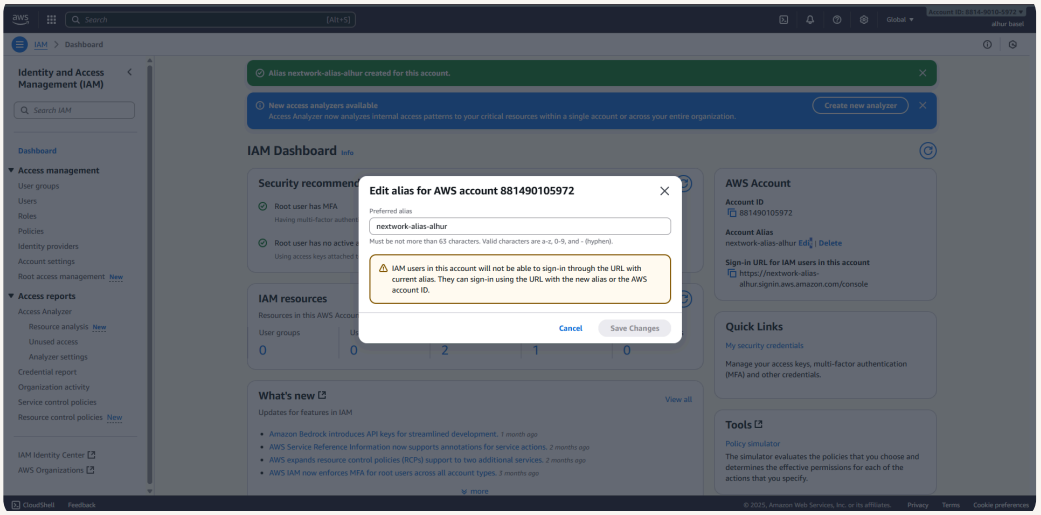
My JSON Policy



Account Alias

An account alias is a friendly name for your AWS account that replaces the long numeric account ID in the login URL, making it easier to remember and share with team members.

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



IAM Users and User Groups

Users

IAM users are individual accounts created in AWS for people or applications, giving them a secure way to log in and access resources with permissions defined by policies or groups.

User Groups

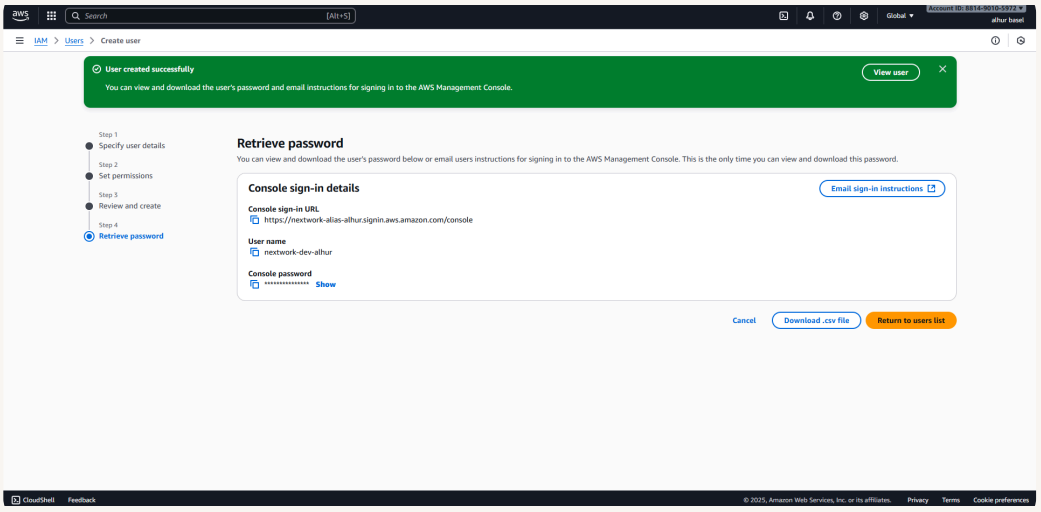
IAM user groups are collections of IAM users that let you manage permissions for multiple users at once by attaching policies to the group instead of each user individually.

Attaching a policy to my user group applies the permissions in that policy to all users in the group, ensuring they can access only the resources allowed, like the development EC2 instance, without managing each user individually.

Logging in as an IAM User

You can share a new user's sign-in details by sending the login URL, username, and temporary password directly, or by having them set up their own password through an email invitation from AWS.

Once I logged in as my IAM user, I noticed that some dashboard panels showed Access Denied. This was because the user only has permissions assigned through the NextWorkDevEnvironmentPolicy and cannot access production resources.

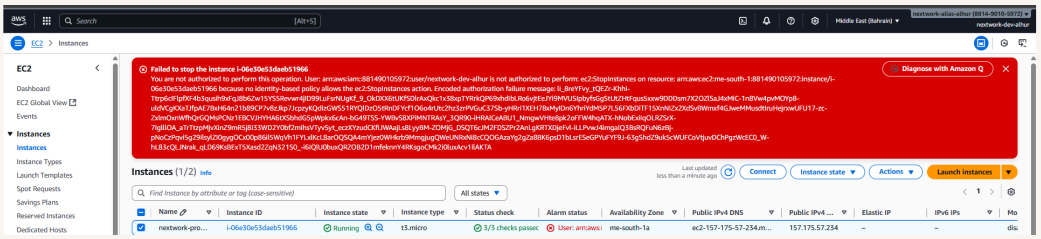


Testing IAM Policies

I tested my JSON IAM policy by trying to stop both EC2 instances. I was denied access for the production instance but was able to stop the development instance successfully.

Stopping the production instance

When I tried to stop the production instance, I received an Access Denied error. This was because my IAM user only has permissions for development instances and cannot perform actions on production resources.



Testing IAM Policies

Stopping the development instance

Next, when I tried to stop the development instance, it stopped successfully. This was because my IAM user has permissions for instances tagged with Env = development.