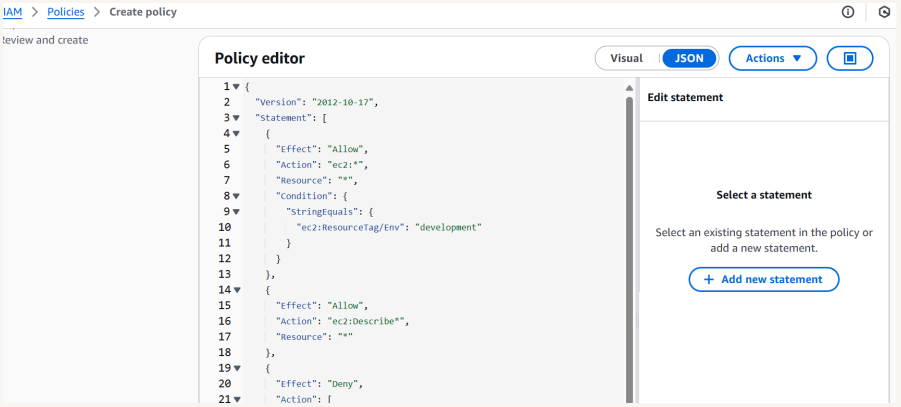
nextwork.org

Cloud Security with AWS IAM

Odinaka Dialah



Introducing Today's Project!

Today, i am here to learn how to manage EC2 instances, create and apply IAM policies, set up IAM users and user groups, and configure an AWS account alias. These steps help control access and manage AWS resources securely and efficiently.

Tools and concepts

In this project, I learned key AWS services and concepts like EC2 for computing power, IAM for managing users, groups, and permissions, and how to use tags and policies to control access. I also learned how to create an Account Alias for easier login

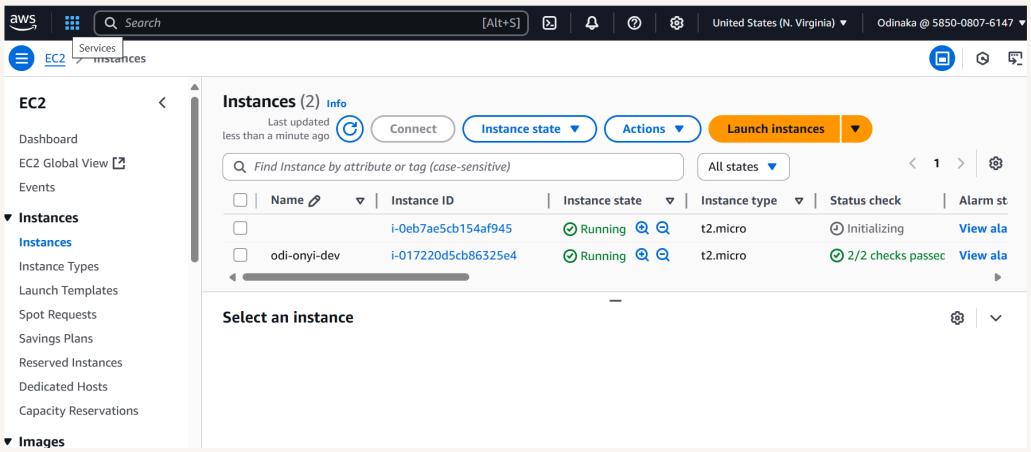
Project reflection

This project took about 1 to 2 hours to complete, including setting up EC2 instances, IAM users and groups, creating and testing policies, and configuring the account alias. The time may vary based on familiarity with AWS.

Tags

Tags in AWS are labels in the form of key-value pairs that you assign to resources like EC2 instances. They help organize, identify, and manage resources by project, team, or environment, and are useful for cost tracking, automation, and access c

For the two EC2 instances, I assigned the following tags and values: Name: odi-onyi-dev / onyi-odi2 Environment: Development/ Production Owner: Odinaka These tags help with identification, organization, and cost tracking.



Odinaka Dialah
NextWork Student

IAM Policies

IAM policies define permissions in AWS. They control what actions users, groups, or roles can perform on specific resources. For example, a policy can allow a user to start EC2 instances or deny access to S3 buckets. They help secure and manage acces

The policy I set up

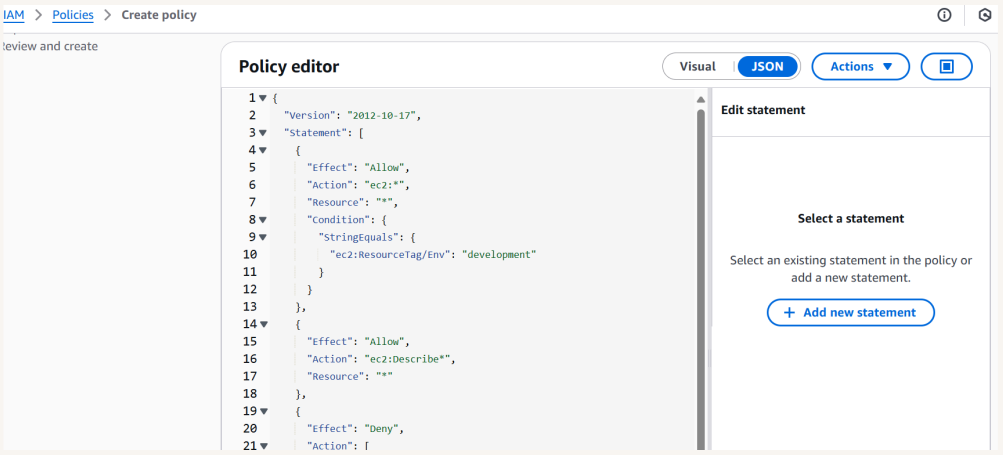
I set up the policy using JSON. This method gives more control and flexibility, allowing precise permission settings—like granting access based on tags such as "Environment": "Development".

The effect of the policy is to allow the user to start, stop, reboot, and describe EC2 instances only if they are tagged with "Environment": "Development". This ensures access is limited to development resources for security and organization.

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

In a JSON IAM policy: Effect: Defines whether the action is allowed or denied. It can be "Allow" or "Deny". Action: Specifies what AWS operations are permitted (e.g., ec2:StartInstances). Resource: Indicates the specific AWS resources the actions

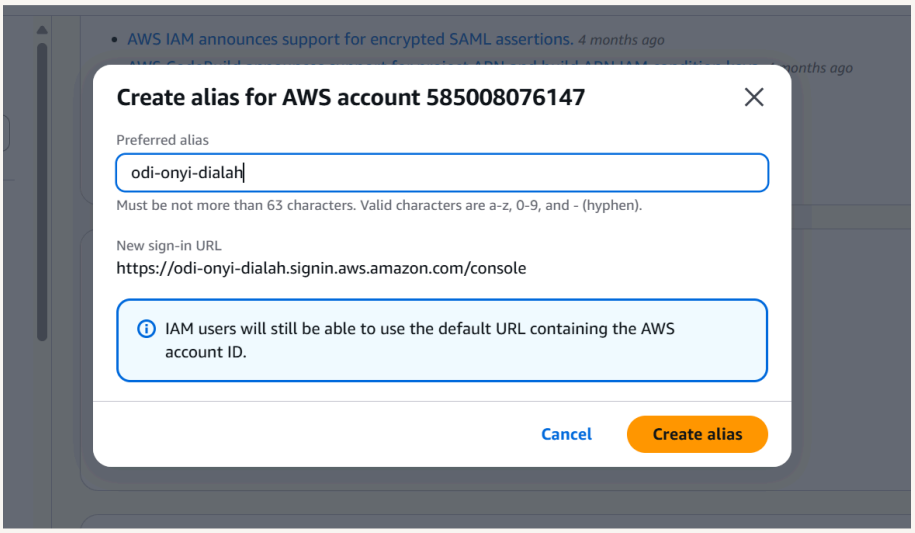
My JSON Policy



Account Alias

An Account Alias is a custom name you create to replace your AWS account ID in the sign-in URL. It makes logging in easier and more user-friendly. For example, instead of a long number, users can go to https://nextwork.awsapps.com.

Setting up an Account Alias in AWS takes just a few minutes. You simply go to the IAM dashboard, choose "Account Alias," enter your custom name, and save it. It's quick and easy to do.



IAM Users and User Groups

Users

IAM users are individual identities in AWS with their own login credentials. Each user represents a person or application and can be given specific permissions to access AWS services. This helps manage access securely and separately.

User Groups

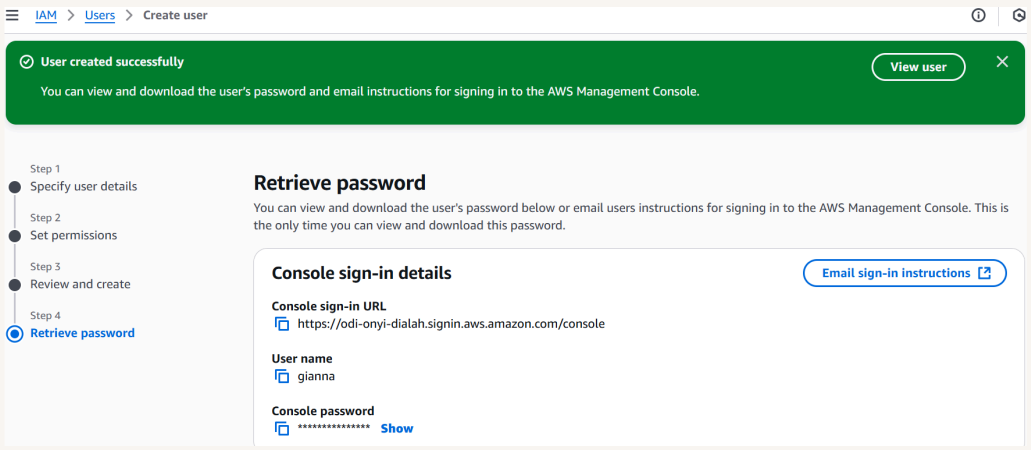
IAM user groups are collections of IAM users that share the same permissions. Instead of assigning policies to each user individually, you attach policies to the group,making it easier to manage access for teams like interns, developers, or admins.

Attaching a policy to a user group gives all users in that group the same permissions defined by the policy. For example, if the policy allows EC2 access, every intern in the group can manage EC2 instances based on that access level.

Logging in as an IAM User

I can share a new IAM user's sign-in details in two ways: 1 Email the credentials directly from the IAM setup page. 2 Download the CSV file with their login URL, username, and password, then share it securely (e.g. encrypted email or file transfer)

As a new user, I noticed that some of your dashboard panel are showing Access denied.This is because,It means your permissions are limited by the IAM policies assigned. This ensures users only access the resources needed, keeping the AWs account safe

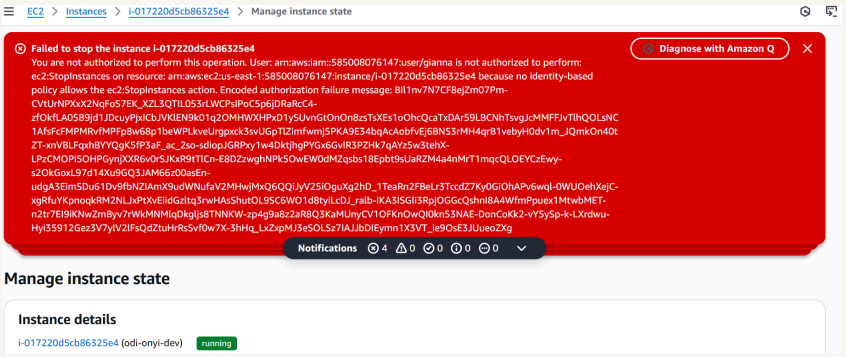


Testing IAM Policies

I tested access by attempting to stop both EC2 instances. The intern IAM user was able to stop the development instance but received an "Access Denied" error when trying to stop the production instance, as expected based on the applied IAM policy.

Stopping the production instance

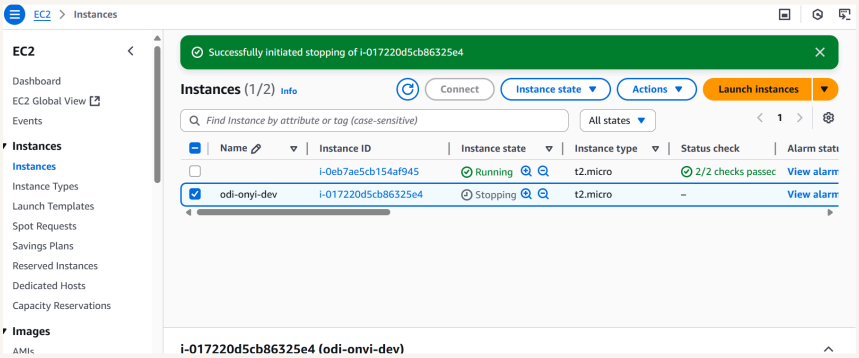
When I tried to stop the production instance, I received a banner top of my page, an angry-looking banner tells us we've failed to stop this instance. The banner tells us it's because we're not authorized! We don't have permission to stop



Testing IAM Policies

Stopping the development instance

When I tried to stop the development instance, the action was successful. The IAM policy allowed it because the instance was tagged with "Environment": "Development", matching the access rules set for the intern user.



The place to learn & showcase your skills

Check out nextwork.org for more projects