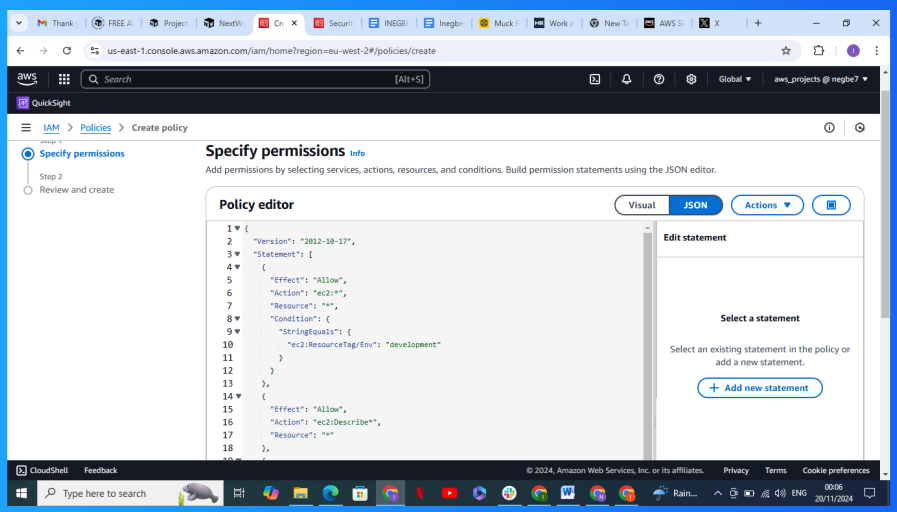
NextWork.org

Cloud Security with AWS IAM

NE negbenosepierre@gmail.com



NE
negbenosepierre@gmail....
NextWork Student
NextWork.org

Introducing today's project!

What is AWS IAM?

AWS IAM (Identity and Access Management) is a service that helps you securely manage access to AWS resources. It allows you to define who can access what resources and under what conditions, ensuring secure and controlled access.

How I'm using AWS IAM in this project

In today's project, I used AWS IAM to create policies, user groups, and users. I assigned permissions to restrict access to resources like EC2 instances, ensuring only authorized actions were allowed in the development and production environments.

One thing I didn't expect...

the level of ease

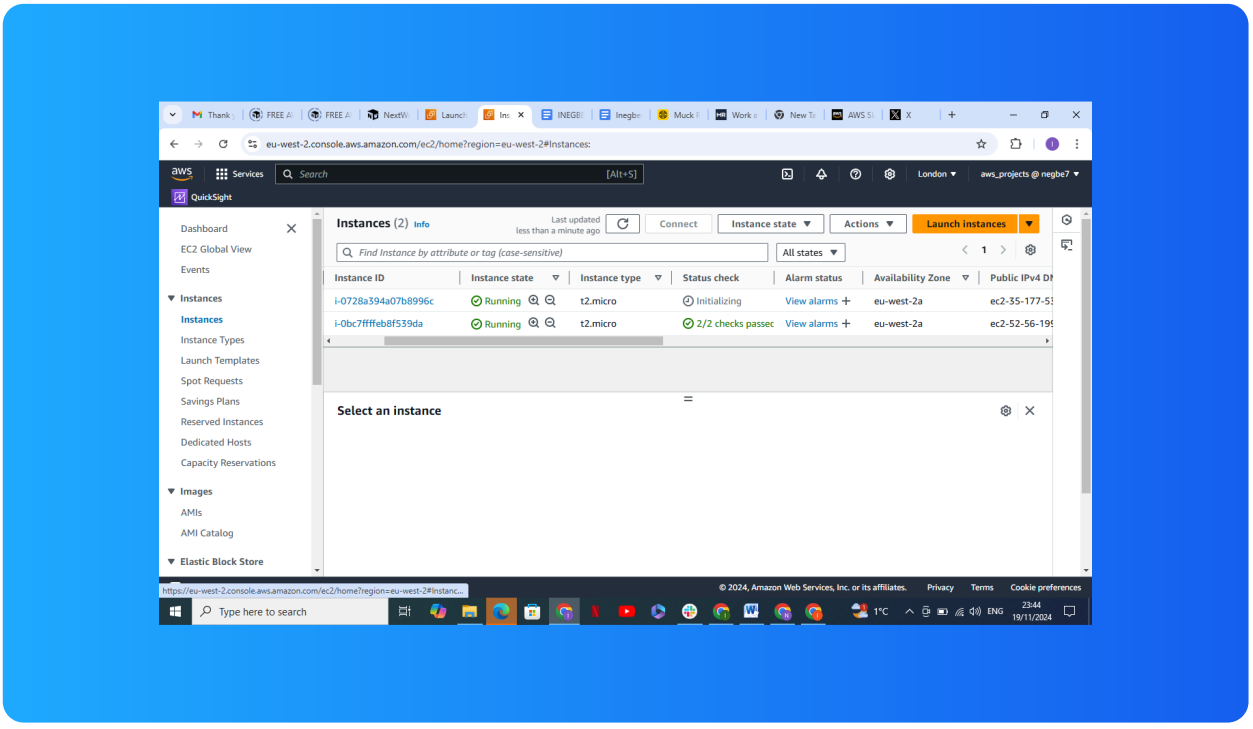
This project took me...

1 hour

Tags

Tags are metadata labels assigned to AWS resources like EC2 instances, S3 buckets, or RDS databases. They consist of key-value pairs (e.g., Key: Env, Value: Development) that help you organize, manage, and filter resources effectively.

The tag I've used on my EC2 instances is called Env. The values I've assigned for my instances are Production and Development, which reflect their roles in the software development life cycle.



IAM Policies

IAM Policies are rules that define permissions for actions on AWS resources. They determine who can access what, specifying actions, resources, and conditions. For example, a policy might allow users to start EC2 instances but deny them the ability t

The policy I set up

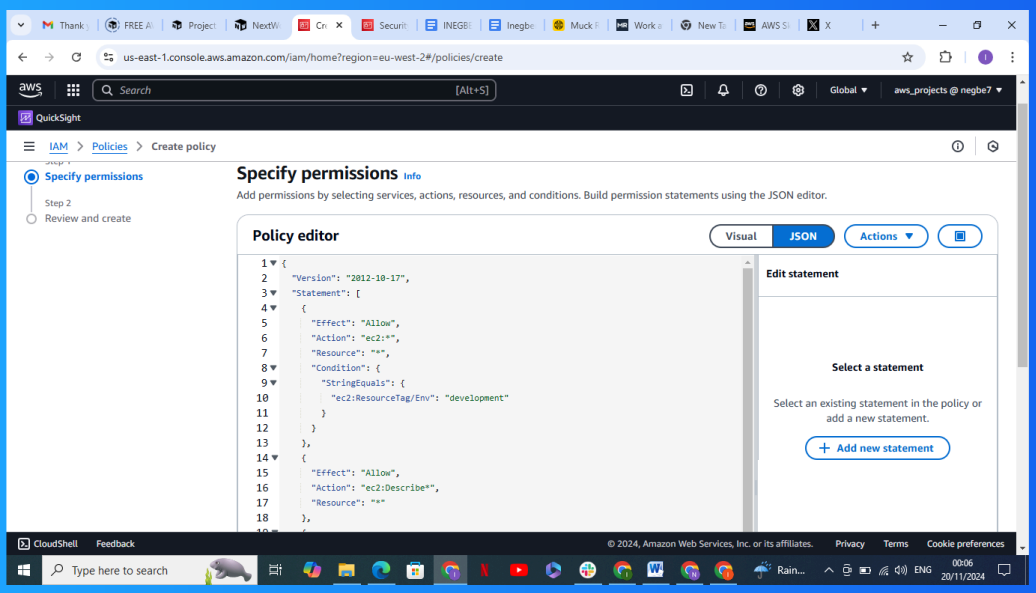
For this project, I've set up a policy using JSON. This method allowed me to define precise permissions, including actions, resources, and conditions, tailored to my requirements.

I've created a policy that allows full EC2 actions on instances tagged with Env=development, permits describing EC2 resources, and denies creating or deleting tags on any instance to enforce strict access control and environment separation.

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

The Effect, Action, and Resource attributes of a JSON policy define access control. Effect (Allow/Deny) decides permission, Action specifies operations (e.g., ec2:DescribeInstances), and Resource targets specific AWS resources or all (*).

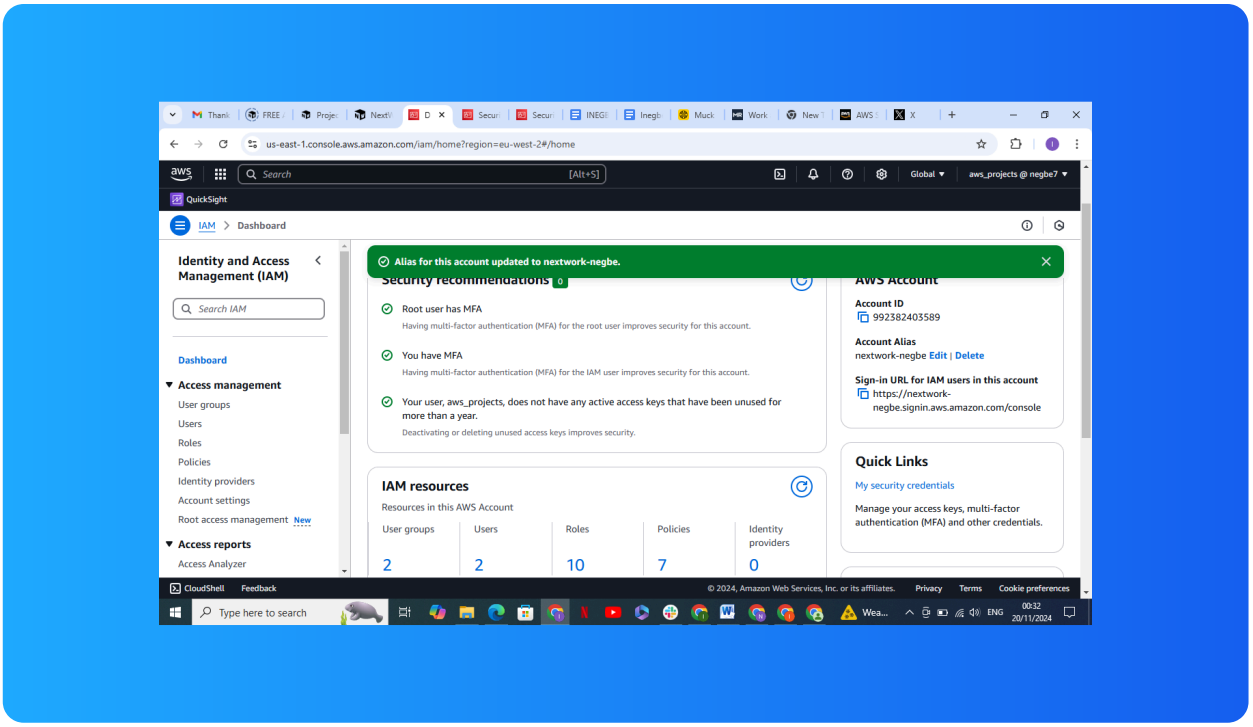
My JSON Policy



Account Alias

An account alias is a user-friendly name for your AWS account, replacing the default numeric account ID in the login URL. It simplifies access by making the sign-in URL easier to remember and share, e.ghttps://YourAlias.signin.aws.amazon.com/console

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



IAM Users and User Groups

Users

IAM users are individual accounts within AWS assigned to people or applications. They have unique credentials for access and can inherit permissions via policies.

User Groups

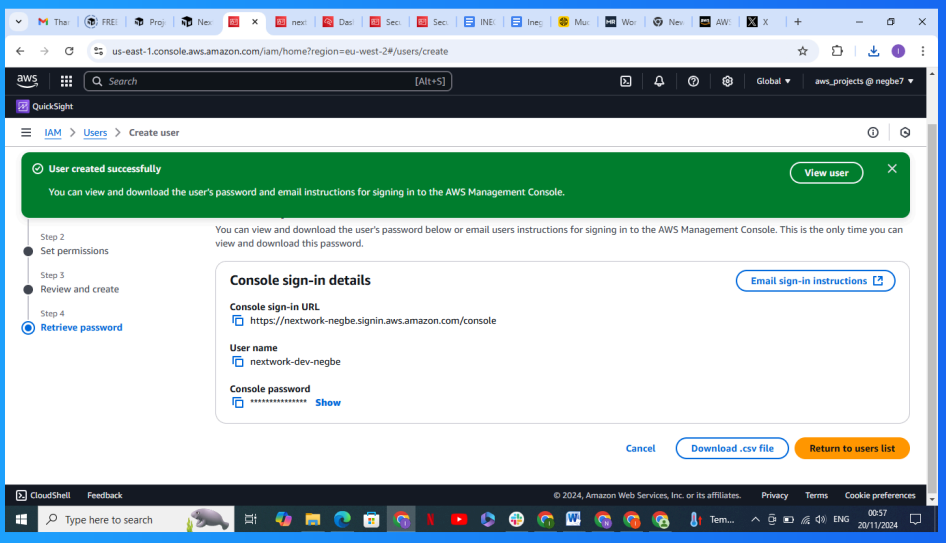
IAM user groups are collections of IAM users that share the same permissions. By grouping users, you can assign policies and manage access to AWS resources collectively rather than individually. This simplifies administration and ensures consistency

I attached the policy I created to this user group, which means all users in the group inherit permissions defined in the policy. They can access EC2 instances tagged Env=development and are restricted from creating or deleting tags, ensuring secure

Logging in as an IAM User

The first way is to securely email the user's sign-in URL, username, and password. The second way is to use a secure messaging tool to share these details directly, ensuring sensitive information is protected.

Once I logged in as my IAM user, I noticed AWS treated me like a new user with walkthroughs, and I saw some restrictions. This was because the IAM user had limited permissions due to the policy settings.

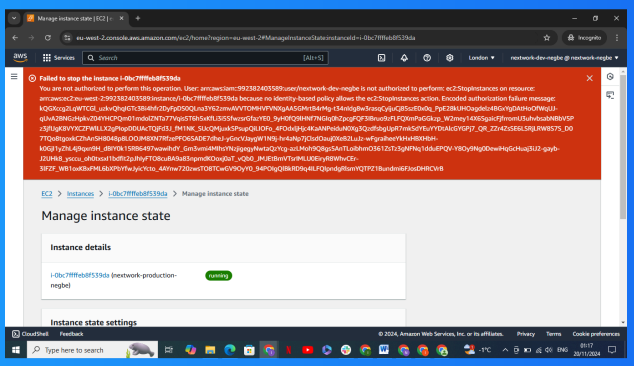


Testing IAM Policies

I tested my JSON IAM policy by attempting to stop both EC2 instances. The action on the production instance failed due to restricted permissions, while stopping the development environment instance was successful, demonstrating the policy's functiona

Stopping the production instance

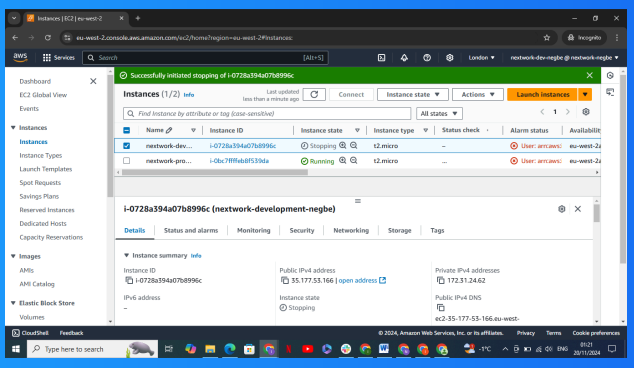
When I tried to stop the production instance, I encountered an authorization error. This was because the IAM policy attached to my user group explicitly restricted access to manage instances tagged as "production."



Testing IAM Policies

Stopping the development instance

Next, when I tried to stop the development instance, the action was successful. This was because the IAM policy allowed actions on resources tagged with "Env: development." The policy explicitly permitted stopping instances in the development environ



Everyone should be in a job they love.

Check out nextwork.org for more projects