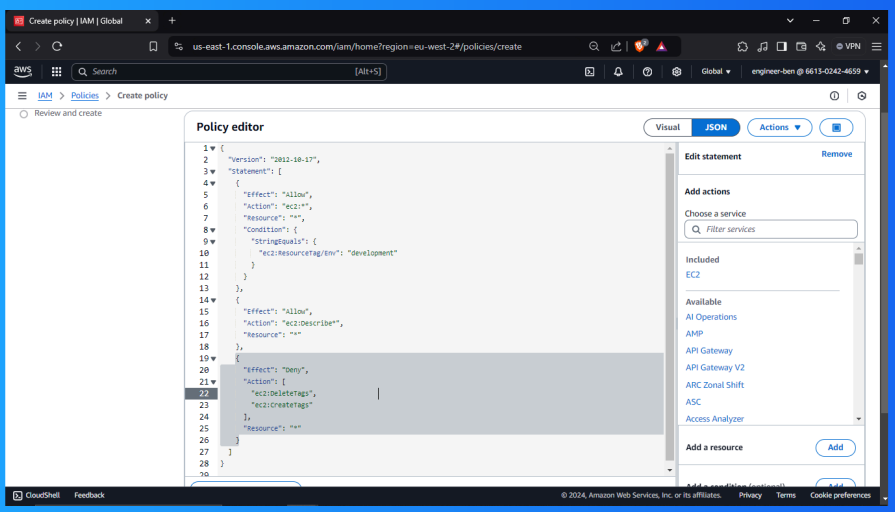
NextWork.org

Cloud Security with AWS IAM

AM ampahben3@gmail.com



AM
ampahben3@gmail.com
NextWork Student
NextWork.org

Introducing today's project!

What is AWS IAM?

AWS IAM (Identity and Access Management) is a service that controls access to AWS resources. It allows you to manage users, groups, and permissions, ensuring secure, fine-grained access control to services and data in the AWS cloud.

How I'm using AWS IAM in this project

In today's project, I used AWS IAM to create policies and assign them to IAM users and groups. This allowed me to control access to specific resources, ensuring secure operations while restricting permissions for certain actions.

One thing I didn't expect...

One thing I didn't expect in this project was encountering access denial errors despite following the setup steps. It highlighted the importance of carefully configuring IAM policies to ensure the correct permissions are in place for all actions.

This project took me...

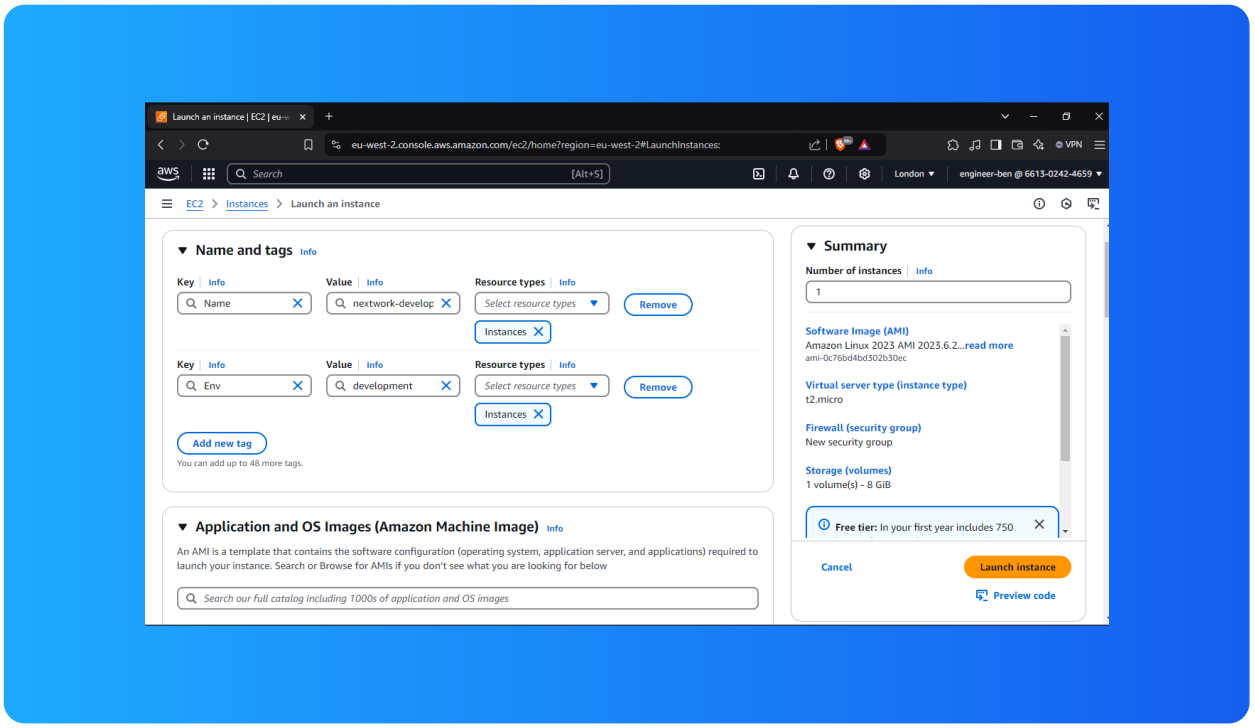
This project took me about an hour to complete. I spent time creating and testing IAM policies, which led to some unexpected access denial errors. Troubleshooting these errors helped me gain a deeper understanding of permission configurations in AWS.

AM
ampahben3@gmail.com
NextWork Student

Tags

Tags are labels added to resources like EC2 to improve tracking, cost allocation, and management. They organize resources by environment, project, or owner, enabling efficient governance and automation. A consistent tagging strategy is recommended...

The tag I've used on my EC2 instances is called env. For the production server, the tag is set as `name=env, value=production`, and for the development server, it is set as `name=env, value=development`. This helps distinguish environments always....



AM
ampahben3@gmail.com
NextWork Student
NextWork.org

IAM Policies

IAM policies are rules that define who can do what with AWS resources. They grant permissions to IAM users, groups, or roles, specifying allowed or denied actions on resources and the conditions under which these permissions apply.

The policy I set up

For this project, I've set up a policy using the JSON editor. This approach allowed me to directly define and customize the permissions by specifying the actions, resources, and conditions, ensuring precise control over access to AWS resources.

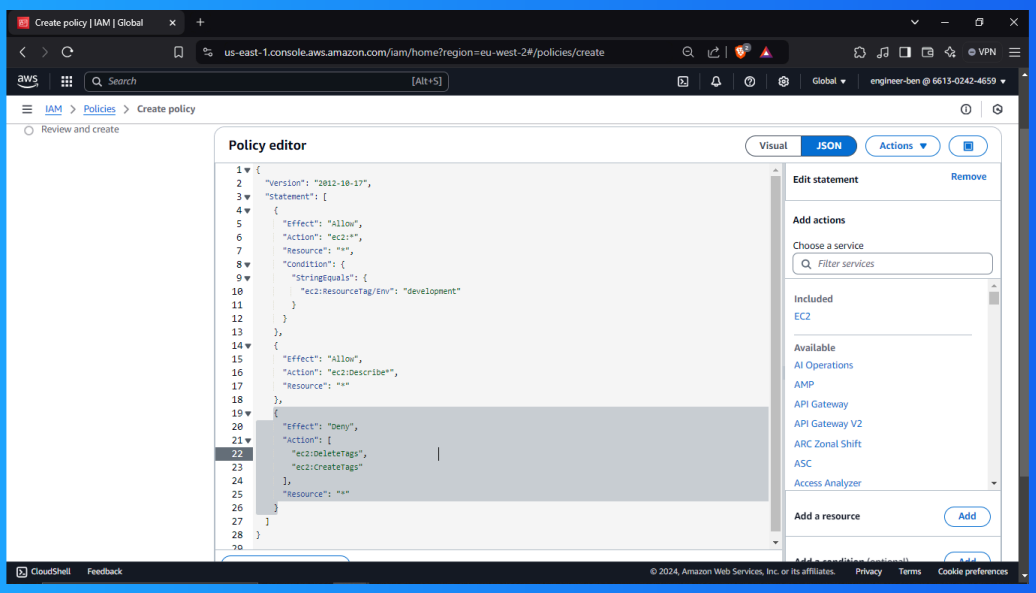
I've created a policy that allows actions on EC2 resources tagged with `Env=development` while restricting actions to only describe EC2 resources globally. Additionally, it explicitly denies the ability to create or delete tags on any EC2 resource.

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

The Effect, Action, and Resource attributes in a JSON policy define key components.Effect determines if the action is allowed or denied.Action specifies the operations that are permitted or denied.Resource indicates which resource the policy applies.

AM
ampahben3@gmail.com
NextWork Student

My JSON Policy

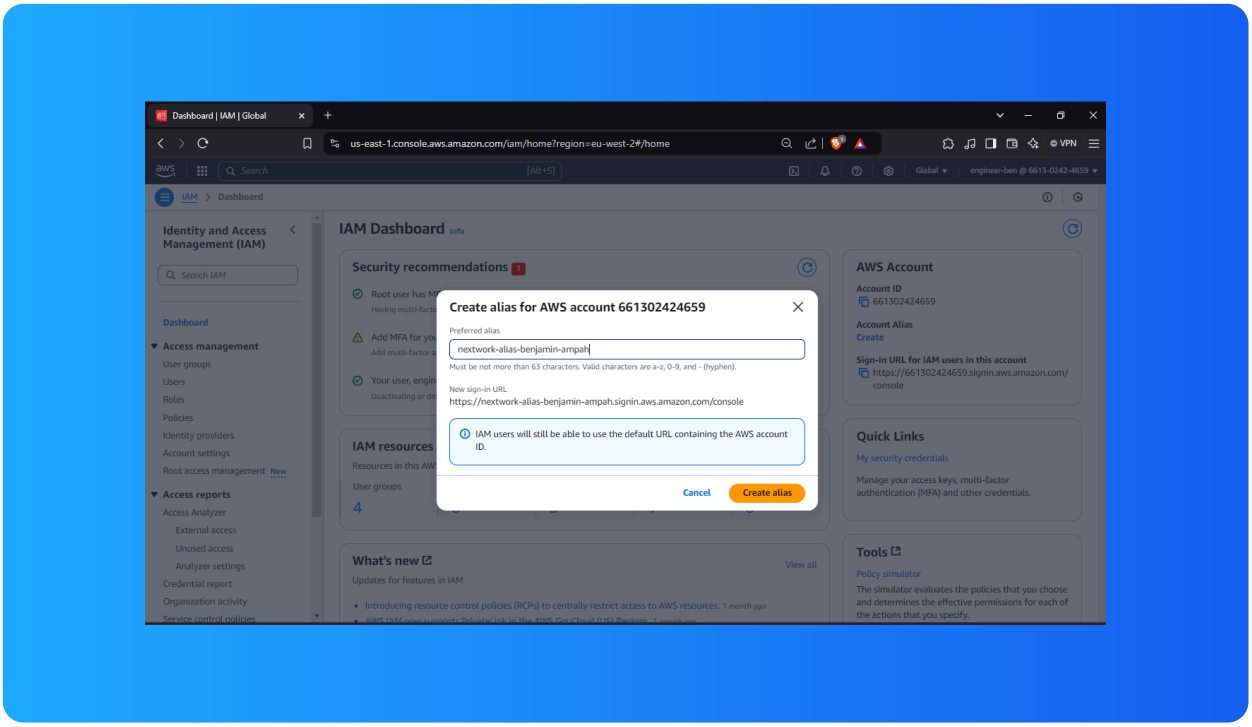


AM
ampahben3@gmail.com
NextWork Student

Account Alias

An account alias is a custom name for your AWS account, making it easier to identify and access instead of using the AWS account ID. It simplifies managing your account and improves usability, especially when accessing the AWS Management Console.

Creating an account alias took me less than a minute. Now, my new AWS console sign-in URL is simplified, as I just had to ensure my name followed the rules to avoid any errors.



AM
ampahben3@gmail.com
NextWork Student
NextWork.org

IAM Users and User Groups

Users

IAM users are individual identities within AWS that allow access to resources. Each user has its own credentials (username, password, access keys) and can be assigned specific permissions based on the roles or policies granted to them.

User Groups

IAM user groups are collections of IAM users that share the same permissions. Instead of assigning permissions to each user individually, you assign them to the group, simplifying access management and ensuring consistent permissions for users.

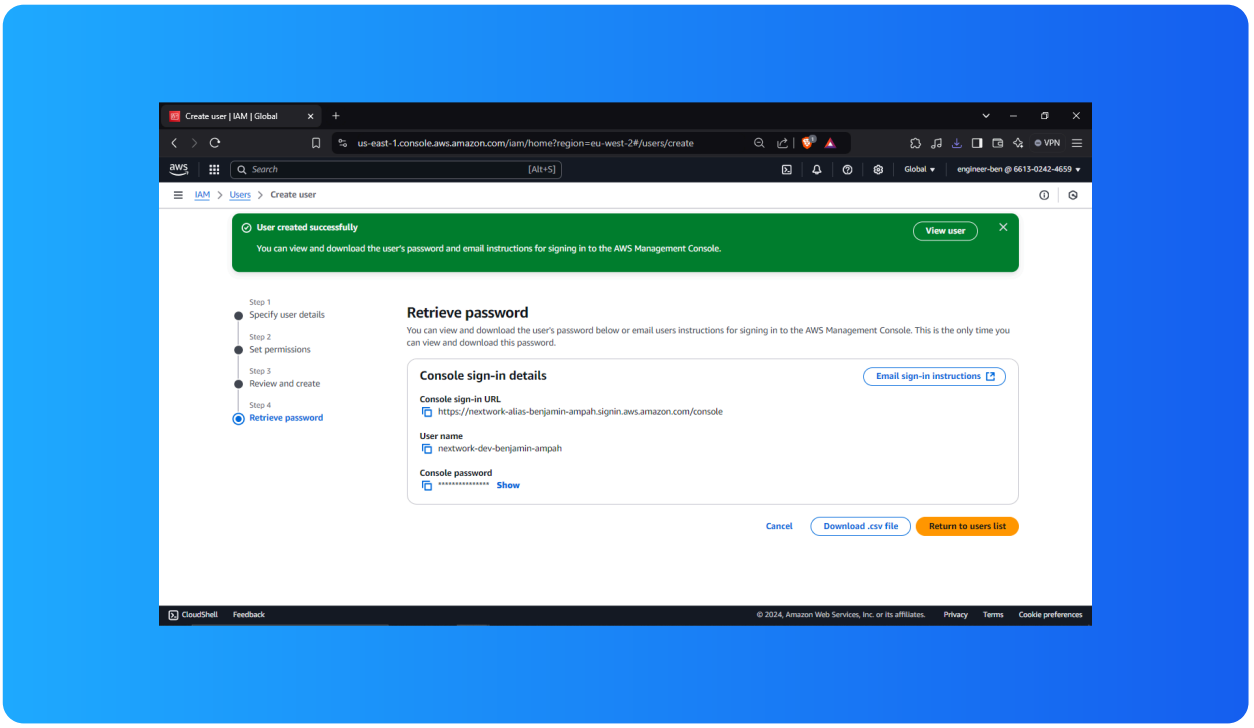
I attached the policy I created to this user group, which means all users in the group inherit the permissions defined in the policy. This ensures consistent access control and simplifies permission management for users with similar roles.

AM
ampahben3@gmail.com
NextWork Student

Logging in as an IAM User

The first way is by sending the user their sign-in URL, username, and temporary password via email or secure messaging. The second way is by generating a sign-in link and credentials directly within the AWS IAM console for the user.

Once I logged in as my IAM user, I noticed "Access Denied" to "servicecatalog:ListApplications". This was because the IAM user didn't have the necessary permissions to access the service catalog, as it wasn't included in their assigned policy.



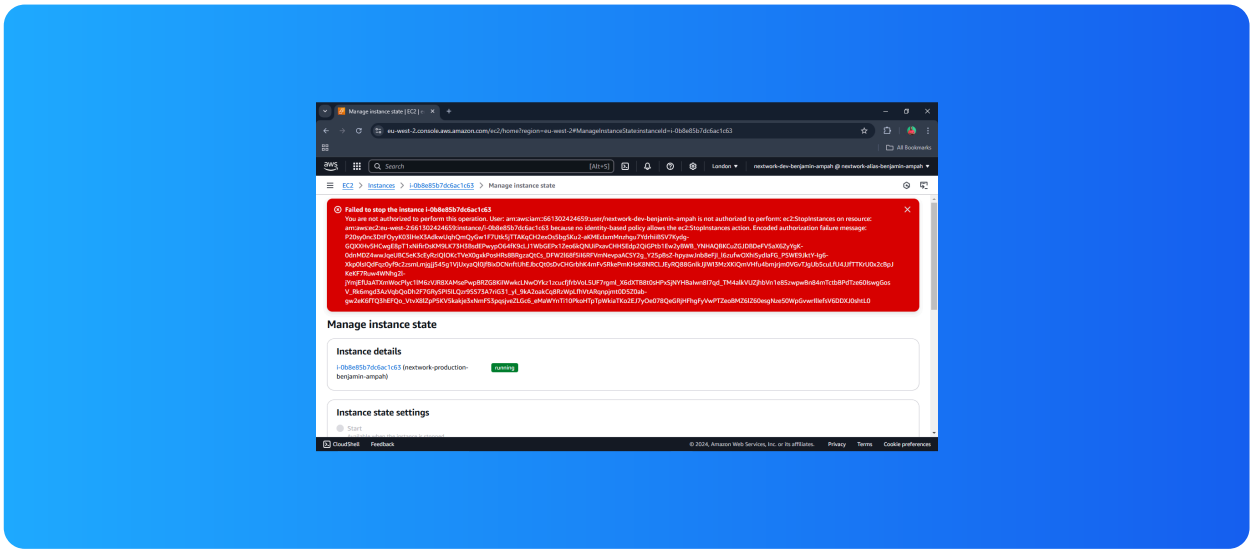
AM
ampahben3@gmail.com
NextWork Student

Testing IAM Policies

I tested my JSON IAM policy by successfully initiating the stopping of i-0c0ea7e645a508047 and encountering an access denial when trying to stop the production instance, as my IAM user didn't have the required permissions for that operation.

Stopping the production instance

When I tried to stop the production instance, I got an error: "You are not authorized to perform this operation." This was because my IAM user didn't have the `ec2:StopInstances` permission in the assigned policy.

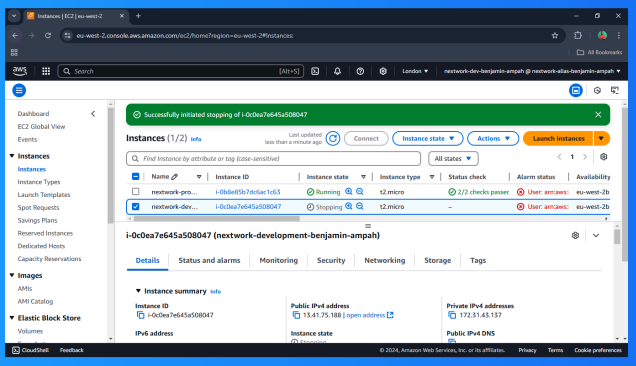


AM
ampahben3@gmail.com
NextWork Student

Testing IAM Policies

Stopping the development instance

Next, when I tried to stop the development instance, the operation was successful. This was because my IAM policy allowed stopping instances with the tag `Env=development`, which the development instance had assigned.



Everyone should be in a job they love.

Check out nextwork.org for more projects