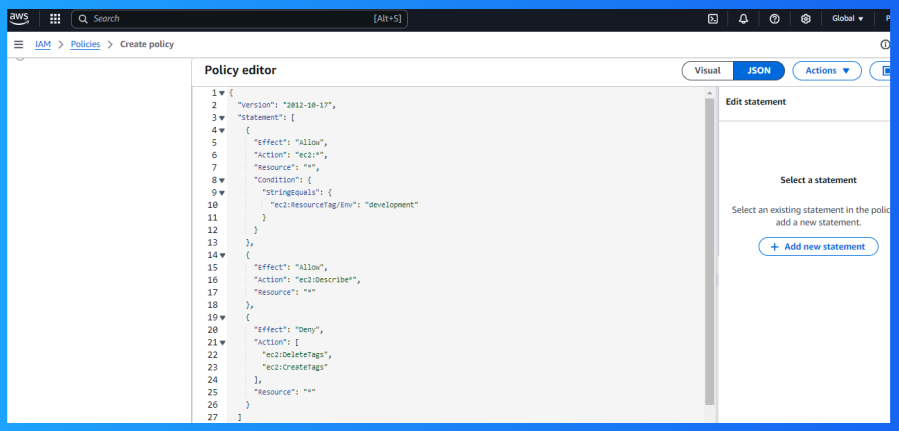
NextWork.org

Cloud Security with AWS IAM

RD rdanu.inbox@gmail.com



RD
rdanu.inbox@gmail.com
NextWork Student
NextWork.org

Introducing today's project!

What is AWS IAM?

AWS IAM is a service that manages user access and permissions to AWS resources. It ensures secure, granular control over who can access and perform actions on specific resources, helping with security and compliance.

How I'm using AWS IAM in this project

In today's project, I used AWS IAM to create policies that allowed access to the development instance and denied access to the production instance for specific users.

One thing I didn't expect...

One thing I didn't expect in this project was how quickly IAM policies can impact access to resources, especially when testing permissions on EC2 instances. It was a good reminder of how precise and powerful access controls are in AWS.

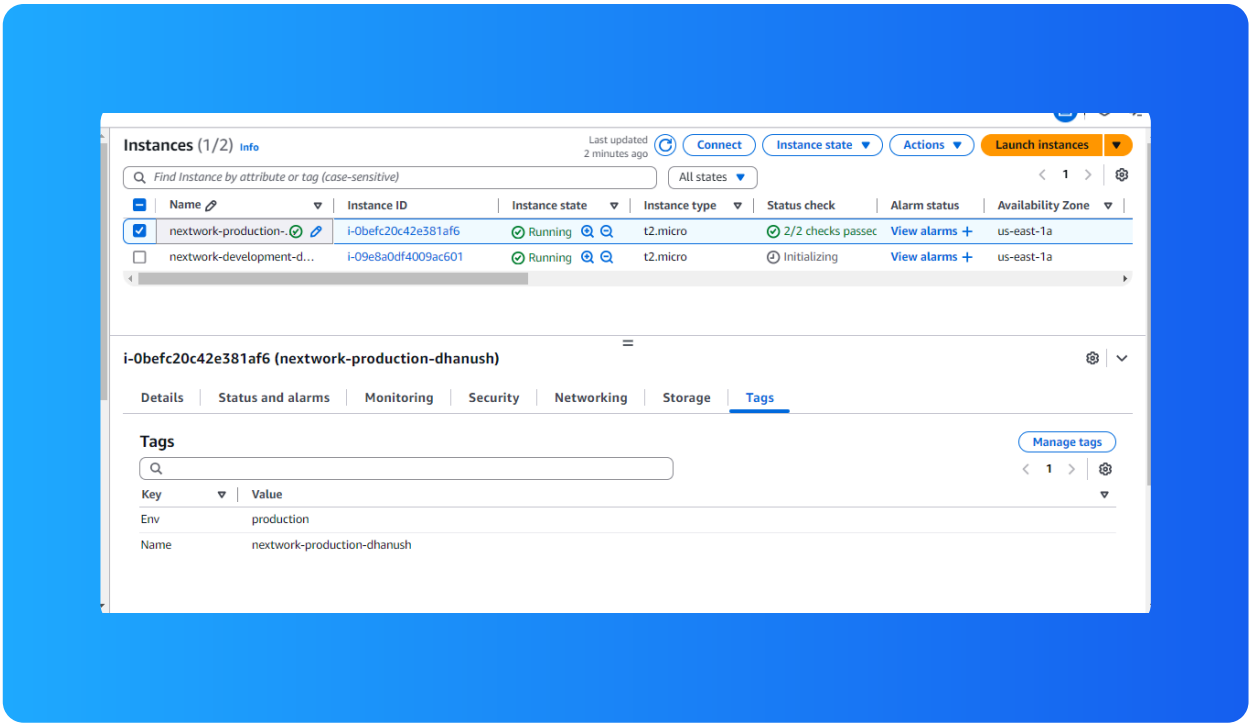
This project took me...

This project took me nearly 75 minutes to complete, including the time spent setting up the EC2 instances, creating IAM policies, and testing access controls. It involved careful configuration to ensure the correct permissions were applied.

Tags

Tags are labels (key-value pairs) assigned to AWS resources for organizing, managing, and tracking them. They're useful for cost allocation, resource management, automation, and enforcing security policies.

The tag I've used on my EC2 instances is called "Env" with a value of "production" or "development" to label the instances used in production vs development environments.



RD
rdanu.inbox@gmail.com
NextWork Student
NextWork.org

IAM Policies

IAM Policies are documents that define permissions for AWS resources, controlling what actions users, groups, or roles can perform. They help manage access securely by specifying allowed or denied actions.

The policy I set up

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

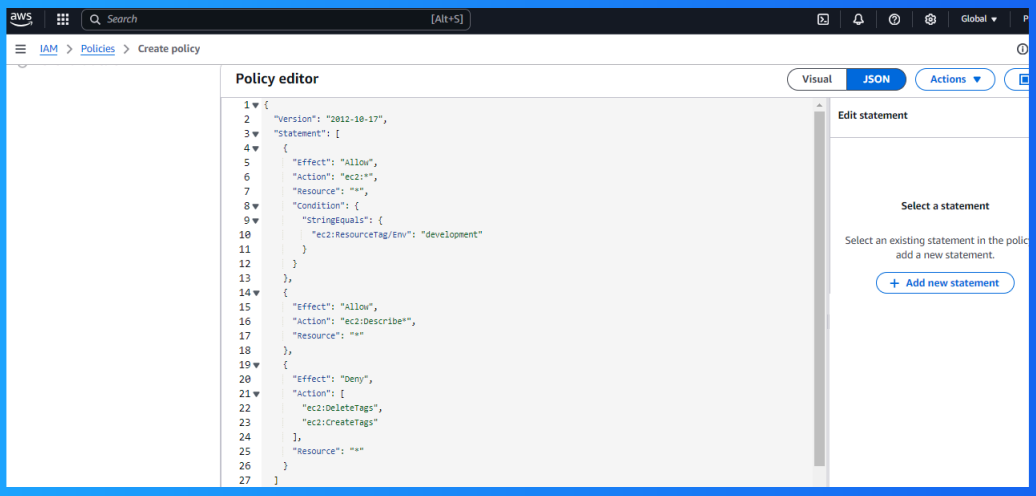
I've created a policy that allows access to the development instance and denies access to the production instance, ensuring environment-specific access control.

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

The Effect determines whether actions are allowed or denied, the Action specifies what operations are permitted, and the Resource defines which AWS resources the policy applies to.

RD
rdanu.inbox@gmail.com
NextWork Student

My JSON Policy

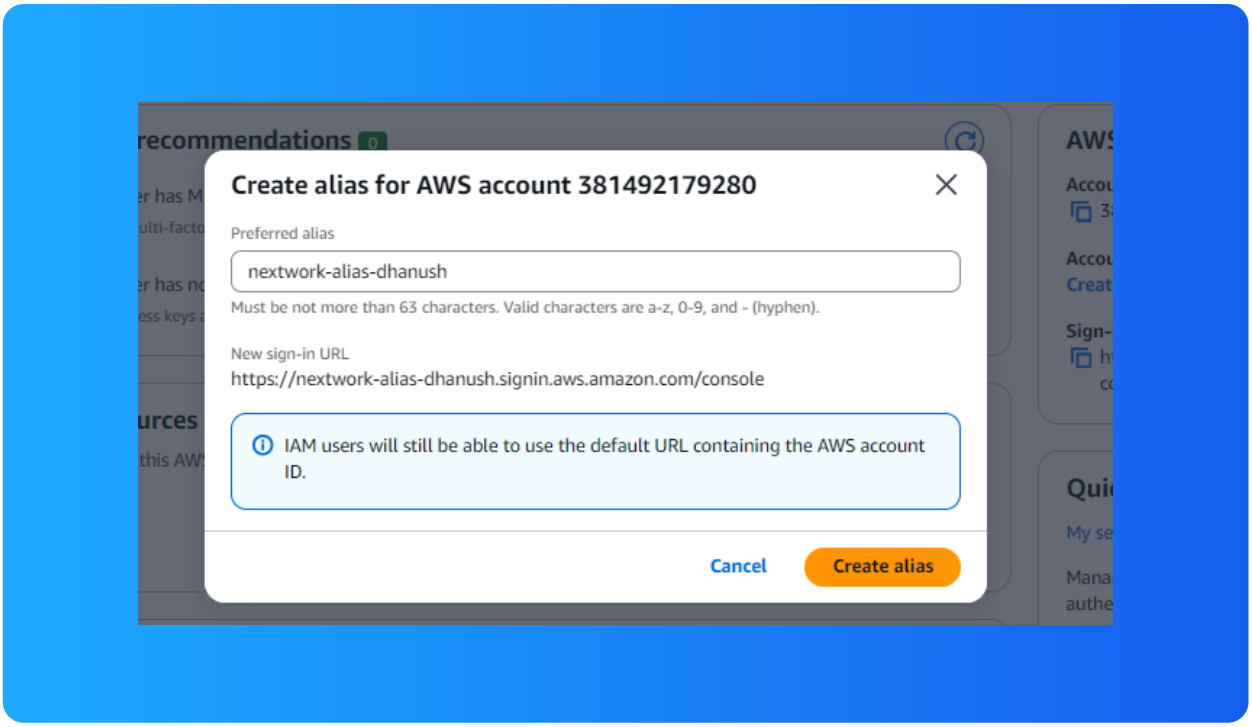


RD
rdanu.inbox@gmail.com
NextWork Student

Account Alias

An account alias is a user-friendly name that you can assign to your AWS account, making it easier to access and manage. It replaces the default AWS account ID in the sign-in URL for convenience.

Creating an account alias took me just a few minutes. Now, my new AWS console sign-in URL is more user-friendly and looks "https://nextwork-alias-dhanush.signin.aws.amazon.com/console".



RD
rdanu.inbox@gmail.com
NextWork Student
NextWork.org

IAM Users and User Groups

Users

IAM users are individual identities within AWS that are used to grant access to AWS resources. Each IAM user has specific permissions based on the policies attached to them, allowing them to perform actions within the AWS environment.

User Groups

IAM user groups are collections of IAM users that allow you to manage permissions for multiple users at once. By assigning users to a group, you can apply the same permissions to all members of the group, simplifying access management.

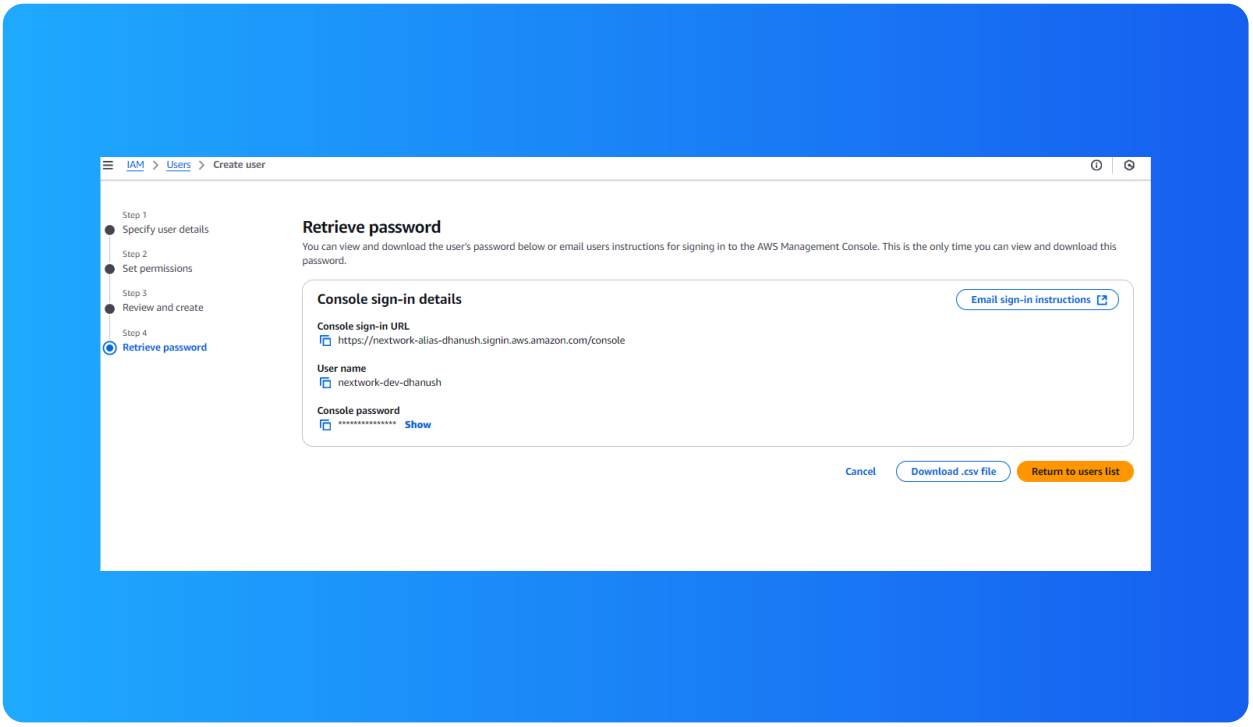
I attached the policy I created to this user group, which means all users in the group will inherit the permissions defined in the policy. This simplifies access management by ensuring consistent permissions for all members of the group.

RD
rdanu.inbox@gmail.com
NextWork Student

Logging in as an IAM User

The first way is to send the sign-in details directly to the user via email, including their username and the temporary password. The 2nd way is to generate a sign-in link from the console and share it with the user, allowing them to access directly.

Once I logged in as my IAM user, I noticed that some of your dashboard panels are showing "Access Denied" already. This was because the IAM user didn't have the necessary permissions to access those resources or services, based on the policies.



RD

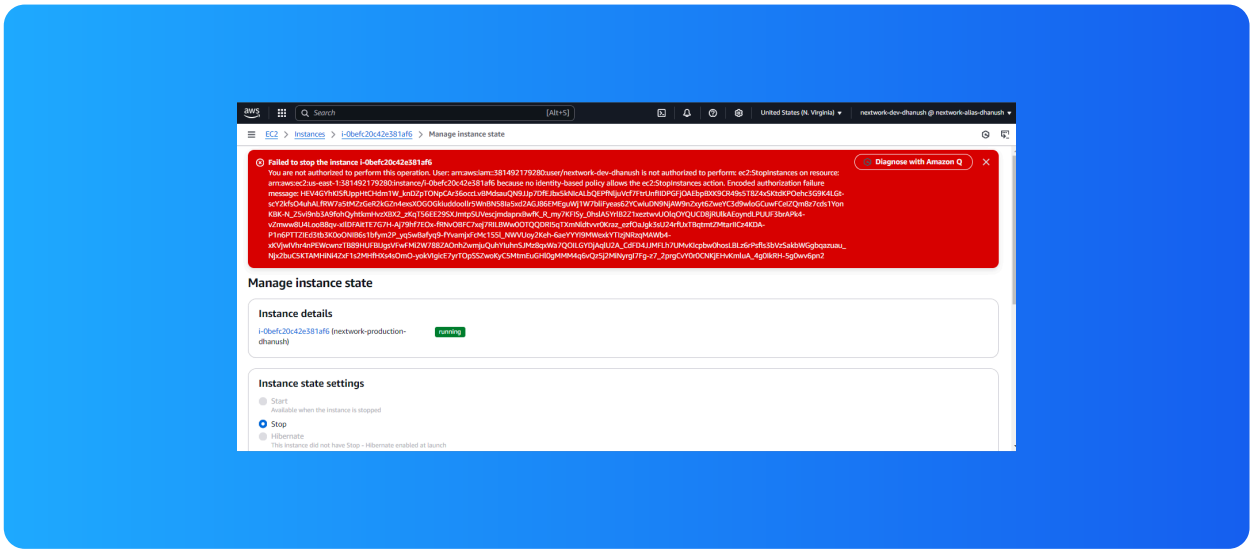
rdanu.inbox@gmail.com
NextWork Student

Testing IAM Policies

I tested my JSON IAM policy by attempting to access my two EC2 instances. For the development instance, the policy allowed access, while for the production instance, access was denied, as per the permissions defined in the policy.

Stopping the production instance

When I tried to stop the production instance says failed to stop this instance. This was because we're not authorized! We don't have permission to stop any instance with the production tag.

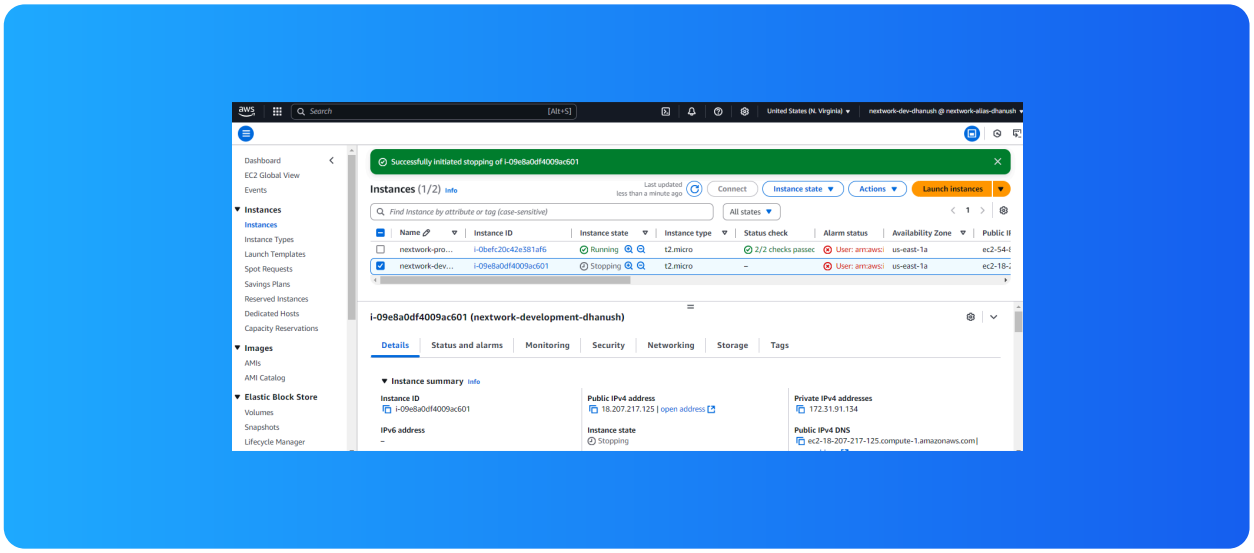


RD
rdanu.inbox@gmail.com
NextWork Student

Testing IAM Policies

Stopping the development instance

Next, when I tried to stop the development instance, I was able to do so successfully. This was because the IAM policy granted the necessary permissions to stop the instance, as the policy allowed actions for the development instance.



Everyone should be in a job they love.

Check out nextwork.org for more projects