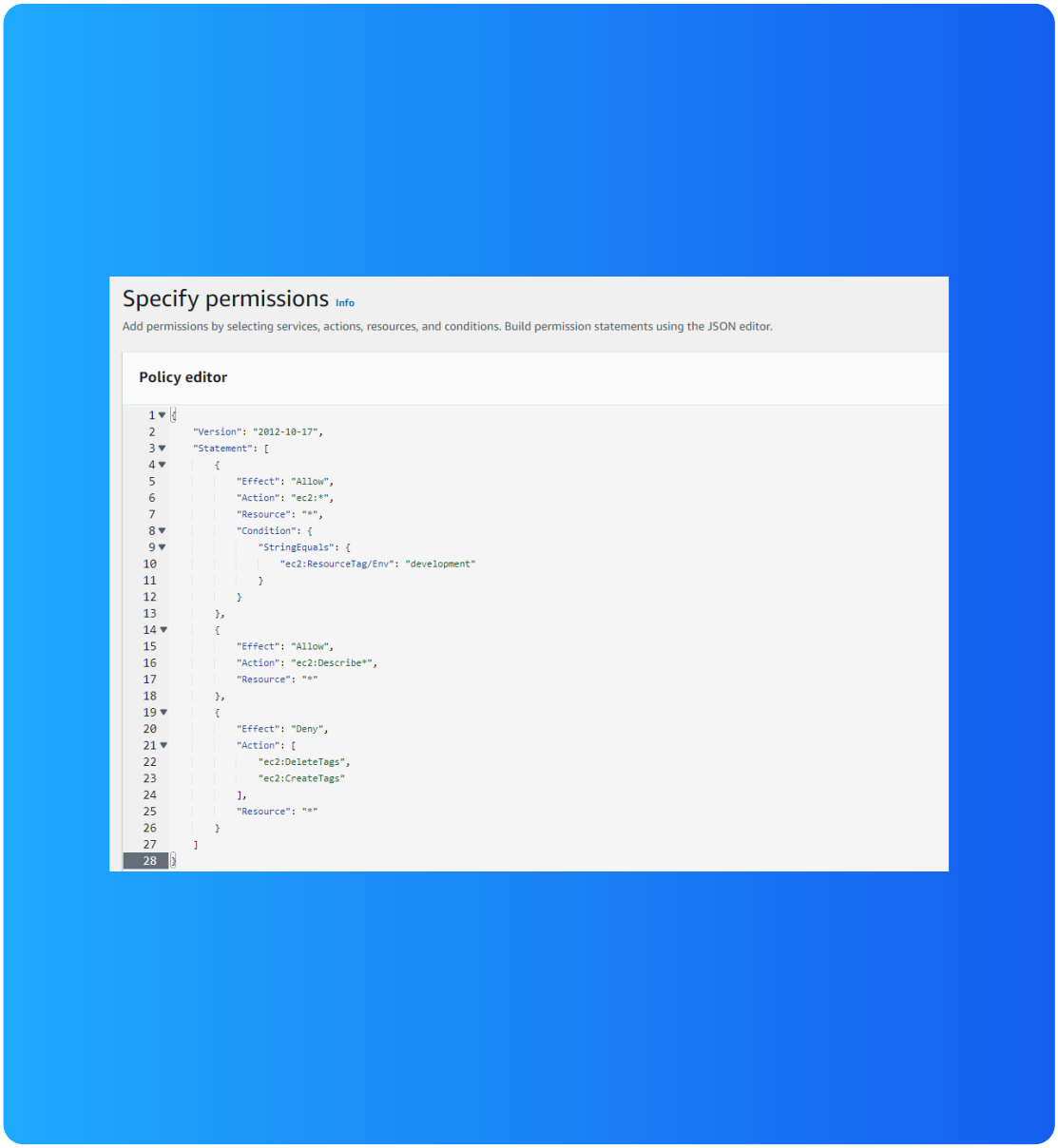
NextWork.org

Cloud Security with AWS IAM

Emmanuel Enalpe III



Emmanuel Enalpe III
NextWork Student

Introducing today's project!

What is AWS IAM?

AWS IAM (Identity and Access Management) is a service that enables you to manage users, groups, and permissions securely. It's useful for controlling access to AWS resources, ensuring that only authorized users can perform specific actions.

How I'm using AWS IAM in this project

In today’s project, I used AWS IAM to create and manage user groups, set up permissions with JSON policies, and test access control by managing EC2 instances. This ensured that users had appropriate access while protecting sensitive resources.

One thing I didn't expect...

One thing I didn’t expect was the level of detail required to correctly configure IAM policies for different types of resources. Ensuring precise permissions while balancing security and usability was more complex than anticipated.

This project took me...

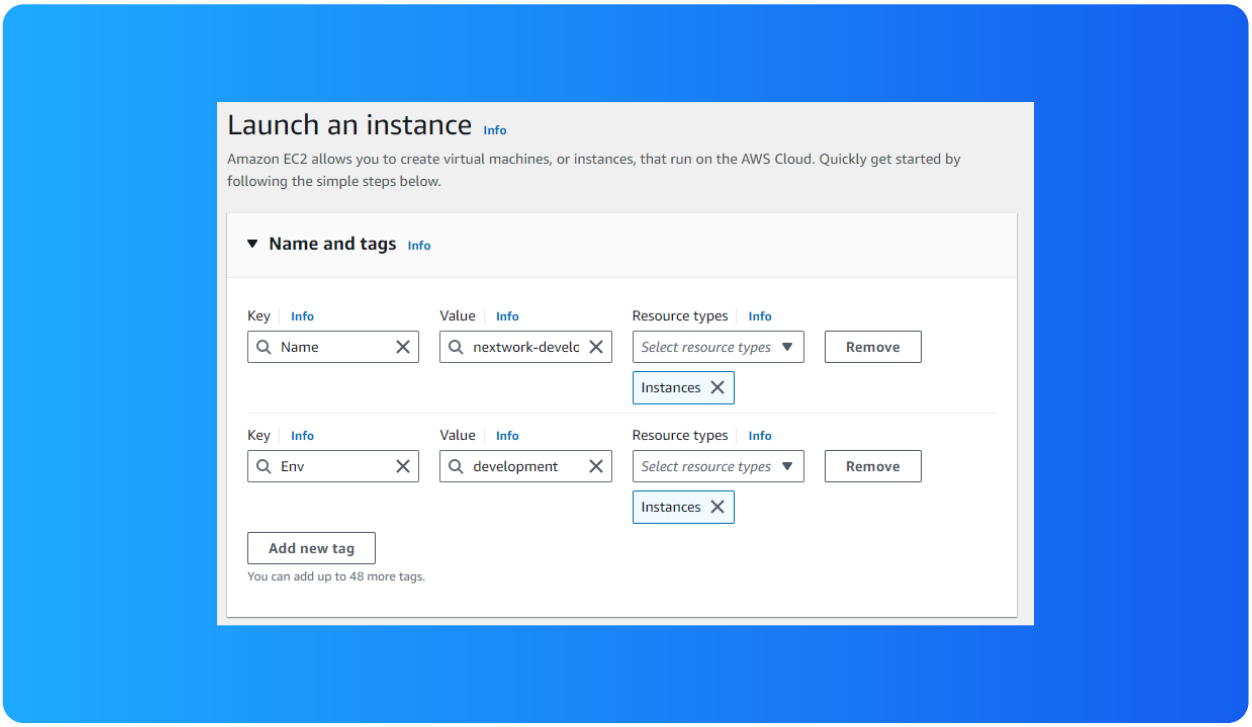
This project took me several minutes, including time for setting up IAM policies, testing permissions, and troubleshooting. The complexity of configuring and verifying the correct access controls contributed to the overall time spent.

Emmanuel Enalpe III
NextWork Student

Tags

Tags are similar to labels; tags are attached to AWS resources for identifying all resources with the same tag at once.

The tag I've used on my EC2 instances is called Name and Env. The value I've assigned for my instances are nextwork-development and development.



Emmanuel Enalpe III
NextWork Student

IAM Policies

IAM Policies are guidelines governing who has access to your AWS resources. It all comes down to granting IAM users, groups, or roles rights, defining what they can and cannot do on specific resources.

The policy I set up

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

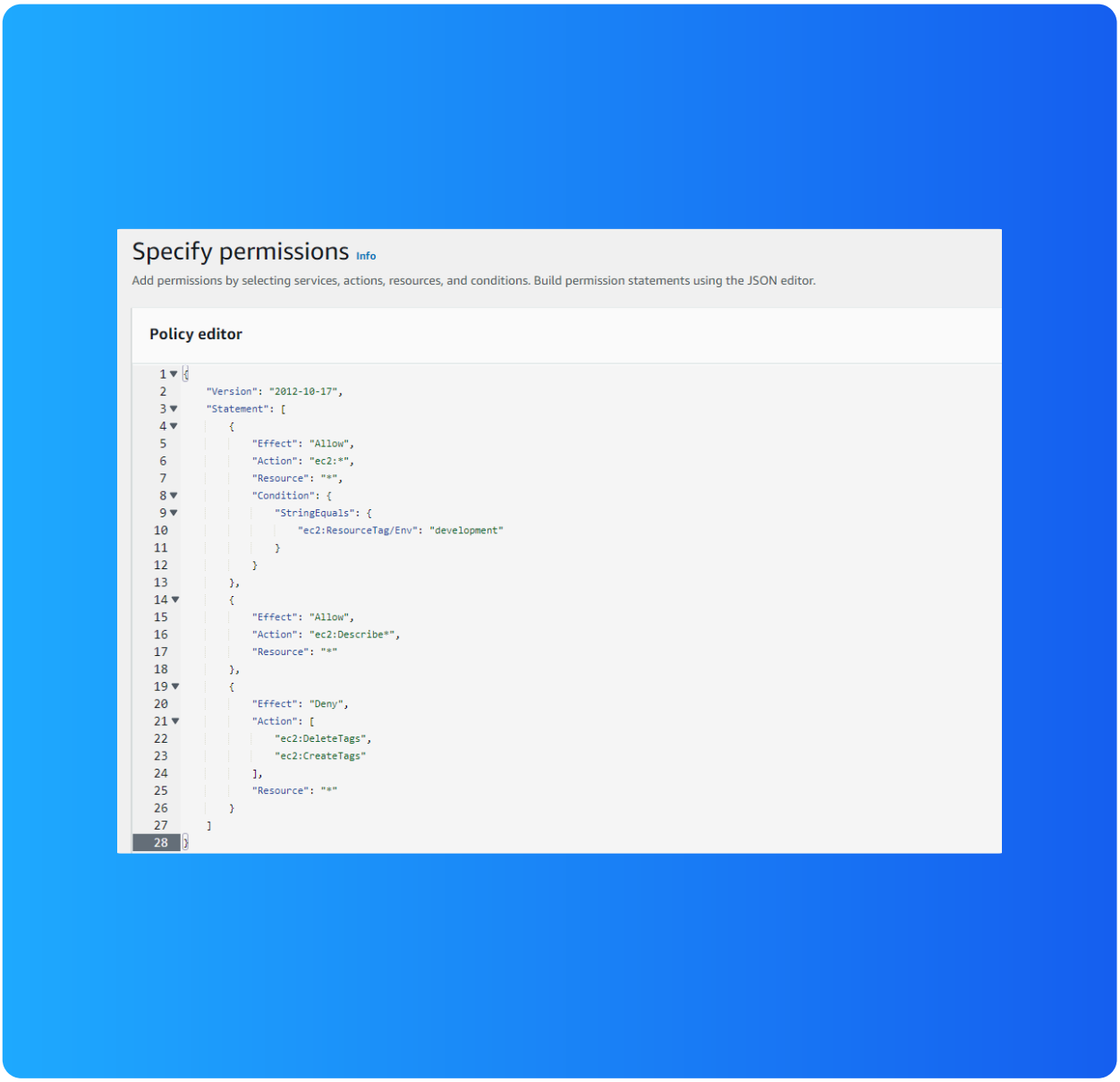
I've created a policy that allows some actions (like starting, stopping, and describing EC2 instances) for instances tagged with "Env = development" while denying the ability to create or delete tags for all instances.

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

The Effect, Action, and Resource attributes of a JSON policy mean the following: *Effect* specifies whether access is allowed or denied, *Action* defines what operations are permitted, and *Resource* indicates which AWS resources the policy applies.

Emmanuel Enalpe III
NextWork Student

My JSON Policy



Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/Env": "development"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:Describe*",
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "ec2:DeleteTags",
                "ec2:CreateTags"
            ],
            "Resource": "*"
        }
    ]
}
```

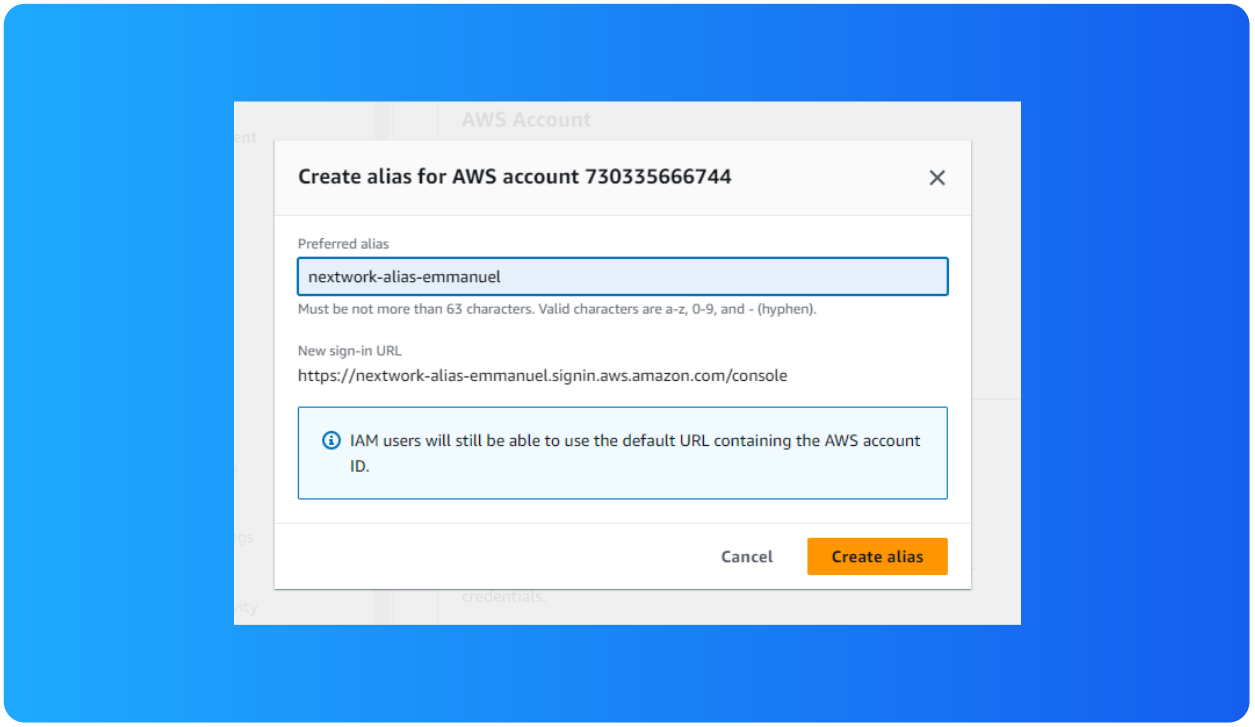
Emmanuel Enalpe III
NextWork Student

Account Alias

An account alias is a friendly name for your AWS account that you can use instead of your account ID to sign in to the AWS Management Console.

Creating an account alias took me about 2–5 minutes.

Now, my new AWS console sign-in URL is https://nextwork-alias-emmanuel.signin.aws.amazon.com/console



Emmanuel Enalpe III
NextWork Student

IAM Users and User Groups

Users

IAM users are individuals or services created in AWS Identity and Access Management (IAM) that have unique credentials to interact with AWS resources. They can be assigned specific permissions and policies to control access based on their roles.

User Groups

IAM user groups are collections of IAM users that allow you to manage permissions more easily by applying the same set of permissions to multiple users.

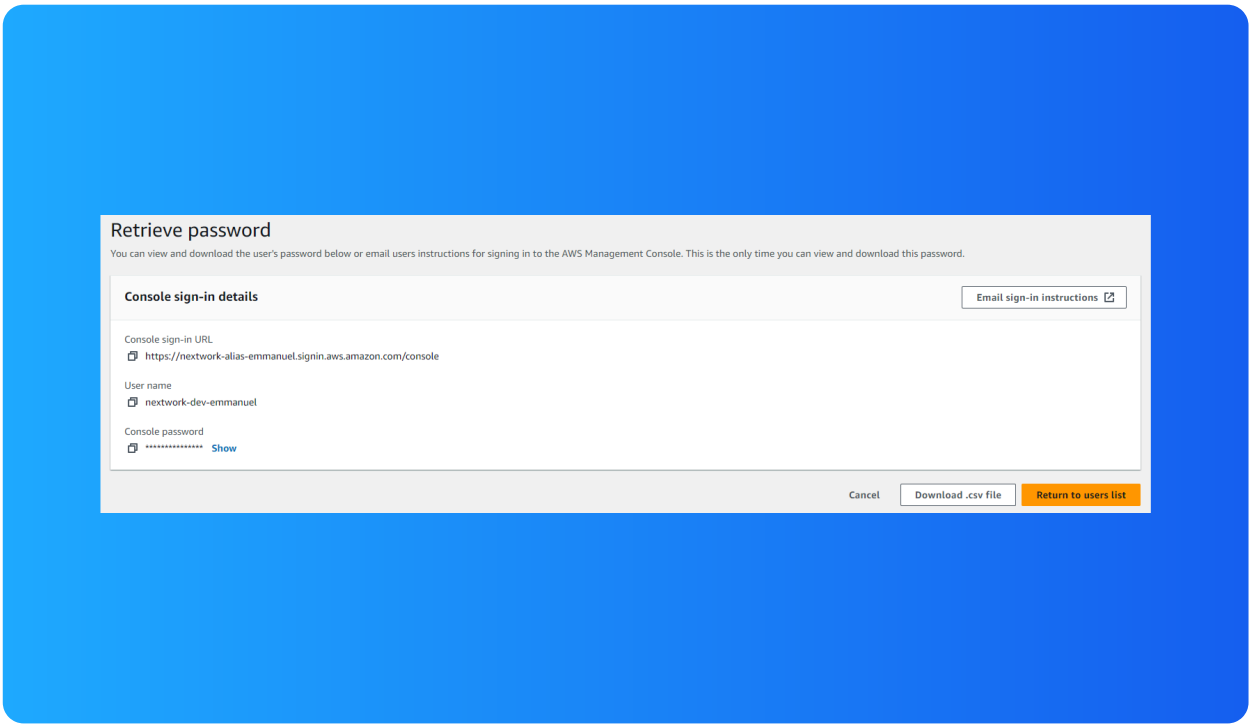
I attached the policy I created to this user group, which means all users in the group will inherit the permissions defined in the policy. This simplifies management by ensuring consistent access control without needing to configure each user.

Emmanuel Enalpe III
NextWork Student

Logging in as an IAM User

The first way is to send the user's sign-in details via email, which includes their username and a temporary password. The second way is to manually provide the login URL, username, and password in person or through secure communication.

Once I logged in as my IAM user, I noticed a simplified AWS dashboard with limited access, showing only the services and features allowed by the permissions set in the policy. The interface is tailored based on the restrictions in place.



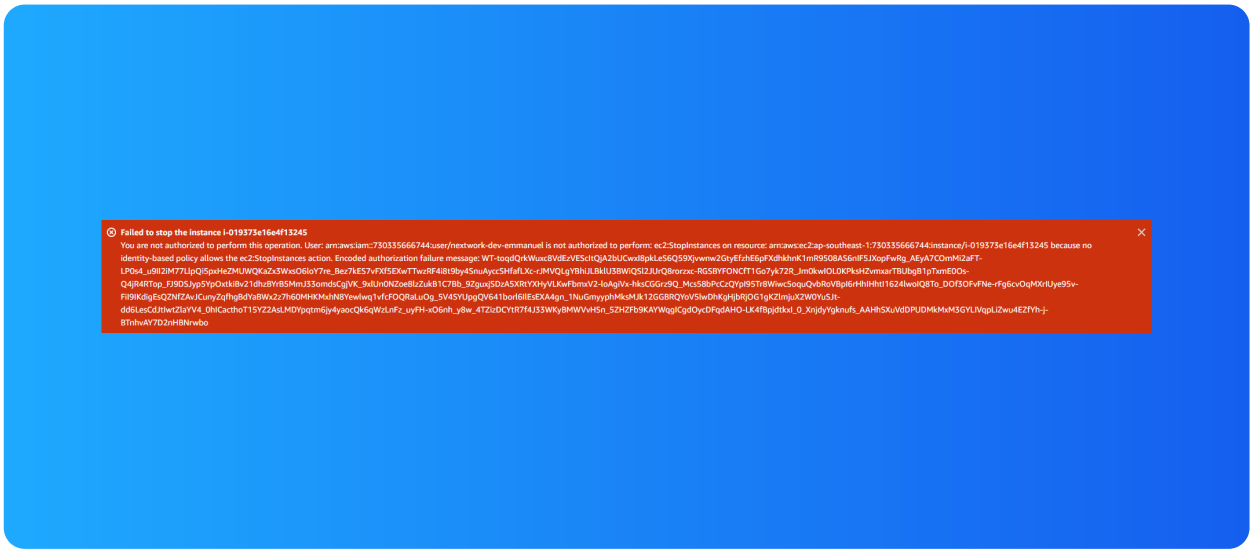
Emmanuel Enalpe III
NextWork Student

Testing IAM Policies

I tested my JSON IAM policy by attempting to start and stop my two EC2 instances, verifying that the permissions allowed these actions as specified. This confirmed the policy was correctly configured for EC2 instance management.

Stopping the production instance

When I tried to stop the production instance, the action was denied, indicating that my IAM policy correctly restricted access to production resources, preventing unauthorized modifications or shutdowns as intended.

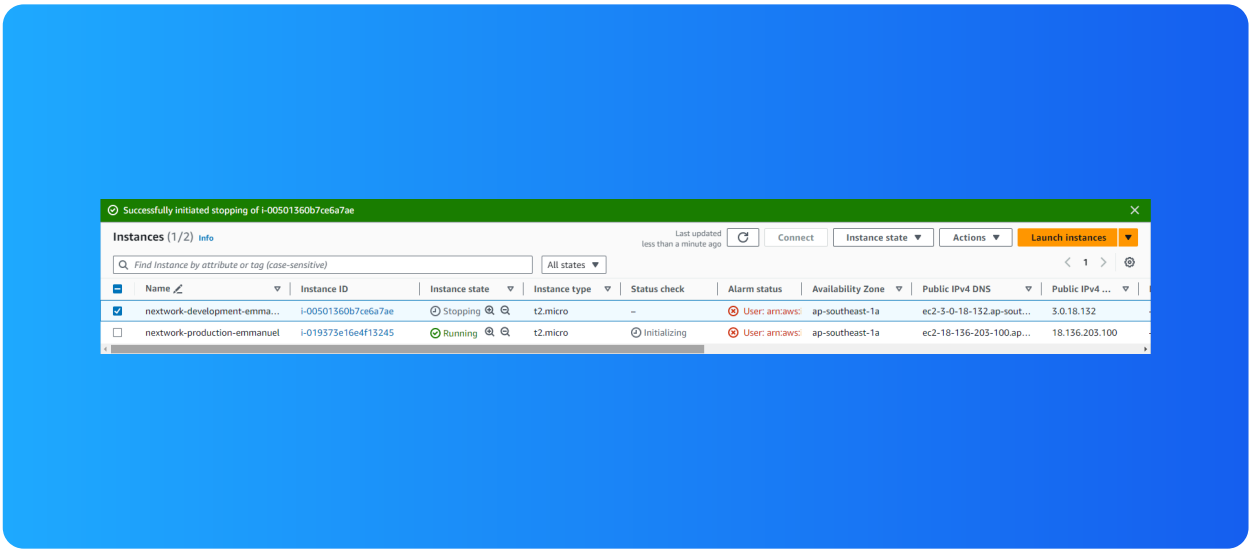


Emmanuel Enalpe III
NextWork Student

Testing IAM Policies

Stopping the development instance

Next, when I tried to stop the development instance, the action was successful, confirming that my IAM policy granted the appropriate permissions for managing non-production resources like the development environment.



Everyone should be in a job they love.

Check out nextwork.org for more projects