nextwork.org

Cloud Security with AWS IAM

Joshua Lopez

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": "*"
    }
  ]
}
```

Joshua Lopez
NextWork Student

Introducing Today's Project!

Using the AWS Identity and Access Management (IAM) service to control who is authenticated (signed in) and who has permissions in our AWS account. We will launch an EC2 instance, then control who has access to it by creating some IAM policies.

Tools and concepts

Services I used were Identity and Access Management (IAM) and Elastic Compute Cloud (EC2). Key concepts I learned are Tags, instances(virtual machine), IAM Policy, account alias, users and groups, and environments (prod and dev).

Project reflection

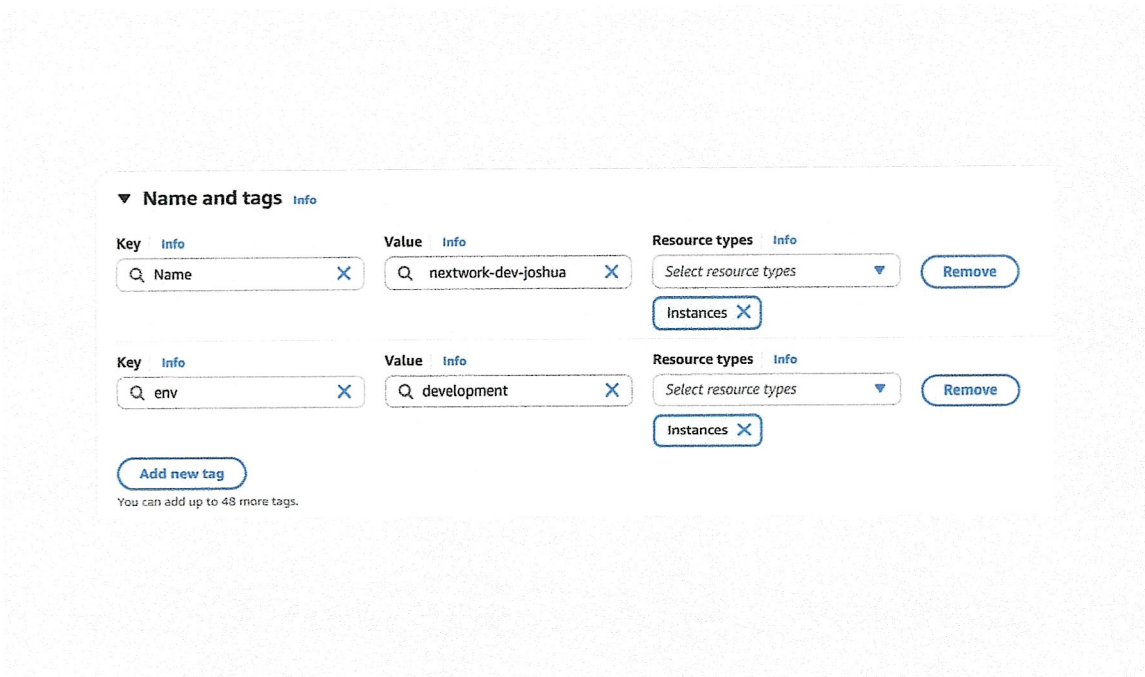
This project took me approximately an hour and a half. The most challenging part was understanding how policies affect access to resources. It was most rewarding to use the Policy Simulator for testing. Key benefit was not affecting aws resources by bringing them down for example. As was the result testing with an actual users login session.

Joshua Lopez
NextWork Student

Tags

Tags are like labels you can attach to AWS resources for organization.

We are creating a tag called "Env" with a value of "production" or "development" to label the instances used in production vs development.



Joshua Lopez
NextWork Student

IAM Policies

An IAM policy is a rule for who can do what with your AWS resources. It's all about giving permissions to IAM users, groups, or roles, saying what they can or can't do on certain resources, and when those rules kick in.

The policy I set up

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

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 are: Effect—Allow or Deny—a particular action. Action is a list of actions that the policy allows or denies. Which resource the policy apply to.

My JSON Policy

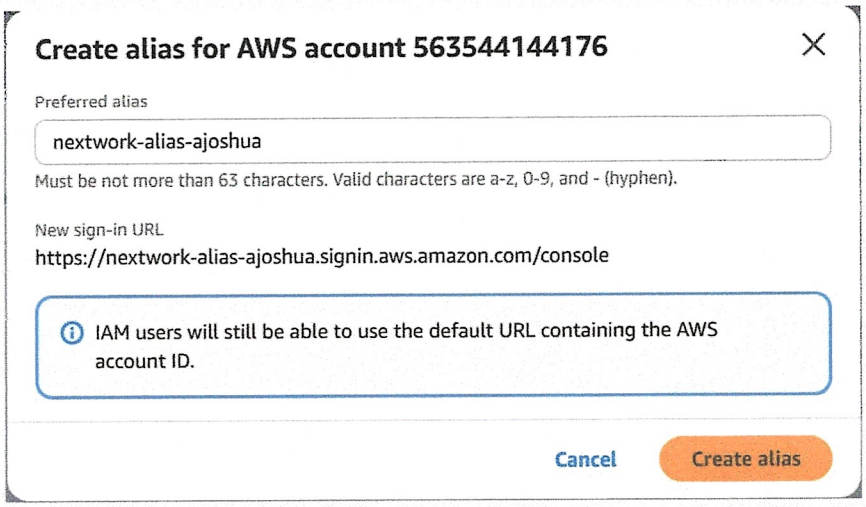
Policy editor

```
{
  "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": "*"
    }
  ]
}
```

Account Alias

Creating an alias makes it easier to remember and share your AWS console's login URL with others, such as NextWork's new intern. Companies often use this to make their AWS account sign-in page more user-friendly.

It took me a couple of minutes to set up an account alias. Now, my new AWS console sign-in URL is https://nextwork-alias-ajoshua.signin.aws.amazon.com/console



Joshua Lopez
NextWork Student

IAM Users and User Groups

Users

IAM stands for Identity and Access Management. We will use AWS IAM to manage the access level that other users and services have to our resources.

User Groups

IAM user groups are a collection/folder of IAM users. It allows you to manage permissions for all the users in your group simultaneously by attaching policies to the group rather than individual users.

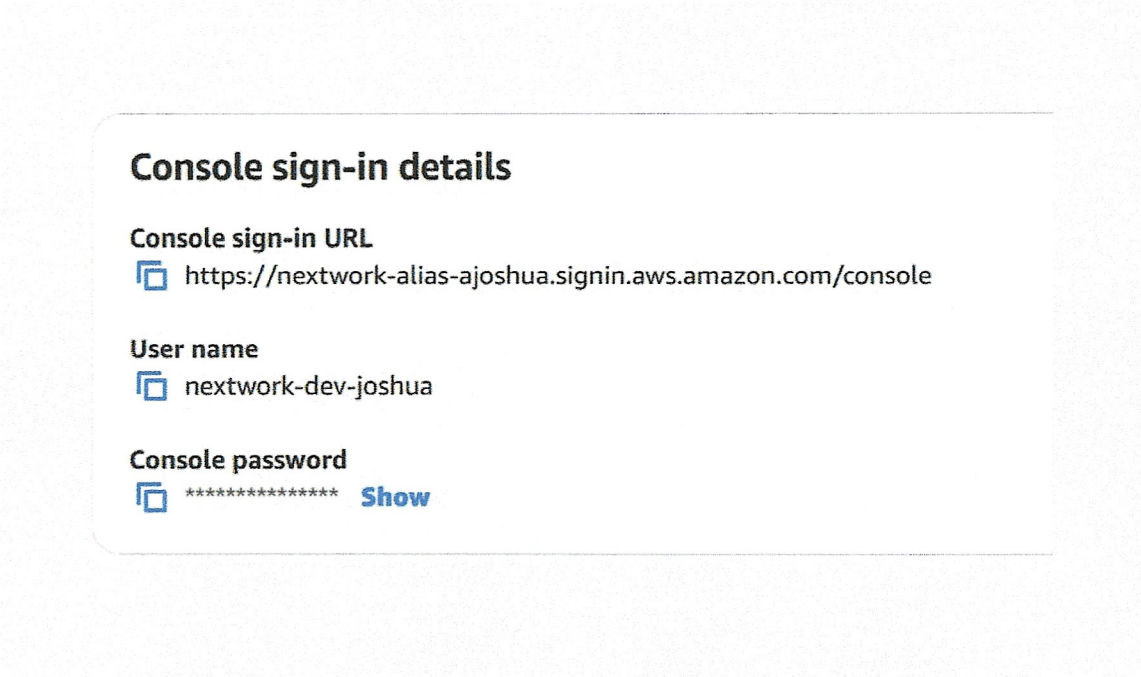
I attached the policy I created to a user group (nextwork-dev-group), which grants them the permissions associated with that group.

Joshua Lopez
NextWork Student

Logging in as an IAM User

The first way is via a unique console log-in URL for their account. The second way is creating an alias to make remembering and sharing the AWS console's (alias) login URL easier.

Once I logged in as my IAM user, I noticed some dashboard panels had access denied. This was because the IAM User is not authorized to access those functionalities.



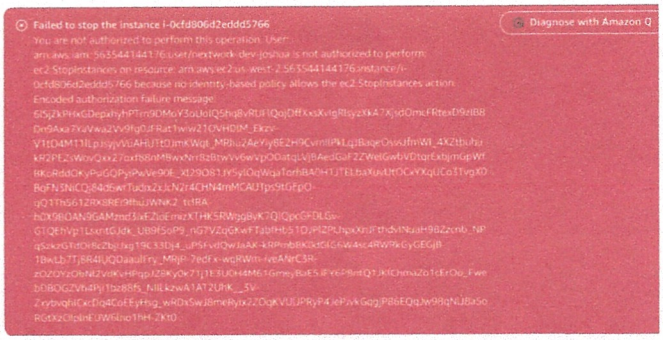
Joshua Lopez
NextWork Student

Testing IAM Policies

I tested my JSON IAM policy by trying to stop the production and dev instances.

Stopping the production instance

When I tried to stop the production instance, I received a red banner that said we were not authorized to do so.

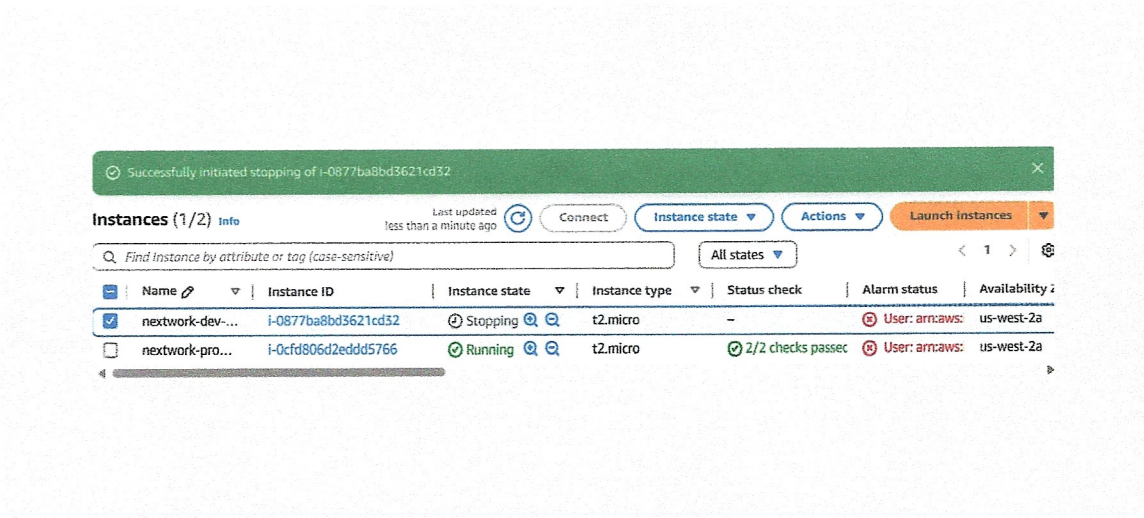


Joshua Lopez
NextWork Student

Testing IAM Policies

Stopping the development instance

Next, when I tried to stop the development instance, I received a success banner initiating the stop process. This was because my permission policy allowed unrestricted access to the dev environment.



Joshua Lopez
NextWork Student

The IAM Policy Simulator

IAM Policy Simulator is used to validate policies without affecting your actual AWS resources.

How I used the simulator

I set up a simulation for stopping instances. The results were permission denied because the simulation resource is set to ' * ' (all). I had to adjust the environment to specify dev only, not all ec2 instances.