nextwork.org

Cloud Security with AWS IAM

SH shravanirg03@gmail.com

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": "*"
    }
  ]
}
```

SH
shravanirg03@gmail.com
NextWork Student
NextWork.org

Introducing today's project!

What is AWS IAM?

Services I used were EC2 instances, IAM Policies, IAM Users and User Groups and AWS Account Alias. Key concepts I learned include access control, group-based permissions, resource restrictions, and simplified login.

How I'm using AWS IAM in this project

This project took me approximately 1.5 hours to complete. The most challenging part was understanding AWS Account Aliases and logins. It was most rewarding to see the Policy work successfully during testing.

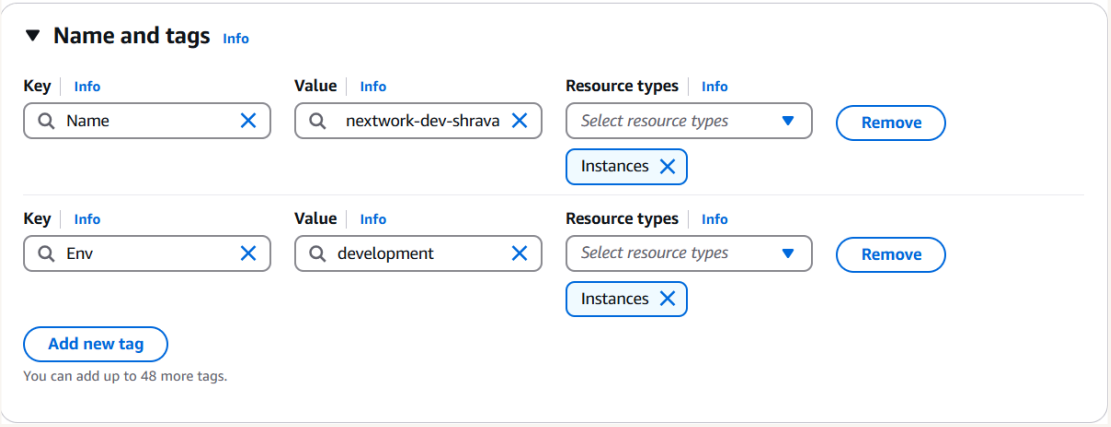
One thing I didn't expect...

I chose to do this project today because I wanted to start with a project that covered the basics of Cloud Security. Something that would make learning with NextWork even better is if there are more free projects.

Tags

Tags are like labels that you can attach to AWS resources for organization. They help the AWS account users to identify and manage their resources. Tags are useful for grouping mass management and applying security policies.

The tag I've used on my EC2 instances is called Env. The value I've assigned for my instances are production, and development. This represents two different environments that I am using to demonstrate the IAM project.



SH
shravanirg03@gmail.com
NextWork Student
NextWork.org

IAM Policies

IAM Policies are rules that help to allow/deny users'/resources' permissions to perform set actions to the AWS Account's resources. It's about giving permissions to IAM users, groups, or roles, defining what they can/can't do on certain resources.

The policy I set up

In the project, I've set up a policy using the JSON editor. I have created a Policy that allows EC2-related actions to EC2 instances having the "Env" tag "developement". It also denies actions to be performed for instances with the "production" tag.

I've created a policy that allows the IAM user to access the EC2 instance with the "development" tag and perform actions related to it. The policy denies the creation and deletion action to be performed for the EC2 instance with the "production" tag.

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

Effect: Specifies whether the policy allows / denies the specified actions. Action: Defines the specific AWS service operations that are permitted or restricted. Resource: Identifies the AWS resources on which the policy applies.

My JSON Policy

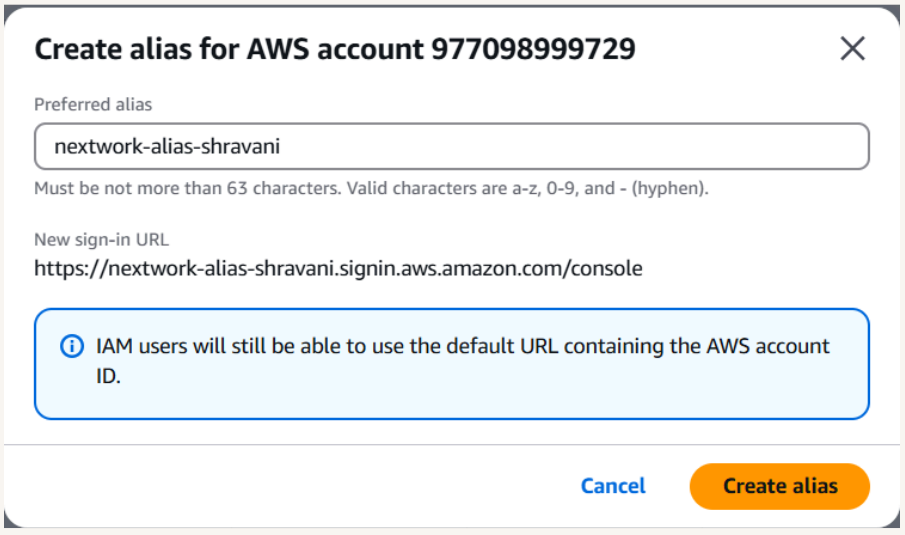
Policy editor

```
{
  "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

An account alias is a custom name that can be assigned to an AWS account, replacing the account ID in the sign-in URL. This allows users to use the alias instead of the 12-digit account ID when signing in to the AWS Management Console.

Creating an account alias took me less than a minute. Now, my new AWS console sign-in URL is https://nextwork-alias-shravani.signin.aws.amazon.com/console



SH
shravanirg03@gmail.com
NextWork Student
NextWork.org

IAM Users and User Groups

Users

IAM users are other log-ins/people who have access to my AWS account. These users are creatd by me using the AWS IAM services. I can designate my IAM users' access to my AWS Account's resources/services.

User Groups

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

I attached the policy I created to this user group, which means the policy is applied to all the users in that specific IAM group. All the IAM users residing in the IAM group will follow the rules stated in the policy.

Logging in as an IAM User

The two ways to share a new user's sign-in details are: Emailing sign-in instructions to the user. Downloading a .csv file containing the credentials.

Once I logged in as my IAM user, I noticed that many panels displayed "Access Denied". This highlighted the difference between the root user and an IAM user, as the root user has unrestricted access to AWS resources.

SH
shravanirg03@gmail.com
NextWork Student

Testing IAM Policies

I tested my JSON IAM policy by attempting to stop both the development and production instances, triggering the "Stop Instance" action.

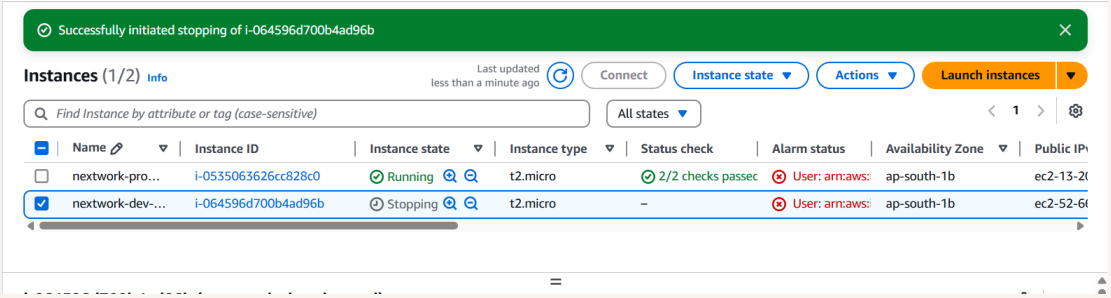
Stopping the production instance

When I tried to stop the production instance, an error message appeared, stating that I was not authorized. This happened because the IAM policy denied the IAM user access to stop the production instance.

Testing IAM Policies

Stopping the development instance

Next, when I tried to stop the development instance, it was successful. This was because the policy I created allowed all EC2 operations on instances with the "Env" tag set to "development".



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for more projects