nextwork.org

Cloud Security with AWS IAM

RA **Ranjith D B**

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": "*"
    }
  ]
}
```

RA
Ranjith D B
NextWork Student

Introducing today's project!

What is AWS IAM?

AWS IAM (Identity and Access Management) is a service that helps securely control access to AWS resources. It allows users to manage permissions, ensuring that only authorized individuals can perform specific actions on AWS services.

How I'm using AWS IAM in this project

I used AWS IAM to create a policy, user groups, and an IAM user with limited permissions, restricting access to production while allowing actions on the development environment.

One thing I didn't expect...

One thing I didn’t expect was how detailed IAM policies can be, especially when using conditions to enforce specific rules like resource tagging.

This project took me...

This project took me about an hour, including setting up policies, creating users and groups, and testing access controls in the AWS Management Console.

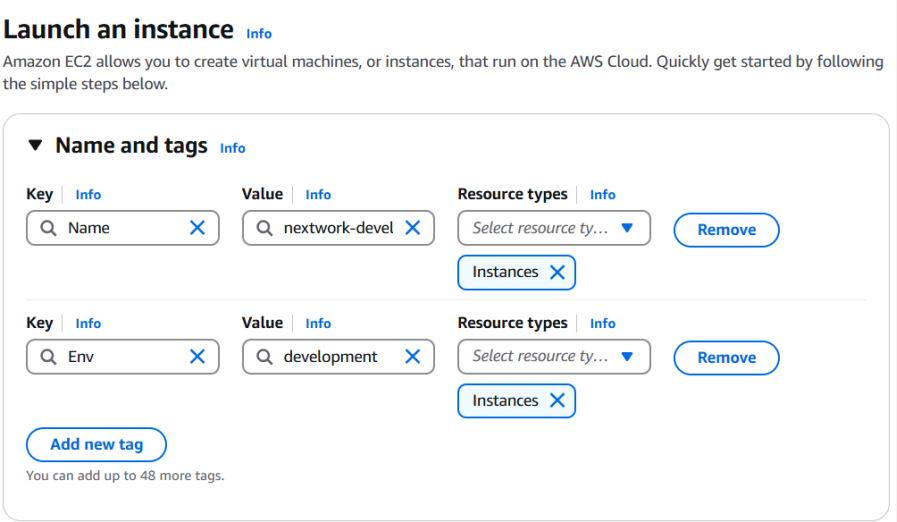
RA

Ranjith D B
NextWork Student

Tags

Tags are like sticky notes for AWS resources! They help organize, track costs, and manage access by assigning key-value labels to your cloud stuff.

The tag I've used on my EC2 instances is called "Env". The values I've assigned for my instances are: nextwork-production-ranjith → Env: production nextwork-development-ranjith → Env: development



RA
Ranjith D B
NextWork Student

IAM Policies

IAM Policies are rules that define permissions for users, groups, or roles, specifying what actions they can perform on AWS resources.

The policy I set up

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

I've created a policy that allows full EC2 access for instances tagged with "Env = development," permits describing any EC2 instance, and denies the ability to create or delete tags.

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: Effect: Specifies whether the action is allowed or denied. Action: Defines the AWS actions the policy applies to. Resource: Identifies which AWS resources the policy affects.

My JSON Policy

Policy editor

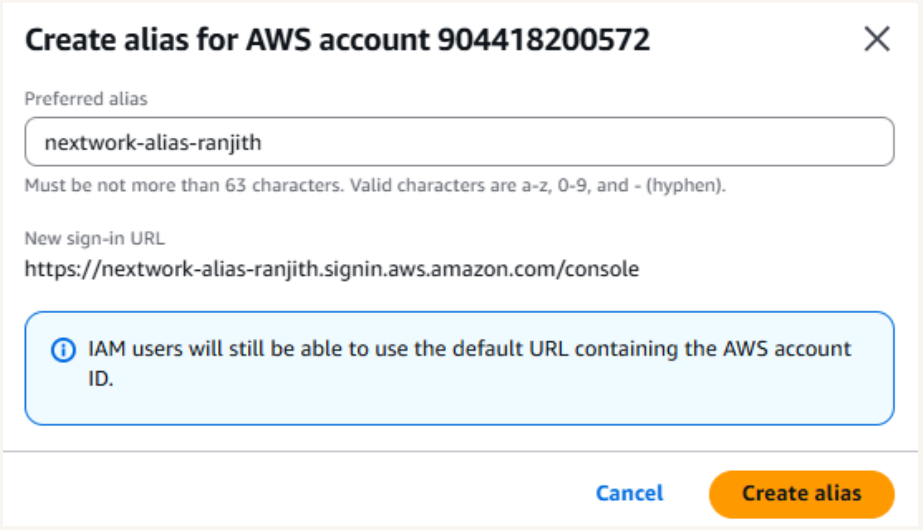
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Env": "development"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:Describe*",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

RA
Ranjith D B
NextWork Student

Account Alias

An account alias is a user-friendly name that replaces your AWS account ID in the sign-in URL, making it easier for users to log in without remembering a long numeric ID.

Creating an account alias took me a few seconds. Now, my new AWS console sign-in URL is https://nextwork-alias-ranjith.signin.aws.amazon.com/console



RA
Ranjith D B
NextWork Student

IAM Users and User Groups

Users

IAM users are individual identities in AWS that have specific credentials and permissions, allowing them to securely access AWS resources based on assigned policies.

User Groups

IAM user groups are collections of IAM users that simplify permission management by allowing policies to be assigned to multiple users at once instead of individually.

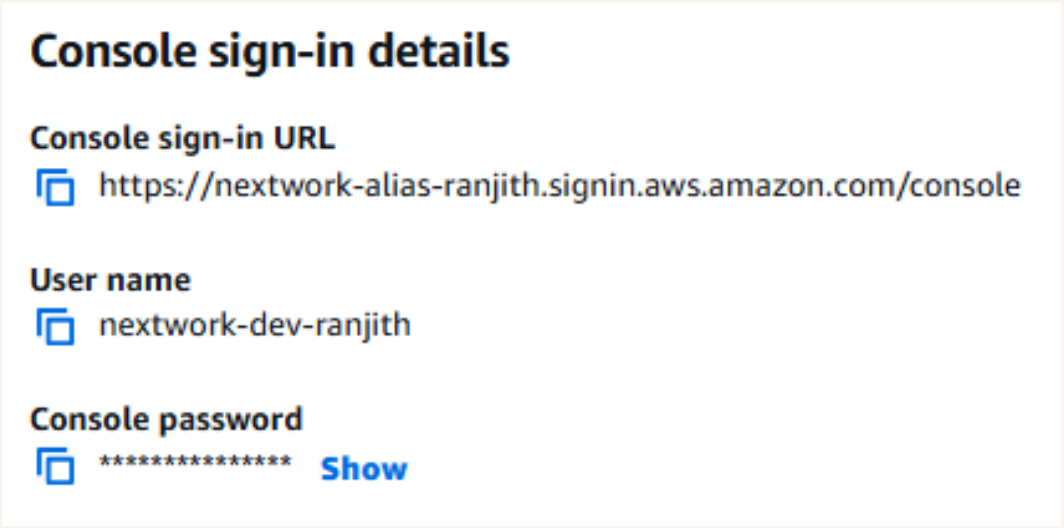
I attached the policy I created to this user group, which means all users in nextwork-dev-group automatically inherit permissions to access the development environment but not the production environment.

RA
Ranjith D B
NextWork Student

Logging in as an IAM User

The first way is by downloading the sign-in credentials CSV file. The second way is by manually sharing the sign-in URL, username, and temporary password with the user.

Once I logged in as my IAM user, I noticed that some dashboard panels showed "Access Denied." This was because the IAM policy only allowed access to the development instance while restricting access to production and certain actions.



RA
Ranjith D B
NextWork Student

Testing IAM Policies

I tested my JSON IAM policy by attempting to stop both my production and development EC2 instances to verify that the policy correctly restricts access to the production instance while allowing control over the development instance.

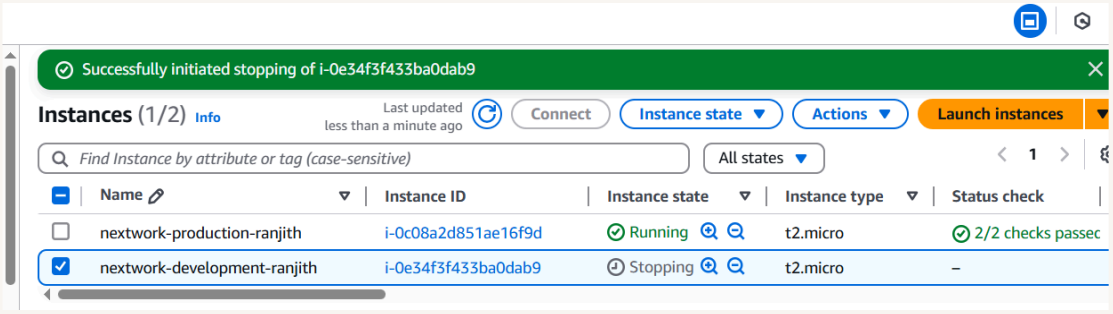
Stopping the production instance

When I tried to stop the production instance, I received an "Access Denied" error. This was because the IAM policy only allows actions on instances tagged with "Env = development" and denies unauthorized actions on the production instance.

Testing IAM Policies

Stopping the development instance

Next, when I tried to stop the development instance, the action was successful. This was because the IAM policy explicitly allows actions on EC2 instances tagged with "Env = development," granting the necessary permissions.



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for more projects