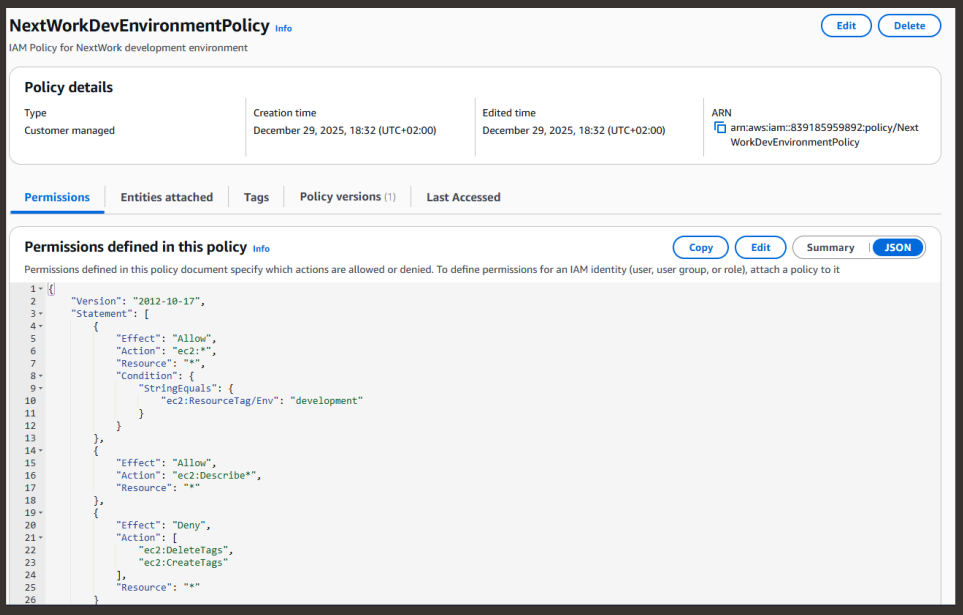
nextwork.org

Cloud Security with AWS IAM

Andrei Boboc



Andrei Boboc
NextWork Student

Introducing Today's Project!

Project overview

In this project, I will demonstrate how to set up a secure and well-structured AWS environment by provisioning EC2 instances and configuring IAM users, groups, and policies following best practices. The goal is to apply the principle of least privilege, improve account security, and ensure controlled access for different user roles.

Tools and concepts

The main services I used were Amazon EC2 for compute resources and AWS IAM for identity and access management. I also worked with IAM tools such as IAM policies, IAM user groups, and the IAM Policy Simulator to design, test, and validate permissions. Key concepts I learned include tag-based access control, the principle of least privilege, environment separation (development vs production), and how to safely validate IAM policies using simulations before applying them in real environments.

Project reflection

This project took me approximately 1 hour to complete. The most challenging part was correctly configuring and testing the tag-based IAM conditions, especially understanding why certain actions were initially denied in the policy simulator.

Andrei Boboc
NextWork Student

Tags

What I did in this step

In this step, I will launch two Amazon EC2 instances to increase NextWork's computing capacity. This allows the environment to handle additional workloads and simulates a real-world scenario where infrastructure needs to be scaled to meet growing demand. I am doing this to demonstrate how compute resources can be provisioned quickly and reliably in AWS, as well as to show an understanding of basic scaling concepts that are essential in DevOps and cloud operations.

Understanding tags

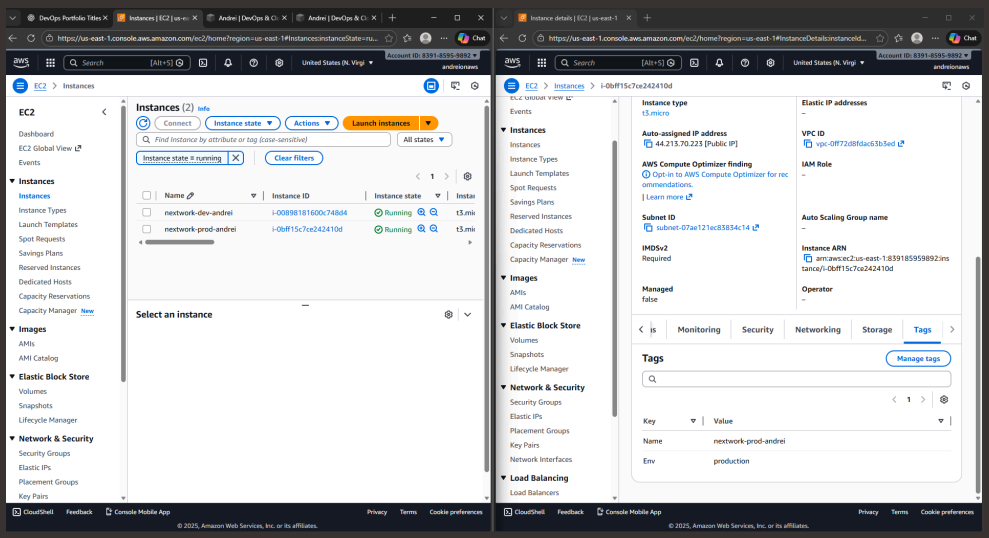
Tags are key–value pairs that are used to label and organize AWS resources such as EC2 instances, volumes, and security components. They provide metadata that helps identify the purpose, owner, environment, or application associated with each resource. Tags are useful for resource management, cost allocation, and operational efficiency. They make it easier to filter and group resources, track spending by team or project, and apply automation or policies based on specific tag values. In DevOps environments, consistent tagging is essential for governance, monitoring, and maintaining scalable cloud infrastructure.

My tag configuration

The tag I've used on my EC2 instances is called Environment.

Andrei Boboc
NextWork Student
nextwork.org

I assigned the value prod to one instance to represent the production environment, and development to the second instance to represent the development environment. These tags help clearly distinguish resources by environment, making it easier to manage, monitor, and apply environment-specific policies or automation in a real-world DevOps setup.



Andrei Boboc
NextWork Student

IAM Policies

What I did in this step

In this step, I will create an IAM policy that grants access only to the development EC2 instance. This ensures that permissions are limited to the resources required and prevents unnecessary access to production systems.

Understanding IAM policies

IAM policies define what actions are allowed or denied on specific AWS resources. They control permissions by specifying who can access a resource, which actions they can perform, and under what conditions.

The policy I set up

For this project, I set up the IAM policy using JSON. This allowed me to precisely define permissions and apply conditional logic based on resource tags.

Policy effect

I’ve created a policy that allows full EC2 access only to instances tagged with Env=development, while still permitting read-only Describe actions across all EC2 resources.

This ensures users can view the overall EC2 environment without being able to modify production resources. At the same time, the policy explicitly denies the ability to create or delete tags, preventing users from changing resource tags to escalate their privileges. Overall, this policy enforces environment-based access control, improves security, and follows the principle of least privilege.

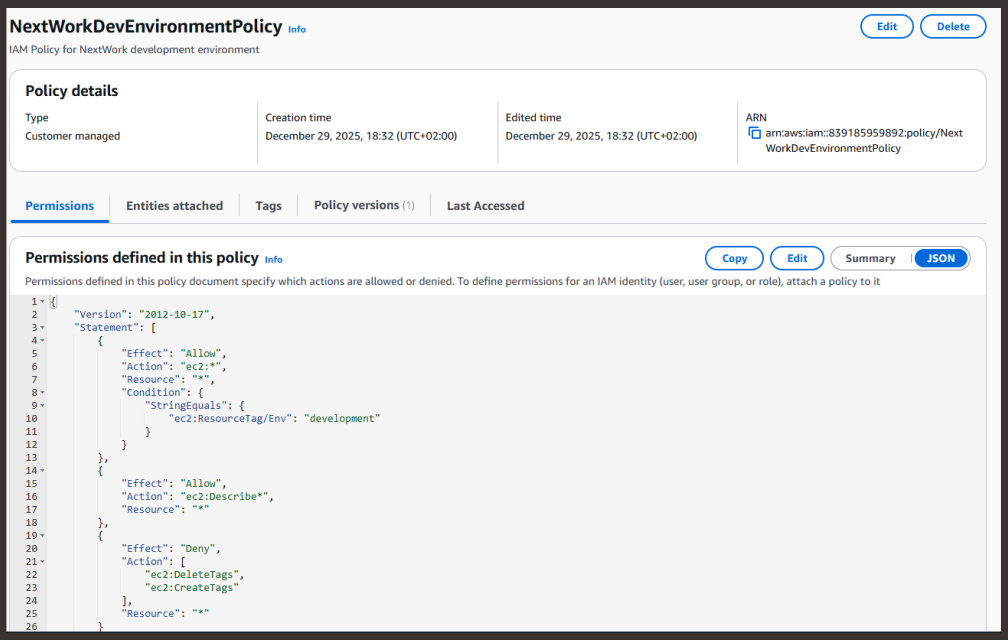
Understanding Effect, Action, and Resource

The Effect element defines whether a permission is allowed or denied. It specifies the outcome of the policy statement when the conditions are met. The Action element lists the specific API actions that are allowed or denied, such as starting an EC2 instance or describing resources. The Resource element identifies which AWS resources the actions apply to, either by specifying individual resource ARNs or using a wildcard to include multiple resources.

Andrei Boboc
NextWork Student

nextwork.org

My JSON Policy



Andrei Boboc
NextWork Student

Account Alias

What I did in this step

In this step, I will create an AWS Account Alias to simplify the login process for users.

Understanding account aliases

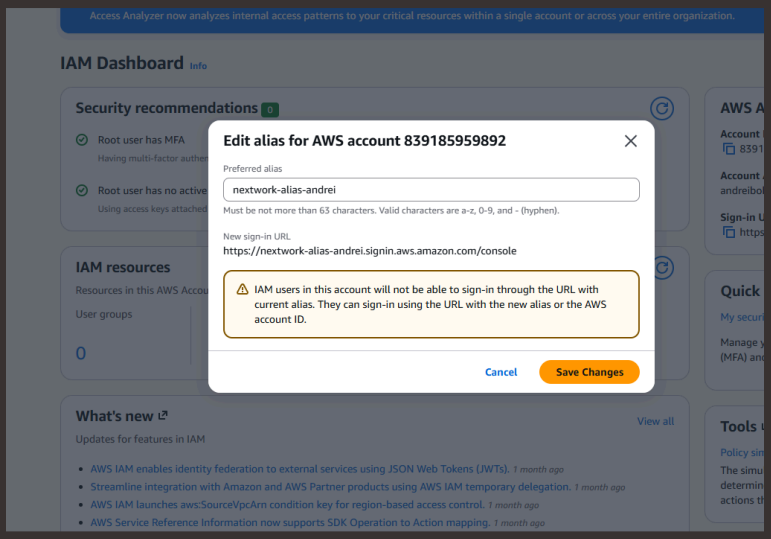
An account alias replaces the numeric AWS account ID with a custom, human-readable name, making it easier for users to access the AWS Management Console.

Setting up my account alias

Creating an account alias took me one minute, now my Alias URL is: https://nextwork-alias-andrei.signin.aws.amazon.com/console

Andrei Boboc
NextWork Student

nextwork.org



Andrei Boboc
NextWork Student

IAM Users and User Groups

What I did in this step

In this step, I will create a dedicated IAM group for all NextWork interns and assign the appropriate permissions to the group. I will also create a dedicated IAM user for a new intern and add that user to the group.

Understanding user groups

IAM user groups are collections of IAM users that share the same permissions. Instead of assigning policies to individual users, permissions are attached to groups, and users inherit those permissions by being added to the group.

Attaching policies to user groups

I attached the policy I created to this user group, which means that all users who are members of the group automatically inherit the permissions defined in the policy.

Understanding IAM users

IAM users are individual identities created within an AWS account that represent a person or application. Each IAM user has its own credentials and is granted permissions through IAM policies, either directly or by being added to IAM groups.

Andrei Boboc
NextWork Student

Logging in as an IAM User

Sharing sign-in details

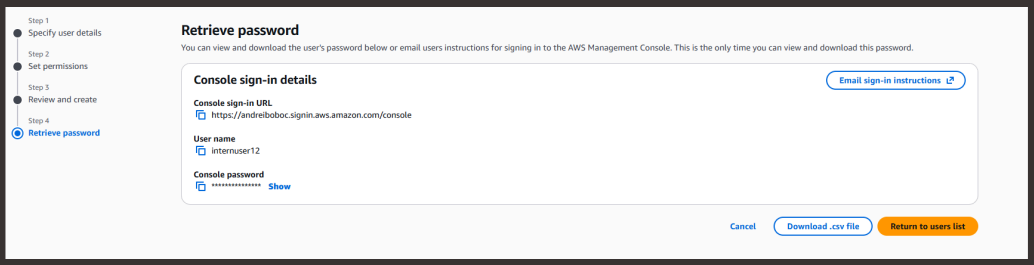
The first way is by downloading the .csv file generated by AWS when the IAM user is created, which contains the username, console sign-in URL, and temporary password. The second way is by manually sharing the console sign-in URL, along with the username and temporary password, through a secure communication channel. In both cases, the user is required to change their password on first login to maintain security.

Observations from the IAM user dashboard

Once I logged in as the IAM user, I noticed that some dashboard panels were already showing “Access denied.” This indicated that the user did not have permission to view or manage certain AWS services or resources.

Andrei Boboc
NextWork Student

nextwork.org



Andrei Boboc
NextWork Student

Testing IAM Policies

What I did in this step

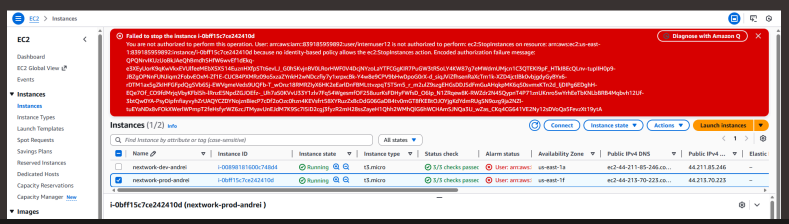
In this step, I will log in to AWS using the intern's IAM user to validate the assigned permissions. I will test the intern's access to both the development and production EC2 instances to confirm that access is granted only where intended.

Testing policy actions

I tested my JSON IAM policy by attempting to stop both the development and production EC2 instances using the intern's IAM user. The stop action succeeded on the development instance but was blocked on the production instance. This confirmed that the policy correctly allows actions only on resources tagged with Env=development and prevents unauthorized actions on production infrastructure.

Stopping the production instance

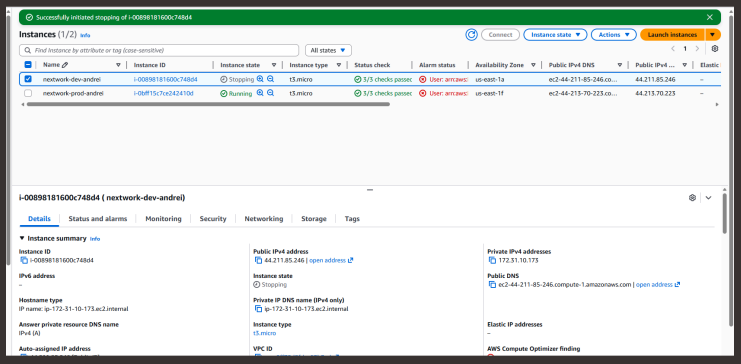
When I tried to stop the production EC2 instance, the action failed and AWS returned an "Access denied" error. The console indicated that the IAM user was not authorized to perform the ec2:StopInstances action on the production resource. This happened because the IAM policy only allows EC2 actions on instances tagged with Env=development. Since the production instance is tagged as Env=prod, the policy correctly blocked the action, preventing unauthorized changes to production infrastructure.



Andrei Boboc
NextWork Student

Stopping the development instance

Next, when I tried to stop the development EC2 instance, the action was successful and the instance transitioned to a stopped state. This was because the IAM policy explicitly allows EC2 actions on resources tagged with Env=development, confirming that the policy behaves as intended and correctly differentiates between development and production environments.



Andrei Boboc
NextWork Student

IAM Policy Simulator

To extend my project, I’m going to use the IAM Policy Simulator to test and validate the behavior of my IAM policy without making changes to actual AWS resources. This allows me to safely verify which actions are allowed or denied for the IAM user and confirm that permissions are enforced as expected.

Understanding the IAM Policy Simulator

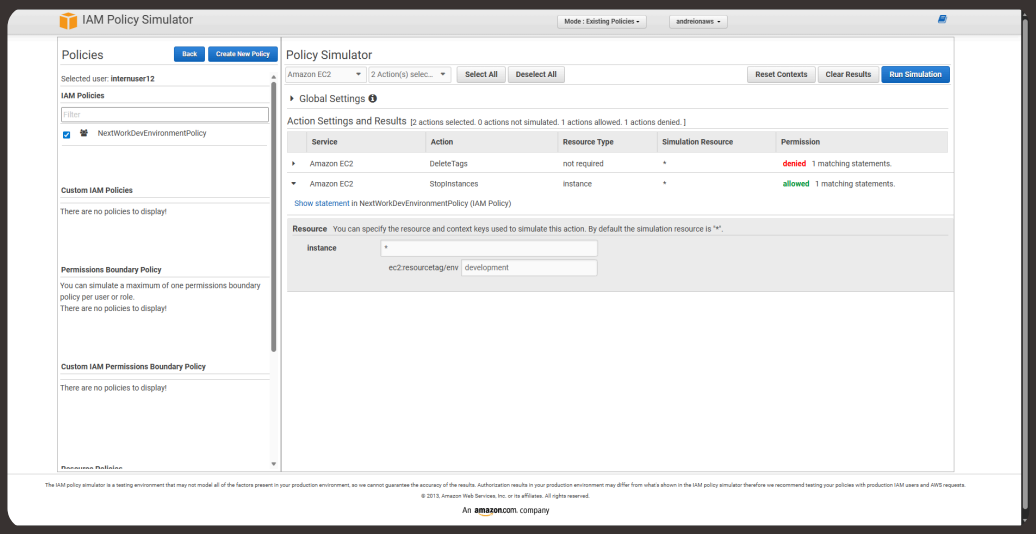
The IAM Policy Simulator is a tool used to test and evaluate IAM policies by simulating API actions without actually performing them on AWS resources. It shows whether specific actions would be allowed or denied based on the current policy configuration.

How I used the simulator

Initially, both simulated actions were denied, including ec2:StopInstances, because the instance did not yet have the required Env=development tag defined in the simulation context. Since the IAM policy relies on resource tags for conditional access, the missing tag caused the policy conditions to fail. After adding the Env=development tag to the simulation, the results updated as expected: the ec2:StopInstances action was allowed, while tag modification actions such as ec2:DeleteTags remained denied. This confirmed that the policy behaves correctly when the required resource tags are present.

Andrei Boboc
NextWork Student

nextwork.org



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects