NEXTWORK

MANAGING PERMISSIONS WITH AWS IAM

Rasha M
@Badry2022
Rasha M.

WHAT IS AWS IAM?

What it does:

- Identity and Access Management service used to manage user access to AWS resources.

Why it's useful:

- IAM secures your organization's digital identity by controlling user access and preventing data breaches.
- IAM helps you comply with regulations and automates tasks like access certifications.
- IAM improves security by creating unique identities, detecting suspicious activity, and integrating with cloud services.

How I'm using it in today's project:

In this project, I used IAM to:

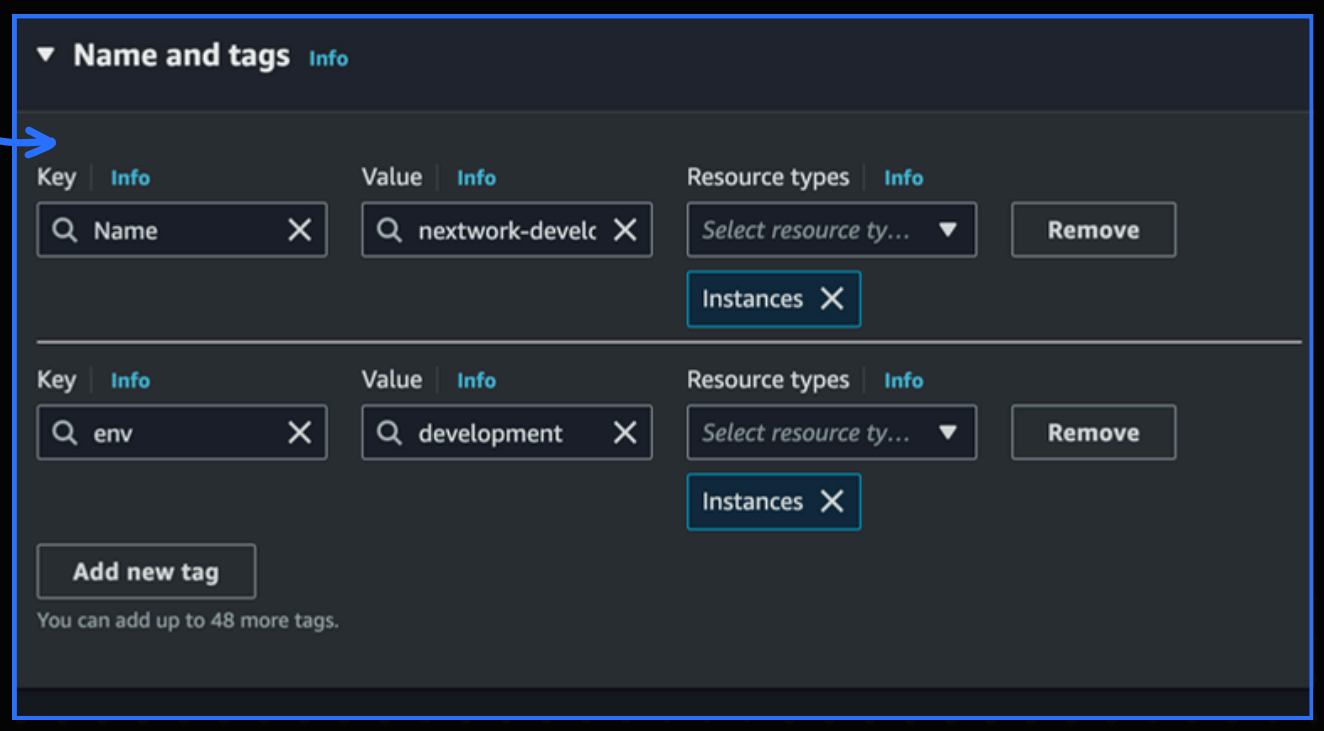
- Create an IAM policy that restricts access to EC2 instances based on tags.
- Create user groups to manage permissions for multiple users efficiently.
- Create IAM users and assign them to user groups to grant them specific permissions.

Rasha M
@Badry2022
Rasha M.

SETTING UP TAGS

- I've set up two EC2 instances to test the effectiveness of the permission settings I'll set up in AWS IAM. I've used tags to label them.
- Tags are keywords or labels that you can attach to AWS resources for easier organization. They act like categorization labels, similar to how you might tag files on your computer. This makes it easier to find specific resources, filter large groups of resources, and automate tasks based on tag criteria.
- The tag I've used on my EC2 instances is called Env. The value I've assigned for my instances are development and production. This way, I can easily distinguish between my development and production environments.

How the tags are set up for my EC2 instances



Rasha M
@Badry2022
Rasha M.

IAM POLICIES

- IAM Policies are essentially a set of rules that define who can do what with AWS resources. They act like a rulebook that specifies which users or groups have permissions to perform certain actions on specific AWS services.
- For this project, I've set up a policy using the JSON method.
- I've created a Policy that allows full access (ec2:* actions) to EC2 instances tagged with "Env=development" while denying creating or deleting tags for all resources.
- Here's a breakdown of the key components in a JSON Policy statement:
- Effect: This specifies whether the statement allows (Allow) or denies (Deny) an action.
- Action: This defines the specific action that can be performed on a resource. For example, "ec2:Stop" allows stopping an EC2 instance.
- Resource: This identifies the specific AWS service or resource that the action applies to. An asterisk (*) can be used as a wildcard to apply the action to all resources of that type.
  -

The policy I've set up in the IAM Policies page!

Policy editor

```
1  {
2
3  "Version": "2012-10-17",
4
5  "Statement": [
6
7  {
8
9  "Effect": "Allow",
10
11 "Action": "ec2:*",
12
13 "Resource": "*",
14
15 "Condition": {
16
17 "StringEquals": {
18
19 "ec2:ResourceTag/Env": "development"          }          }
20
21  {
22
23  "Effect": "Allow",
24
25 "Action": "ec2:Describe*",
26
27 "Resource": "*"       },
28
```

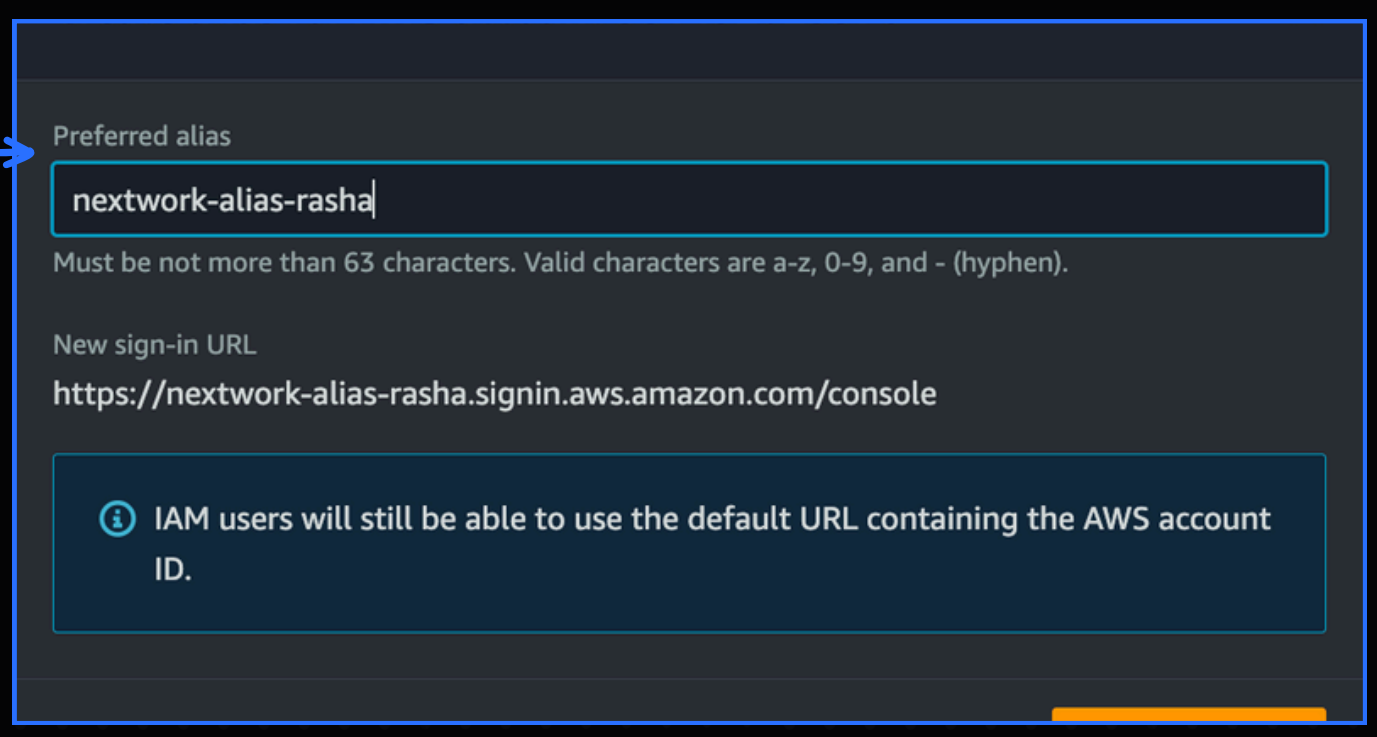
Rasha M
@Badry2022
Rasha M.

AWS ACCOUNT ALIAS

- When new users get onboarded onto my AWS account, they used to sign into a unique URL created for my account's long and cryptic Account ID. This can be cumbersome to remember and type in.
- An account alias is a user-friendly name that replaces your lengthy AWS account ID in the sign-in URL. It makes the sign-in process easier for new users and helps to avoid any confusion caused by the long account ID.
- Creating an account alias took me just a few seconds. It's a very quick and straightforward process.

Now, my new AWS console sign-in URL is :

You get to set up your own account alias name!



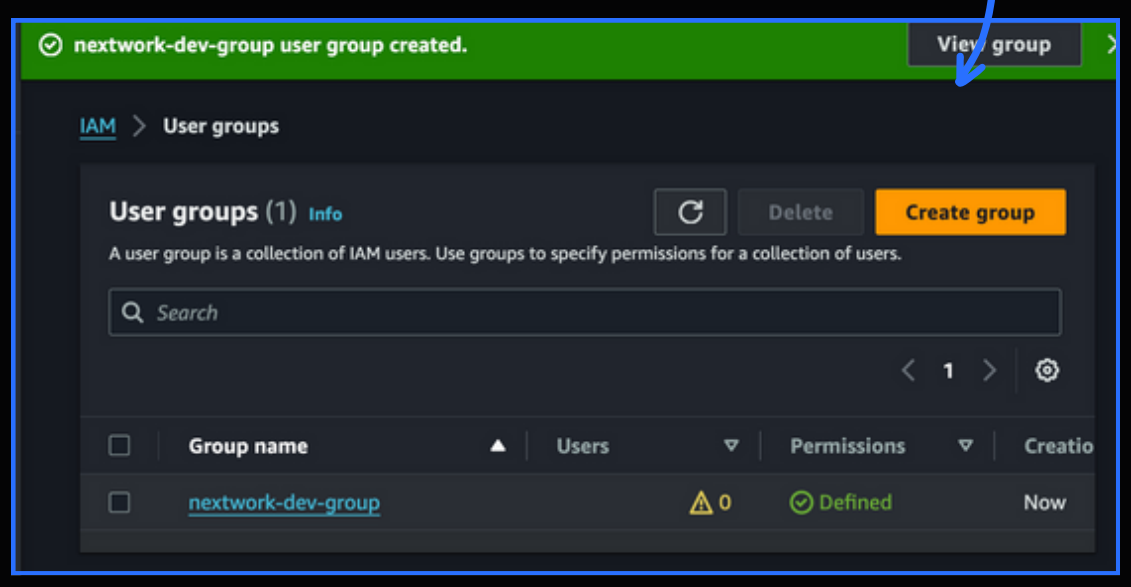
Rasha M
@Badry2022
Rasha M.

IAM USERS + USER GROUPS

- **IAM Users are the individual accounts that people use to access AWS resources. They act like individual identities within your AWS account, and you can assign specific permissions to each user.**

- **I** also created a User Group. User Groups are useful for managing permissions for multiple users efficiently. Instead of assigning permissions to each user individually, you can create a user group and assign permissions to the group. This lets you easily manage permissions for a group of users who all need the same level of access.
- My User Group is called network-dev-group

My User Group!



- I attached the Policy I created to this User Group, which means any users assigned to this group will inherit the permissions defined in the policy.

  - When I created a new User, I had to tick the checkbox for "Provide user access to the AWS Management Console" so the user can sign in to the AWS console and use the permissions assigned to their user group

  - Once my new user was set up, there were two ways I could share its sign-in details:

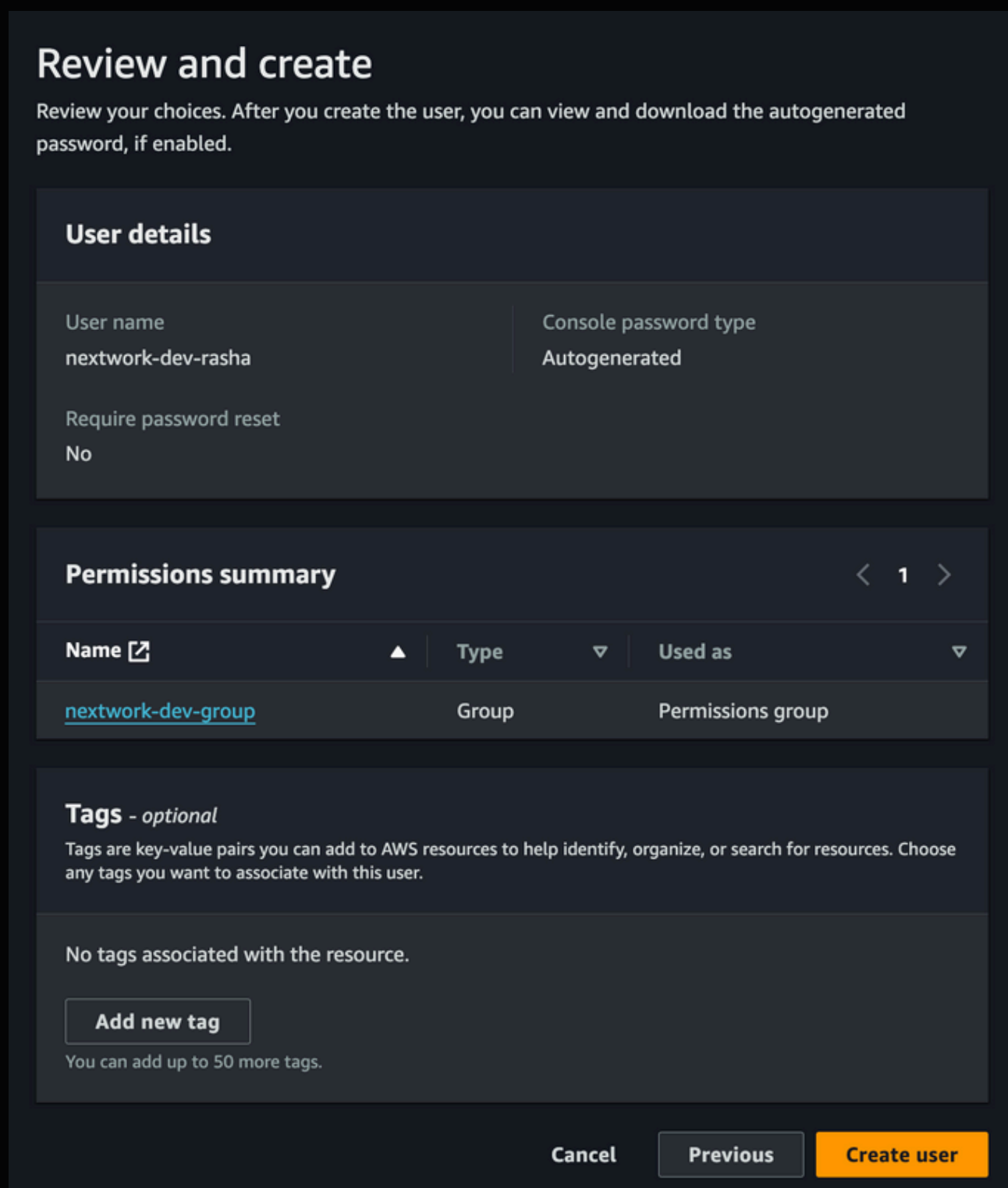
  - My new user had a unique sign-in URL!

My User's sign-in details!

Rasha M
@Badry2022
Rasha M.

IAM USER IN ACTION

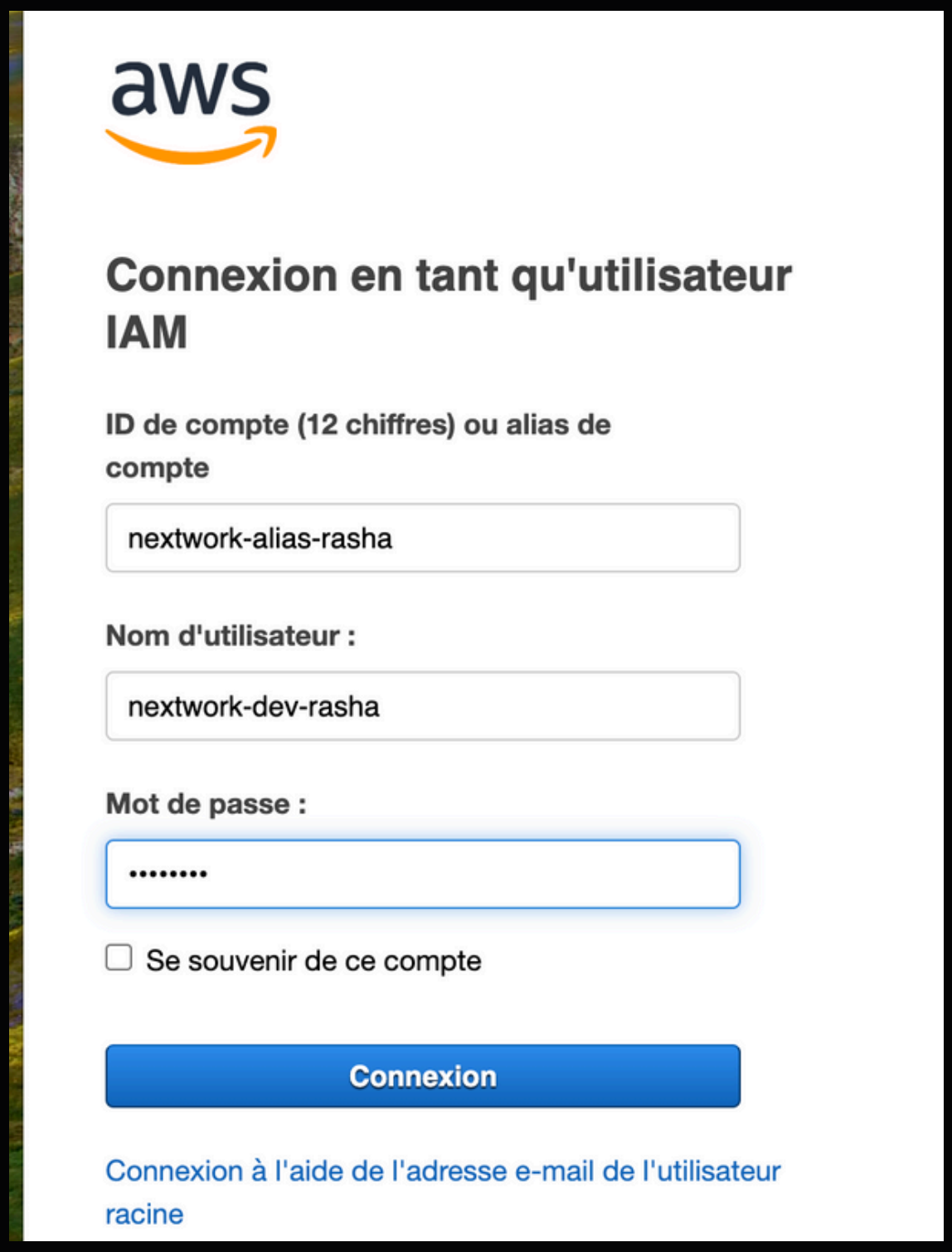
- Now with my IAM Policy, IAM User Group and IAM User all set up, let's put it all together! To do this, I logged into my AWS account as the new user.



Rasha M
@Badry2022
Rasha M.

IAM USER IN ACTION

- To log in as my IAM User, I opened a new incognito window (to avoid any conflicts with my own cookies or browsing data) and pasted the sign-in URL for my account alias:

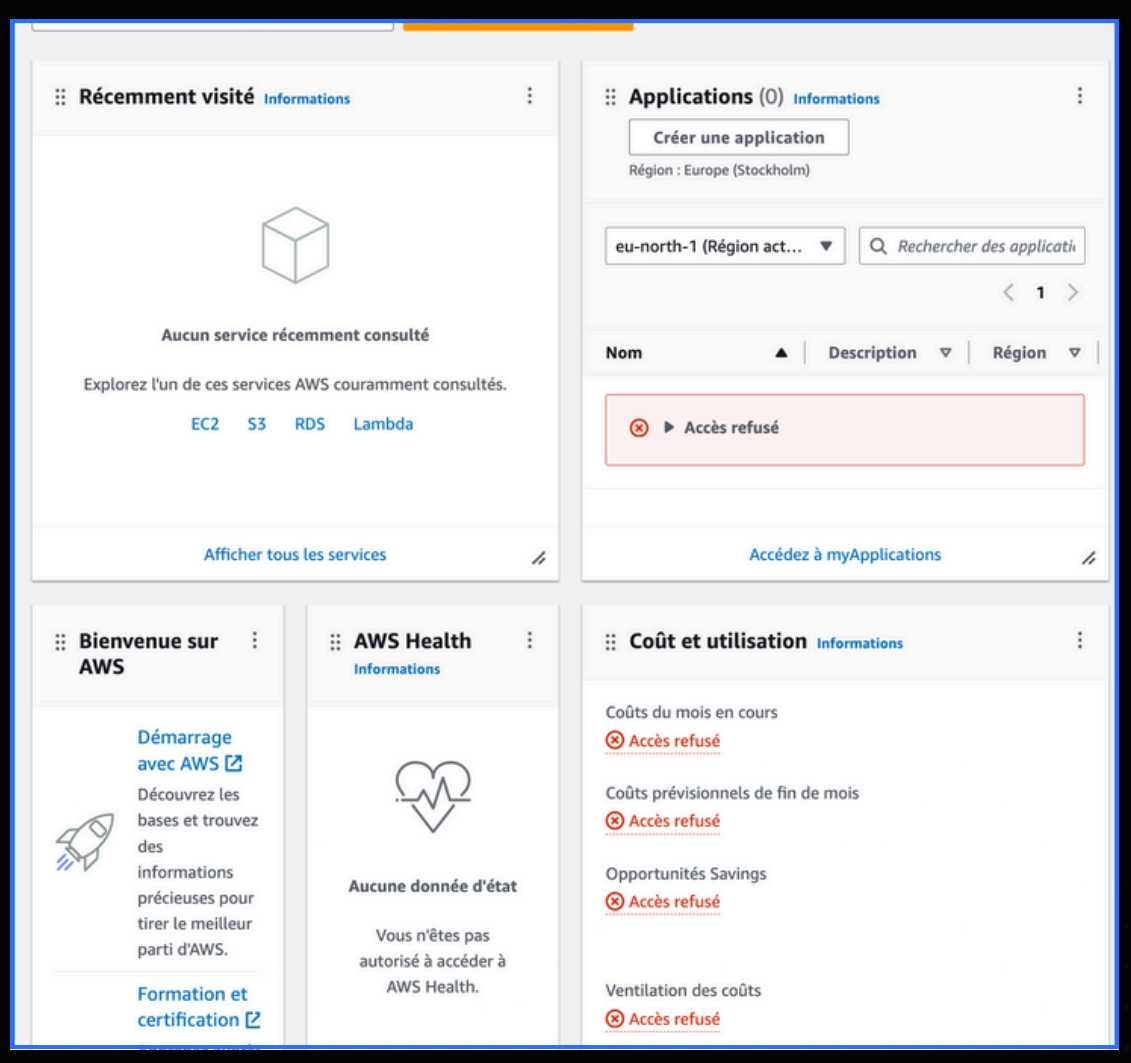


Rasha M
@Badry2022
Rasha M.

IAM USER IN ACTION

- Once I logged in, I noticed access denied messages on some of the dashboard panels. This is because the IAM policy I attached to the user group restricts access to specific resources based on tags. For example, the user might not see any options to manage production EC2 instances (Env=production) since the policy denies those actions. This confirms that the IAM policy and user group permissions are working as intended.

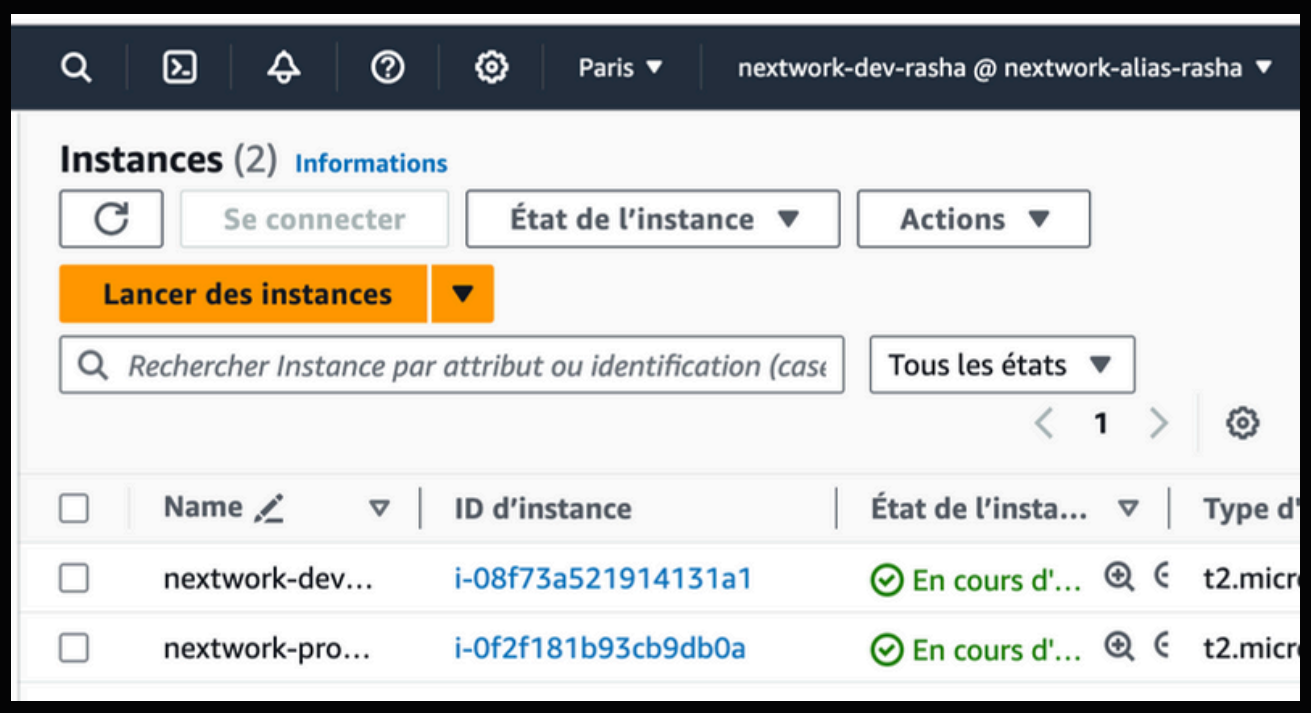
Some of my dashboard's panels showed access denied!

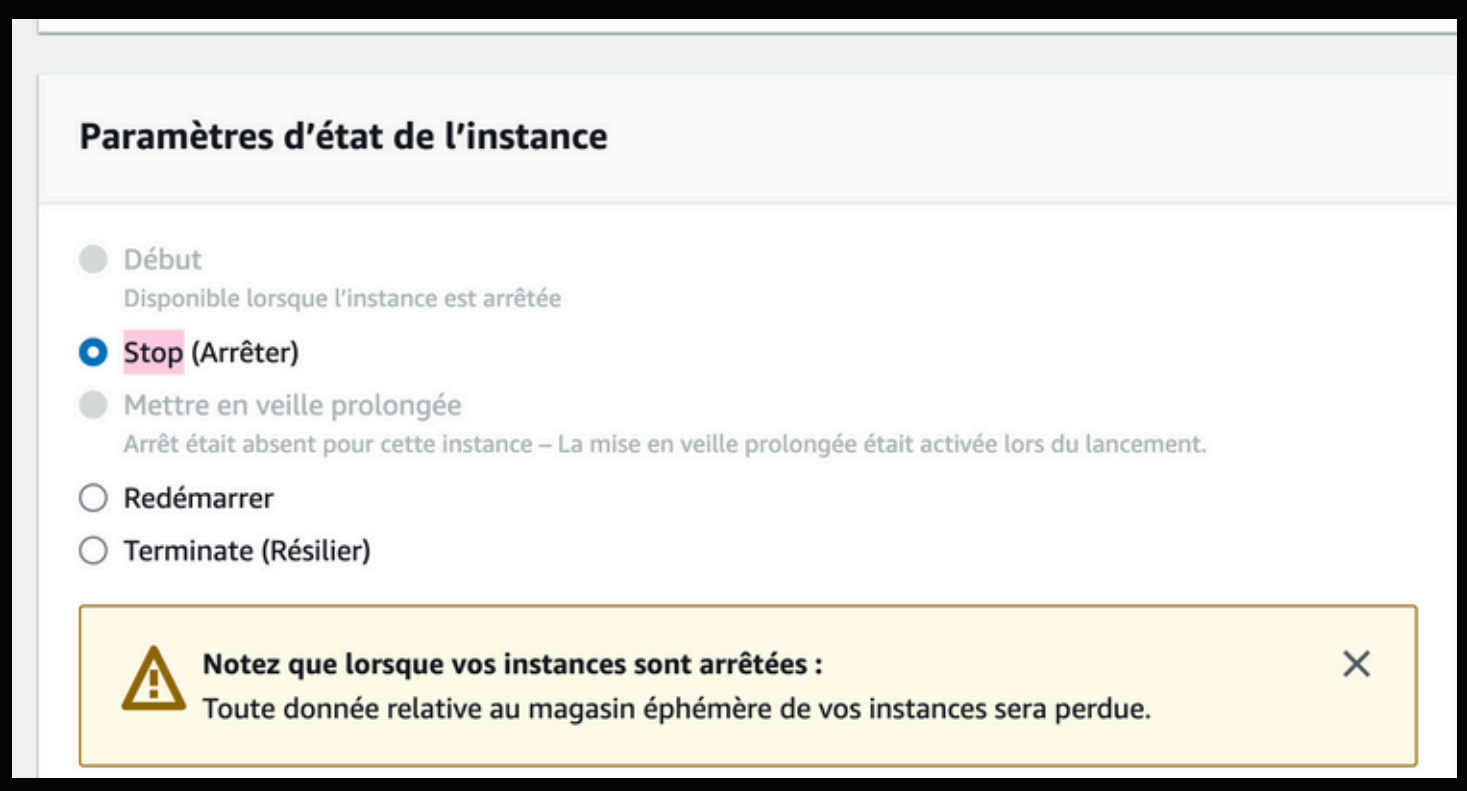


Rasha M
@Badry2022
Rasha M.

IAM POLICIES IN ACTION

- Then, I tested the JSON IAM policy I set up by trying to stop the EC2 instances.
- When I tried to stop the production instance (Env=production)



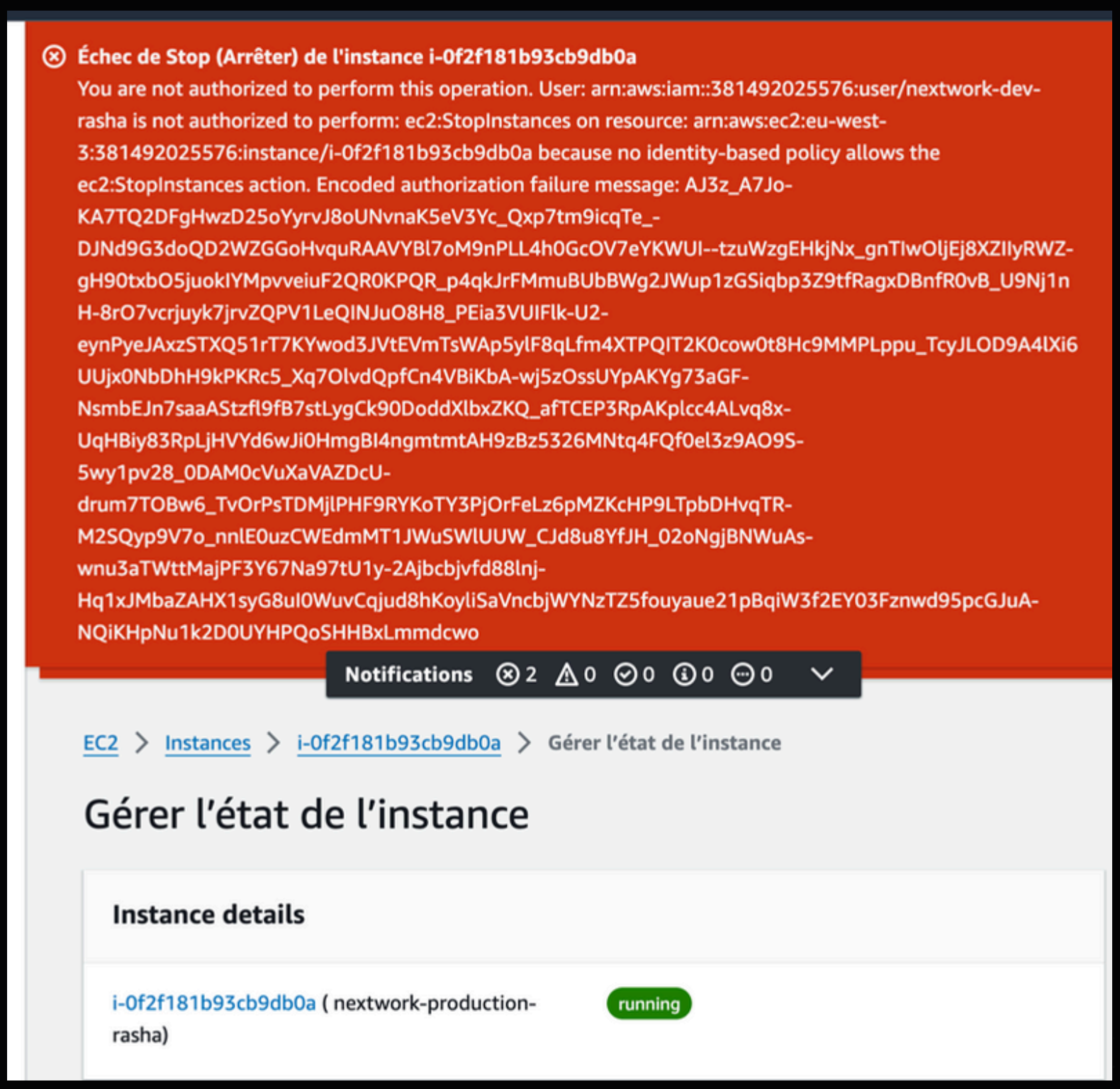


Rasha M
@Badry2022
Rasha M.

IAM POLICIES IN ACTION

I received an error message stating that "I am not authorized to perform the requested operation (ec2:StopInstances)". This confirms that the policy is working correctly, as it specifically denies the "ec2:Stop" action for resources with the "Env=production" tag

Woah! A red fail banner pops up if I stop the production instance

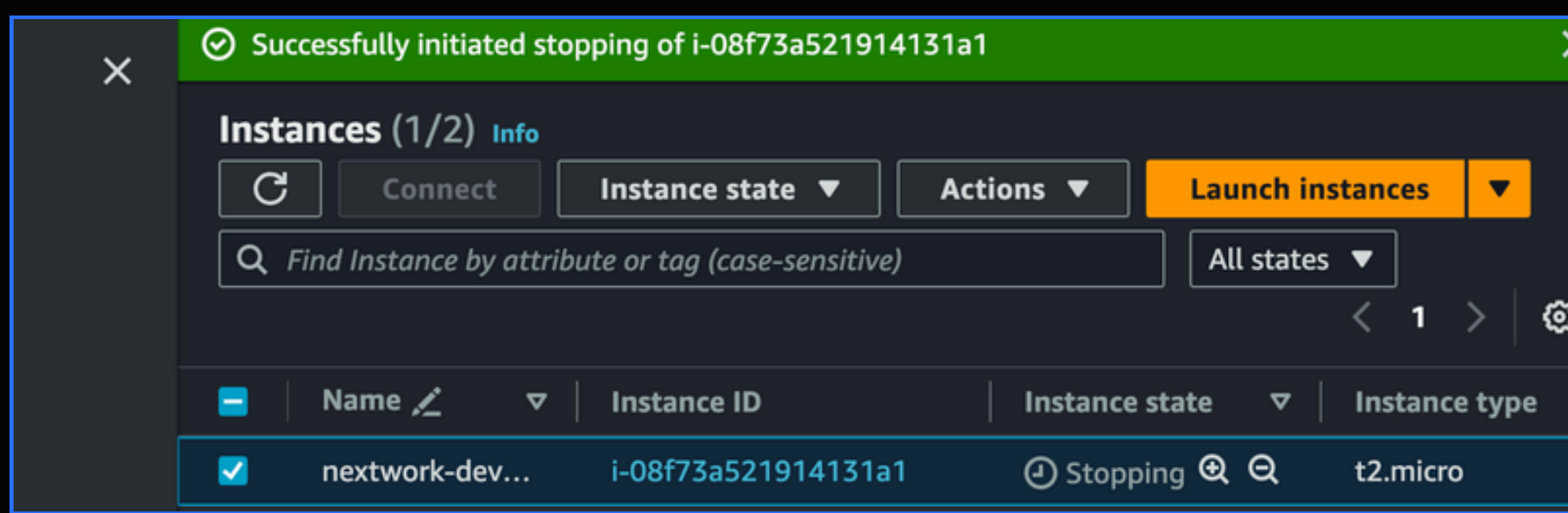


Rasha M
@Badry2022
Rasha M.

IAM POLICIES IN ACTION

- Next, when I tried to stop the development instance (Env=development), it worked successfully. The instance stopped without any errors. This is because the IAM policy I created allows the "ec2:Stop" action for resources with the "Env=development" tag. The policy grants full access (ec2:* actions) to EC2 instances tagged for development, allowing me to stop the instance as expected.

Phew! A green success banner pops up if I stop the development instance



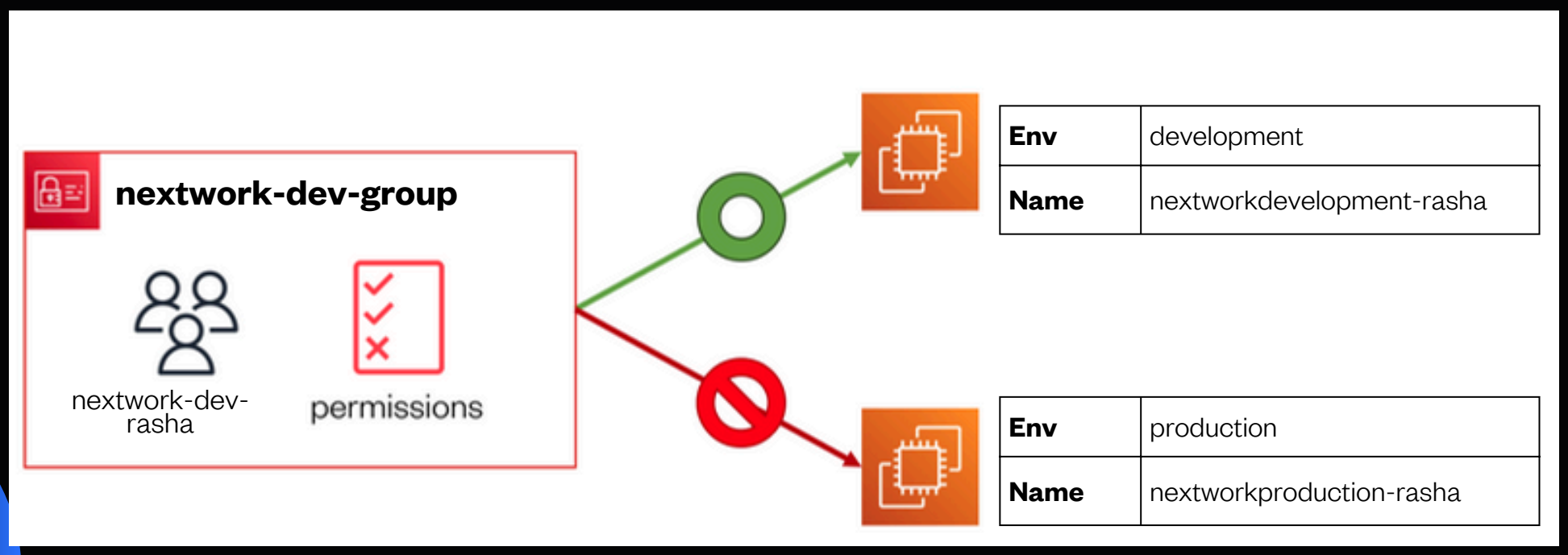
Rasha M
@Badry2022
Rasha M.

TO SUMMARISE

I created:

- An IAM User Group called nextwork-dev-group with defined permissions using an IAM Policy
- An IAM User called nextwork-dev-rasha that is added to the user group
- An EC2 instance with the Env tag development and Name nextworkdevelopment-rasha
- An EC2 instance with the Env tag production and Name nextworkproduction-rasha

The IAM setup grants nextwork-dev-rasha the necessary permissions to manage development resources while ensuring the security and stability of production resources.



Rasha M
@Badry2022
Rasha M.

My Key Learnings

01

What are IAM Policies?

IAM Policies are documents written in JSON that define what actions users can perform on AWS resources. They allow you to implement the principle of least privilege, granting users only the permissions they need for their tasks.

02

What are IAM Users? Why would you create one?

IAM Users are identities within your AWS account that allow people to securely access AWS resources. You create IAM Users for: Granting access to specific people, improving security by avoiding the root user for everyday tasks, managing access for applications that interact with AWS services

03

What are IAM User Groups? Why would you create one?

IAM User Groups streamline permission management, ensure consistent access within groups, and simplify team scaling by grouping users with shared needs.

04

What is an AWS Account Alias?

An AWS Account Alias is a user-friendly name you can give your AWS account ID. It makes your account easier to remember and identify when using the AWS Management Console or CLI.

05

With this project, I have gained a good foundation for managing access and security in your AWS environment.

Rasha M
@Badry2022
Rasha M.

Find this helpful?

Like this post

Leave a comment

Save for later

Let's connect!

pssst... if you want to get this free project guide and documentation template, **check out NextWork!**

Rasha M
@Badry2022
Rasha M.

Thanks NextWork for the free project guide!

NEXTWORK