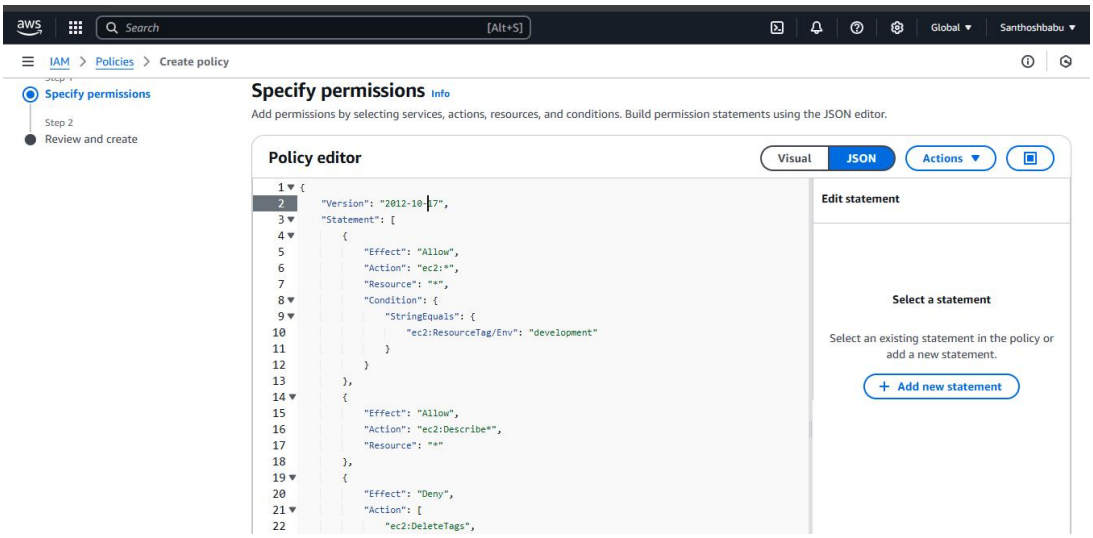
Cloud Security with AWS IAM



Introducing today's project!

What is AWS IAM?

AWS IAM allows you to securely manage access to AWS resources. It lets you create users, assign permissions, and define roles for fine-grained control, ensuring the principle of least privilege and enabling secure, scalable cloud operations.

How I'm using AWS IAM in this project

In today's project,IAM was used to manage user access by creating users, groups, and roles with specific permissions. Policies were applied to ensure least privilege access, securing resources and enabling safe automation and cross-account access.

One thing I didn't expect...

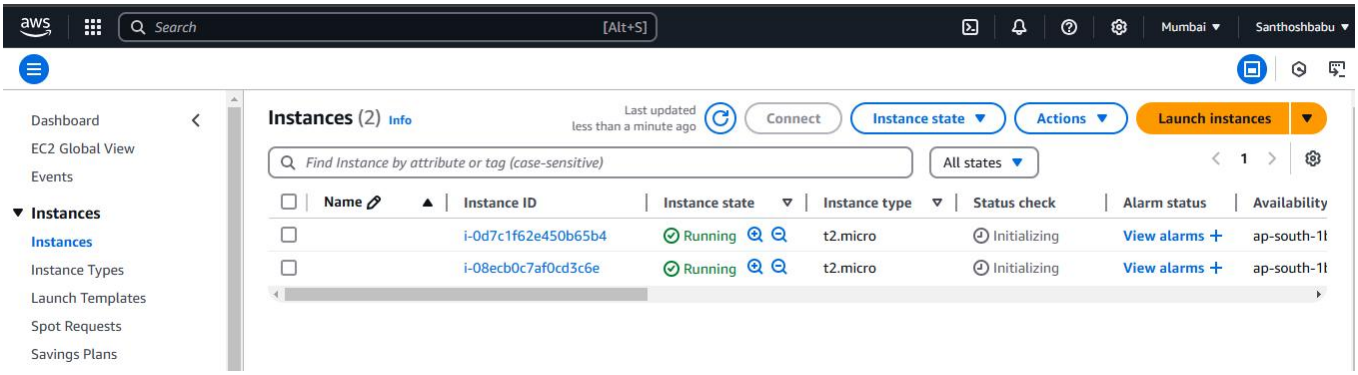
One thing I didn't expect in this project was the complexity of managing fine grained permissions across multiple AWS services. Balancing security and usability, with roles and policies, required careful planning to avoid accidental access issues.

This project took me...

This project took me almost an hour to complete.

Tags

Tags are like labels you can attach to AWS resources for organization. This tagging helps us with identifying all resources with the same tag at once, cost allocation, and applying policies based on environment types.
The tag I've used on my EC2 instances is called env The value I've assigned for my instances are production and development.



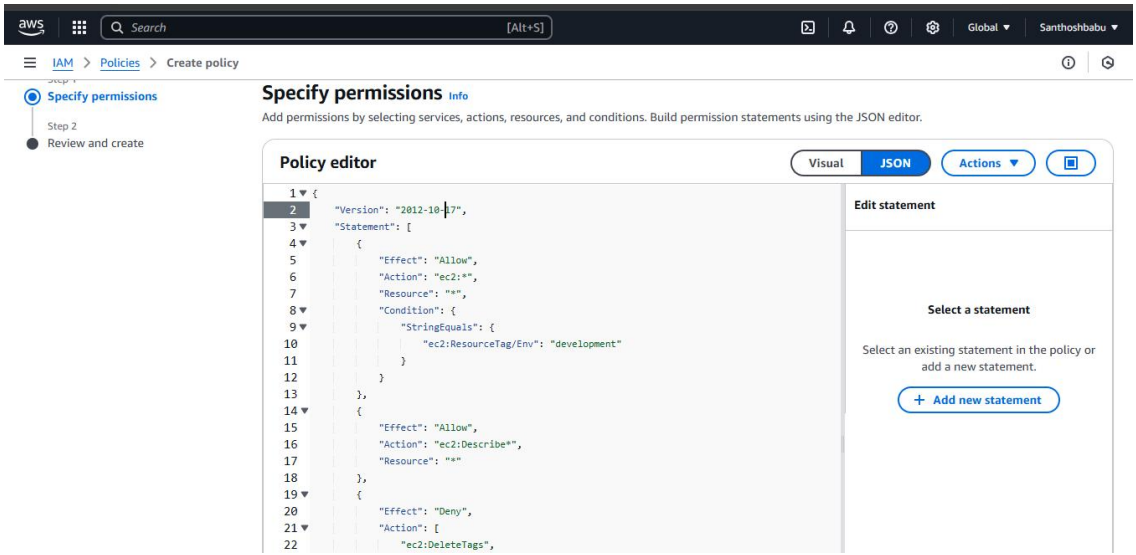
IAM Policies

IAM policies are 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

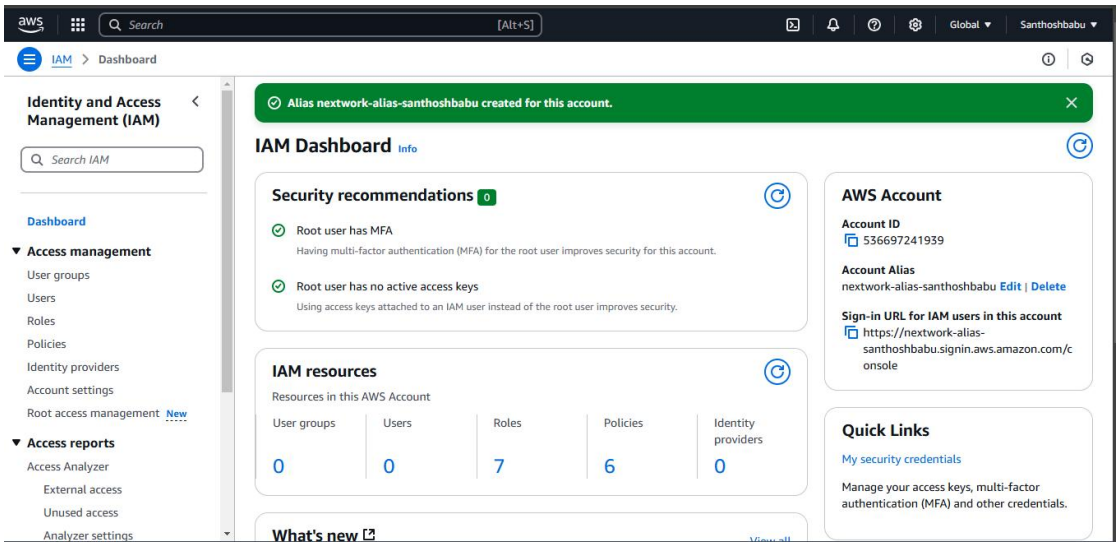
For this project, I've set up a policy using JSON.
I've created a policy that allows some actions (like starting, stopping, and describing EC2 instances) for instances tagged with "Env = development" while denying the ability to create or delete tags for all instances.
When creating a JSON policy, you have to define its Effect, Action and Resource.
The Effect, Action, and Resource attributes of a JSON policy means Effect: Specifies if the action is allowed or denied. Action: Defines what operation can be performed. Resource: Identifies the specific object or service the action applies to.

My JSON Policy



Account Alias

An Account Alias is a friendly name for your AWS account that you can use instead of your account ID (which is usually a bunch of digits) to sign in to the AWS Management Console.
Creating an account alias took me just a minute. Now, my new AWS console sign-in URL is https://nextwork-alias-santhoshbabu.signin.aws.amazon.com/console



IAM Users and User Groups

Users

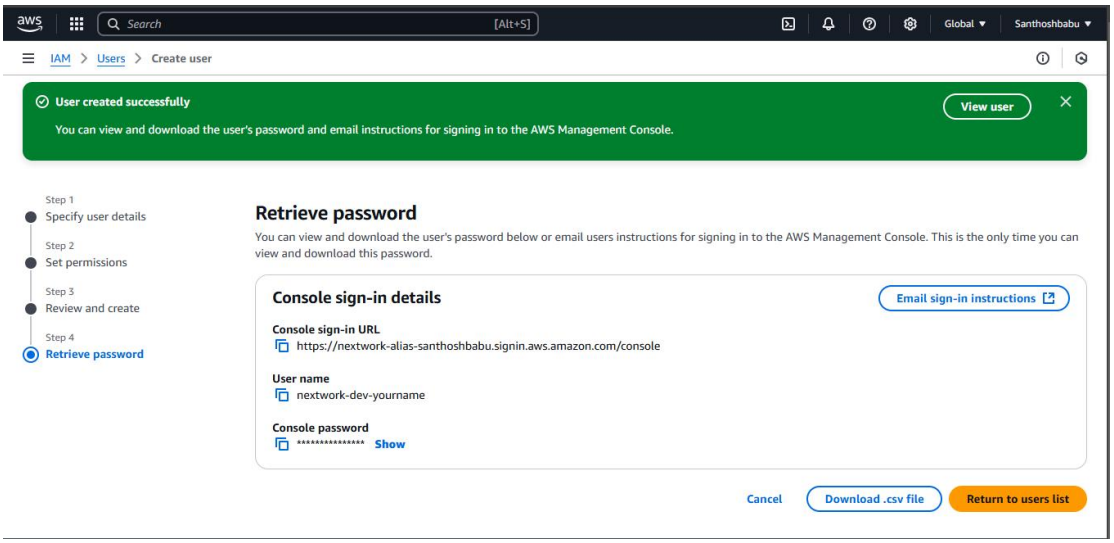
IAM users are the people that will get access to your resources/AWS account, whereas user groups are the collections/folders of users for easier user management.

User Groups

An 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 it grants all users in that group the permissions defined in the policy. Users inherit allowed or denied actions on specified resources, simplifying permission management.

Logging in as an IAM User

The first way is Email: Send the sign-in details (username, temporary password, etc.) securely via email. The second way is Secure Messaging: Share the details through secure communication channels (e.g., encrypted messaging or secure portal).
Once I logged in as my IAM user, I noticed some of your dashboard panels are showing Access denied. This was because your IAM user account does not have the necessary permissions to access those resources or perform actions within AWS environment.



Testing IAM Policies

I tested my JSON IAM policy by stopping both the instances.

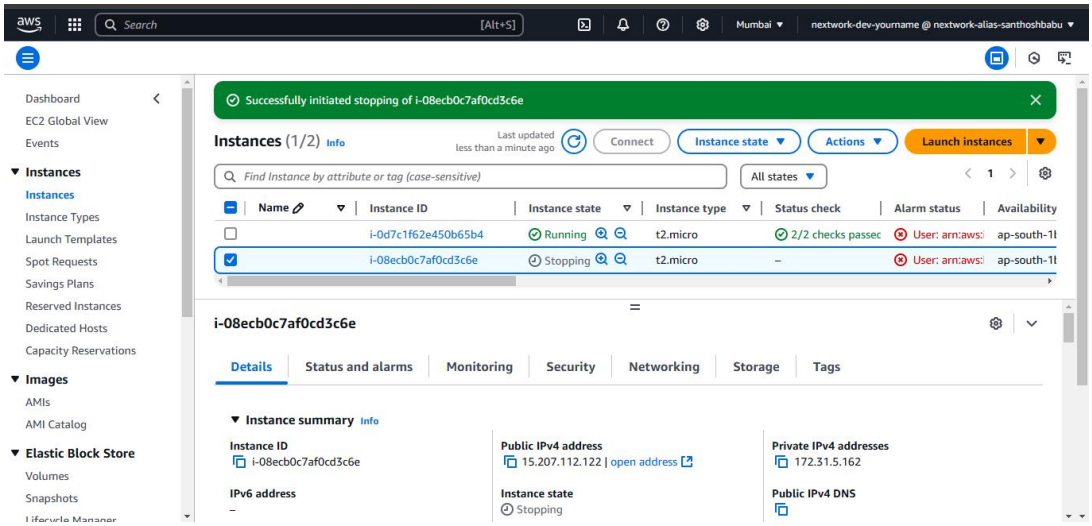
Stopping the production instance

When I tried to stop the production instance failed to stop. This was because we're not authorized! We don't have permission to stop any instance with the production tag.

Testing IAM Policies

Stopping the development instance

Next, when I tried to stop the development instance it stopped. This was because I am authorized to stop any instance with the development tag.



Today you've learned how to:

1. Launch **EC2** instances.
2. Use **tags** for easy identification.
3. Set up **IAM policies** accessing EC2 instances based on their environment (development or production).
4. Create an **IAM user** and assign them to the appropriate user group with the necessary permissions for their role.
5. **Test IAM access** for the User you've created.