St. JOSEPH'S GROUP OF INSTITUTIONS
OMR, CHENNAI - 119

aws

Placement Empowerment Program

Cloud Computing and DevOps Centre

Set Up IAM Roles and Permissions : Create an IAM role on your cloud platform. Assign the role to your VM to restrict/allow specific actions.

Name: Nantha Krishnan.G.K.

Department: AML

St. JOSEPH'S COLLEGE OF ENGINEERING | St. JOSEPH'S INSTITUTE OF TECHNOLOGY

AUTONOMOUS INSTITUTIONS, AFFILIATED TO ANNA UNIVERSITY

Introduction

This Proof of Concept (PoC) demonstrates the process of setting up and utilizing IAM roles and permissions in AWS. The goal is to show how to sec ure AWS resources by managing access through roles rather than hardcoding credentials. Specifically, this PoC focuses on creating an IAM role, assigning it to an EC2 instance, and verifying the instance's access to AWS services such as Amazon S3.

Overview

The process is divided into several key steps:

1. Create an IAM Role: Define a role in AWS IAM and attach policies that grant permissions for specific AWS services.

2. Launch an EC2 Instance: Create a virtual machine (VM) in AWS and configure it for testing the assigned IAM role.

3. Assign the IAM Role to the EC2 Instance: Attach the created IAM role to the EC2 instance to enable access to AWS services without using access keys.

4. Verify Access: Test the EC2 instance to confirm that it has the appropriate permissions by interacting with services like Amazon S3.

Objectives

This PoC aims to achieve the following objectives:

1. Secure Access: Implement IAM roles to grant temporary permissions to AWS resources without embedding credentials.

2. Demonstrate Role-Based Permissions: Show how roles can restrict or allow actions based on attached policies.

3. Test Least Privilege Principle: Ensure that the EC2 instance only has the permissions it needs to perform specific tasks.

4. Hands-On Learning: Provide practical experience with IAM roles and their applications in a cloud environment.

Importance

IAM roles and permissions are fundamental to securing cloud environments. They allow for fine-grained access control and improve operational efficiency by:

1. Eliminating Hardcoded Credentials: Reducing security risks by avoiding the storage of access keys in applications or instances.

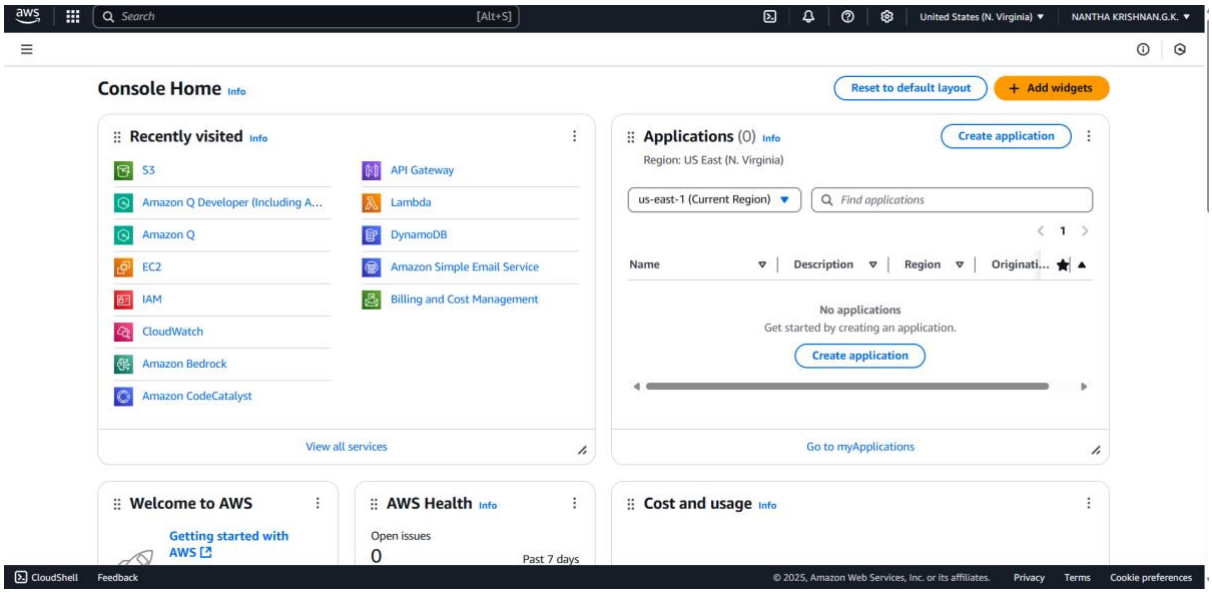
2. Granting Least Privilege Access: Ensuring users and resources only have the permissions they require, minimizing potential misuse.

3. Improving Compliance: Enforcing organizational policies and audit requirements.

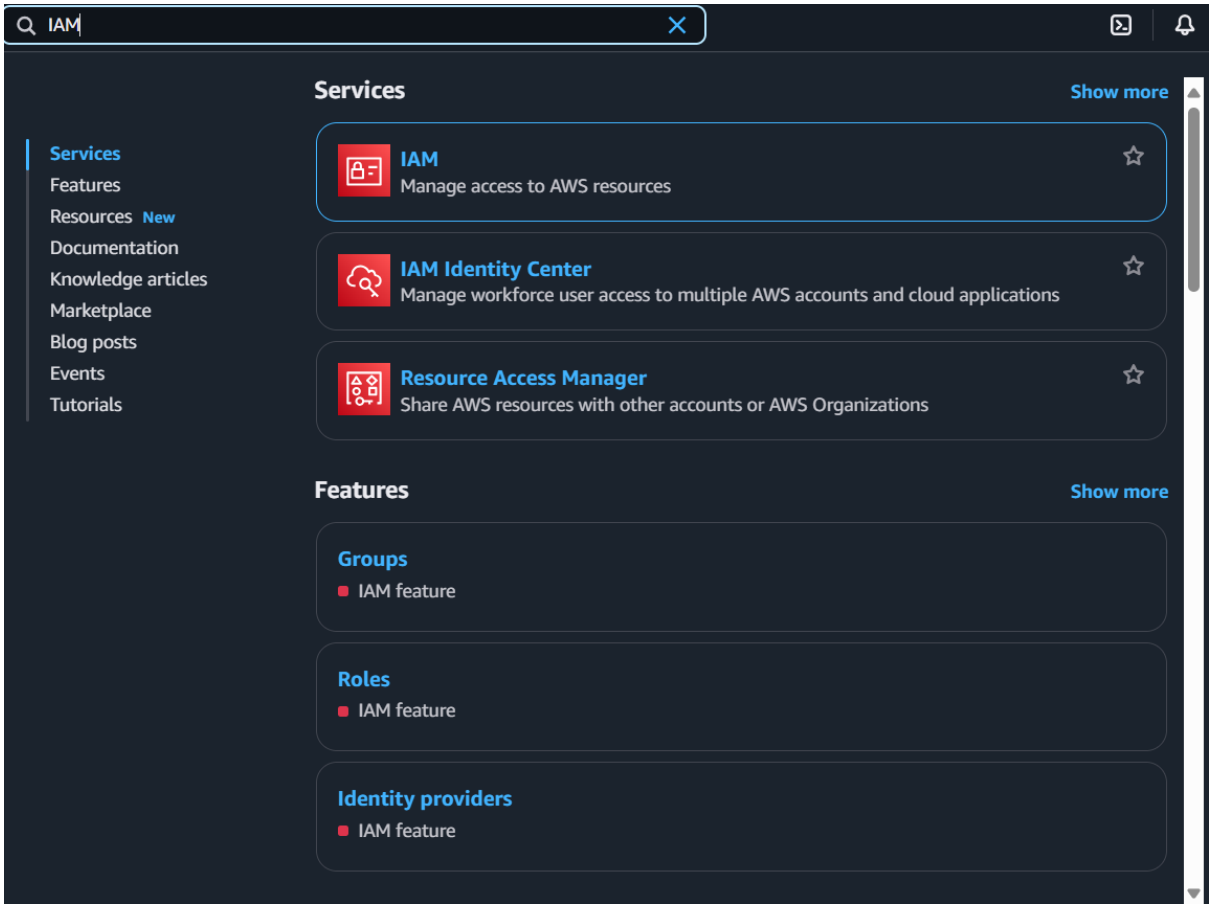
4. Enhancing Automation: Allowing resources like EC2 instances to securely interact with other AWS services.

Step-by-Step Overview

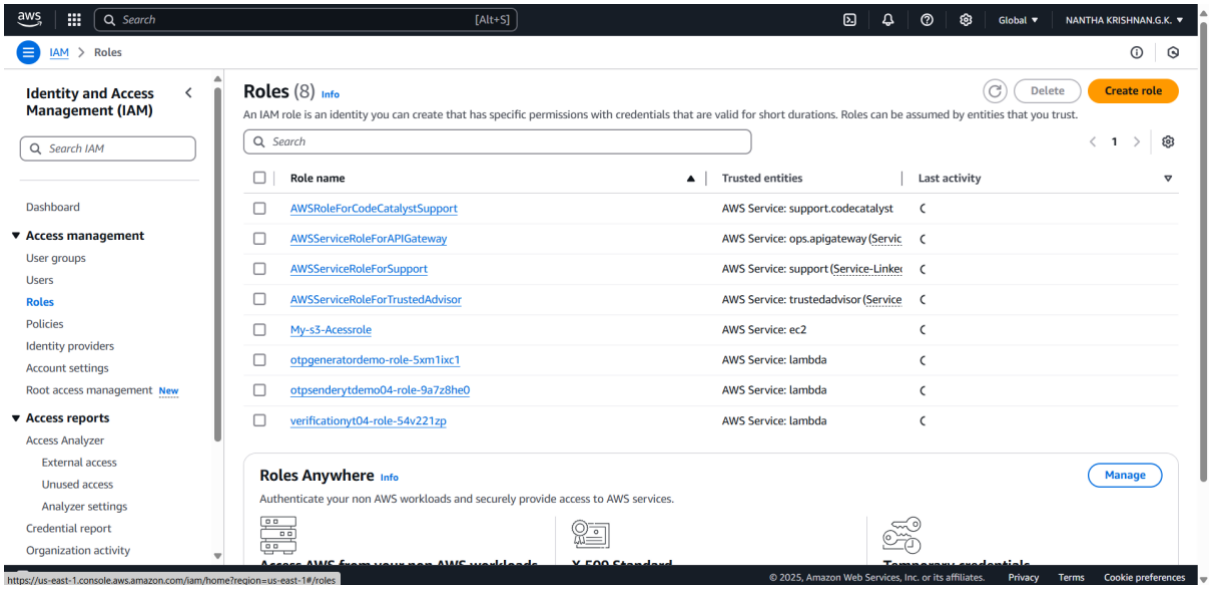
Step 1: Go to the AWS Management Console and enter your username and password to log in.



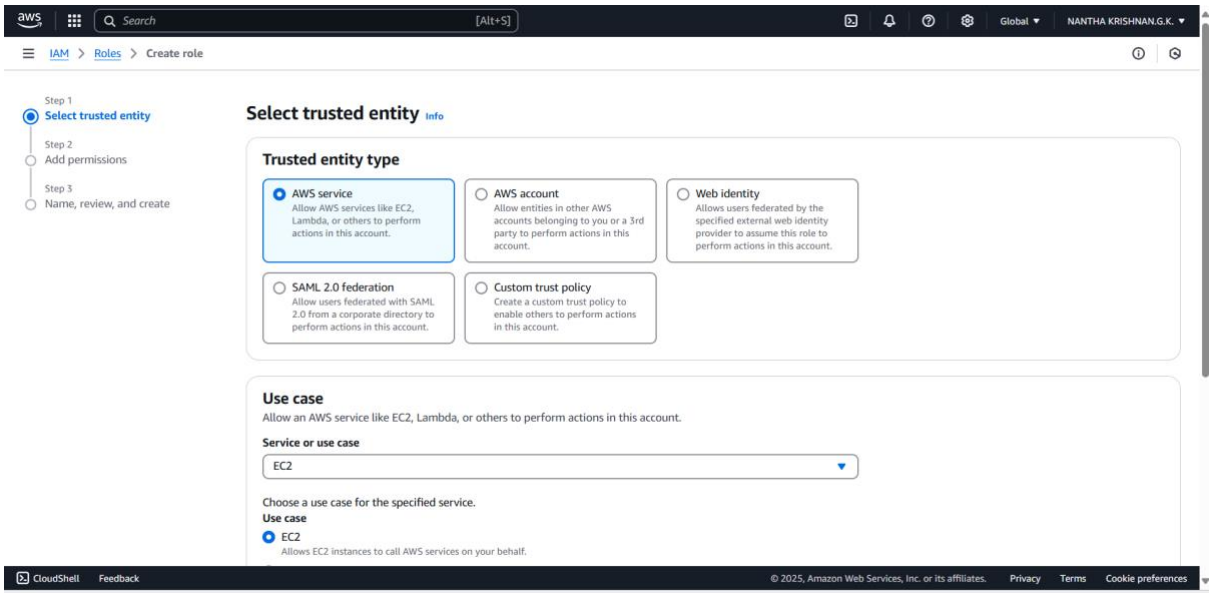
Step 2: In the AWS Management Console, type "IAM" in the search bar at the top. Click on IAM from the search results.



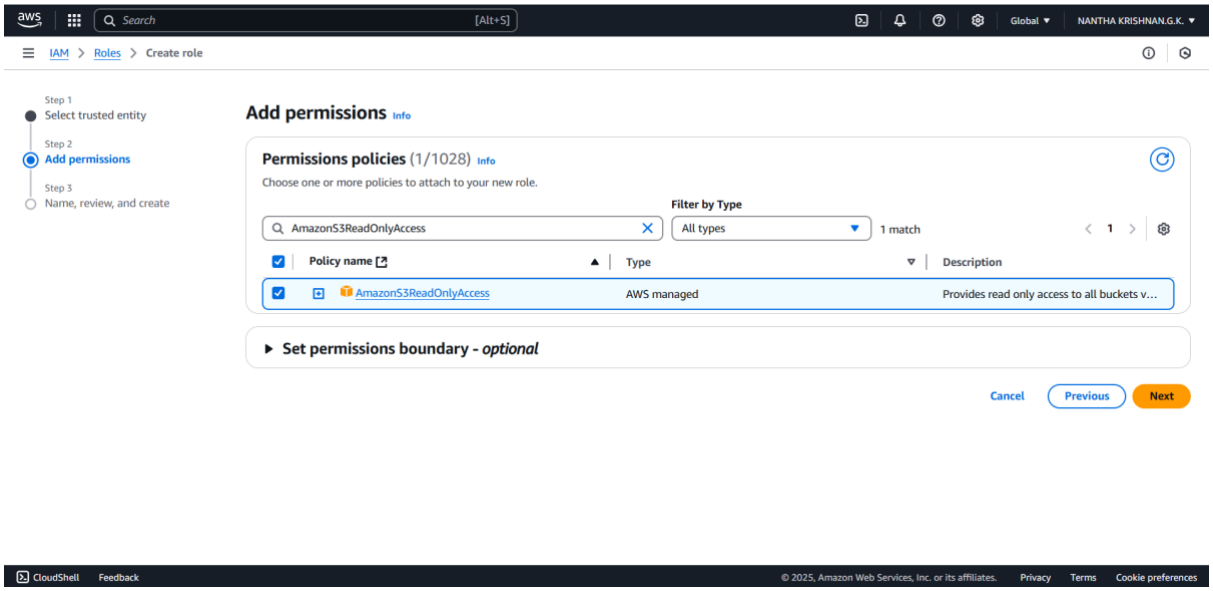
Step 3: On the IAM dashboard, click on "Roles" in the left-hand menu. On the Roles page, click the "Create Role" button.



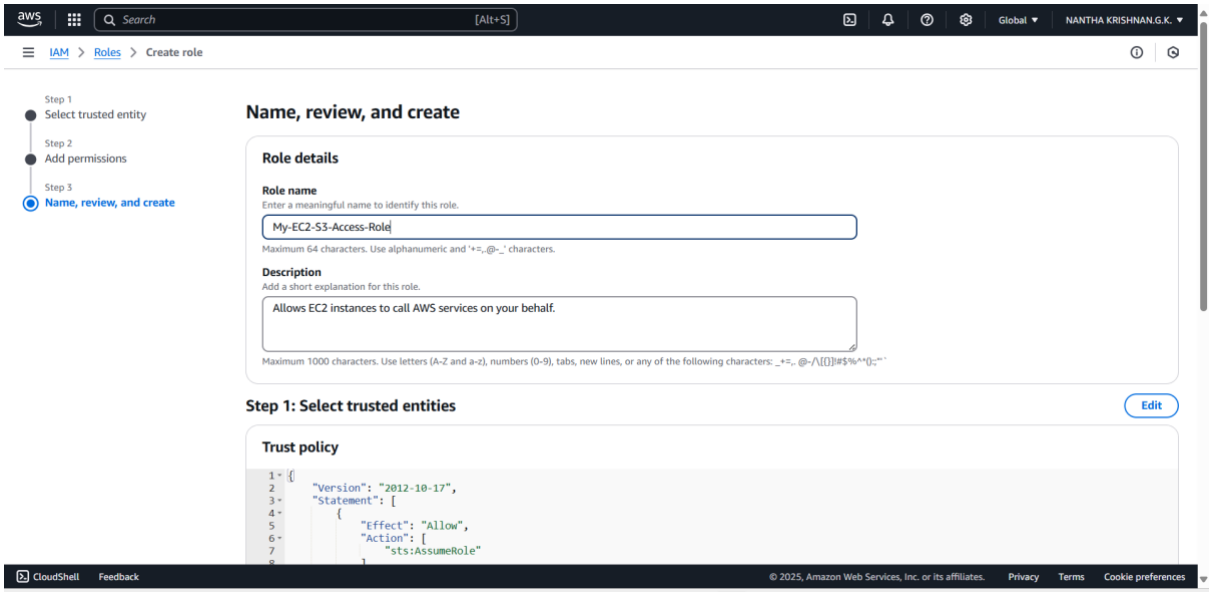
Step 4: On the "Create Role" page, under Trusted Entity Type, select AWS Service (it should be selected by default). In the Use Case dropdown, choose EC2. Click Next to continue.



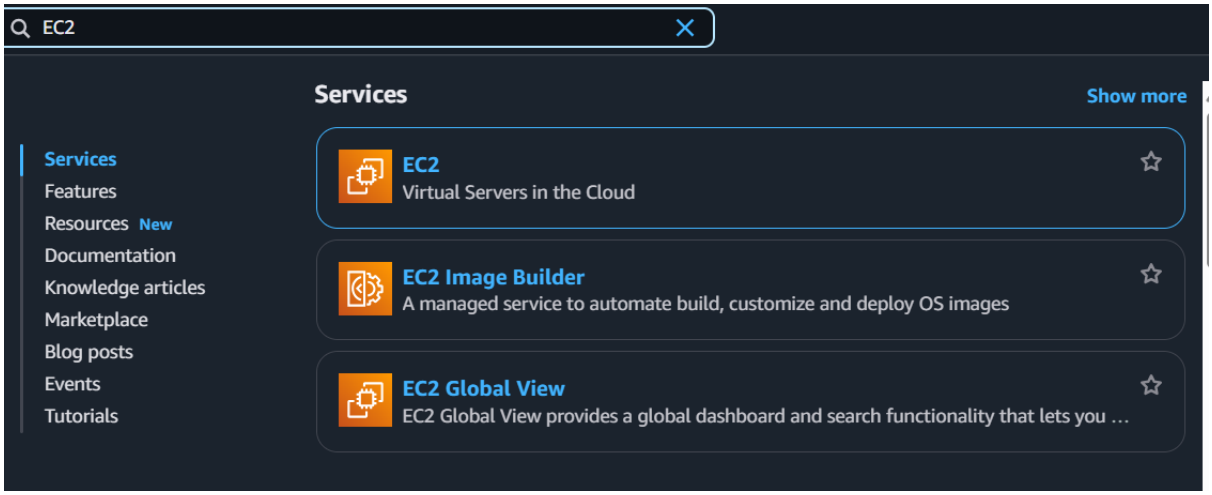
Step 5: On the Permissions page, you'll see a list of policies. Select a policy based on the actions you want the VM to perform. For example, to give the VM read-only access to S3, select AmazonS3ReadOnlyAccess. You can search for policies in the search bar (e.g., type "S3" for S3 policies). Once you've selected a policy, click Next.



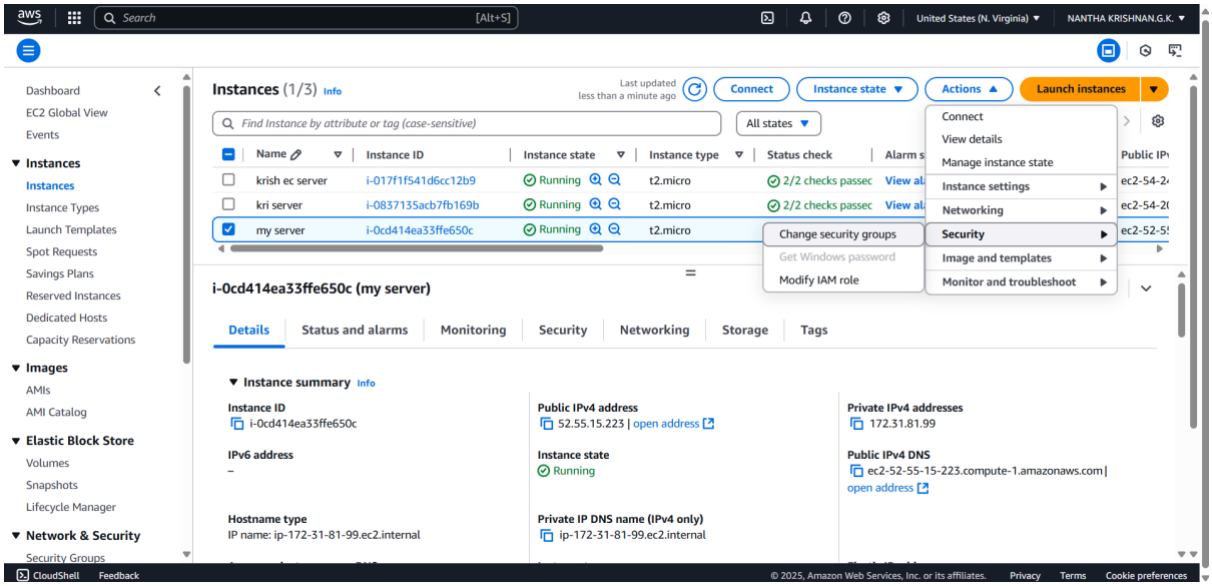
Step 6: On the Role Details page, enter a name for your role (e.g., My-EC2-S3-Access-Role).Add a description or tags if you'd like. Click Create Role to finish.



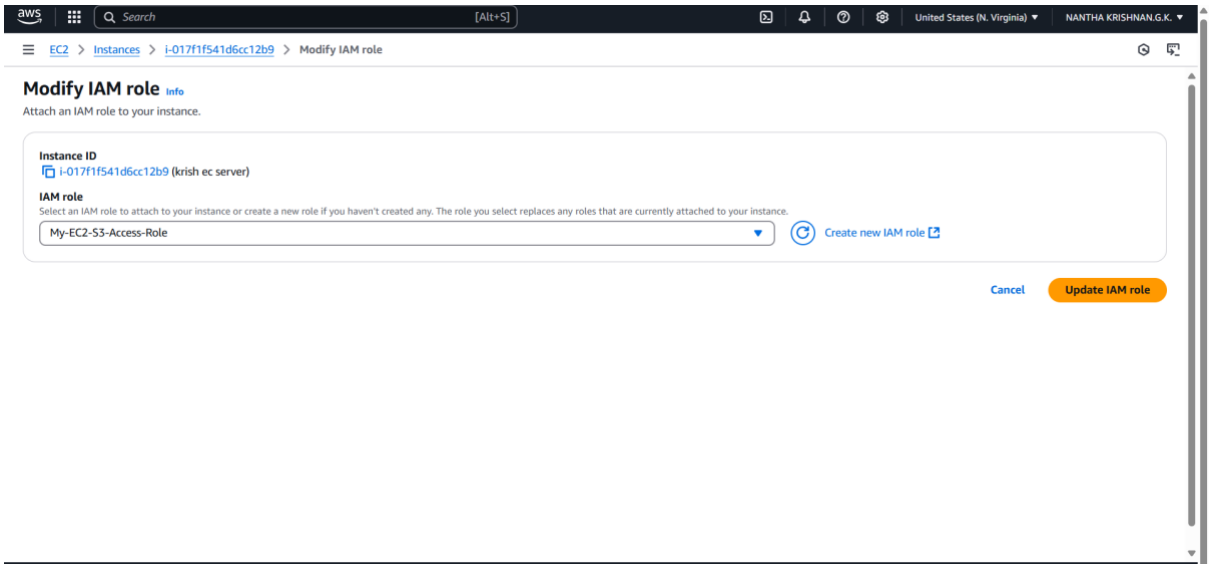
Step 7: In the AWS Management Console, search for EC2 and click to open the EC2 Dashboard. Select the instance (VM) you want to assign the IAM role to.



Step 8: In the Instance details section, click Actions in the top right corner. From the dropdown, choose Security > Modify IAM Role.

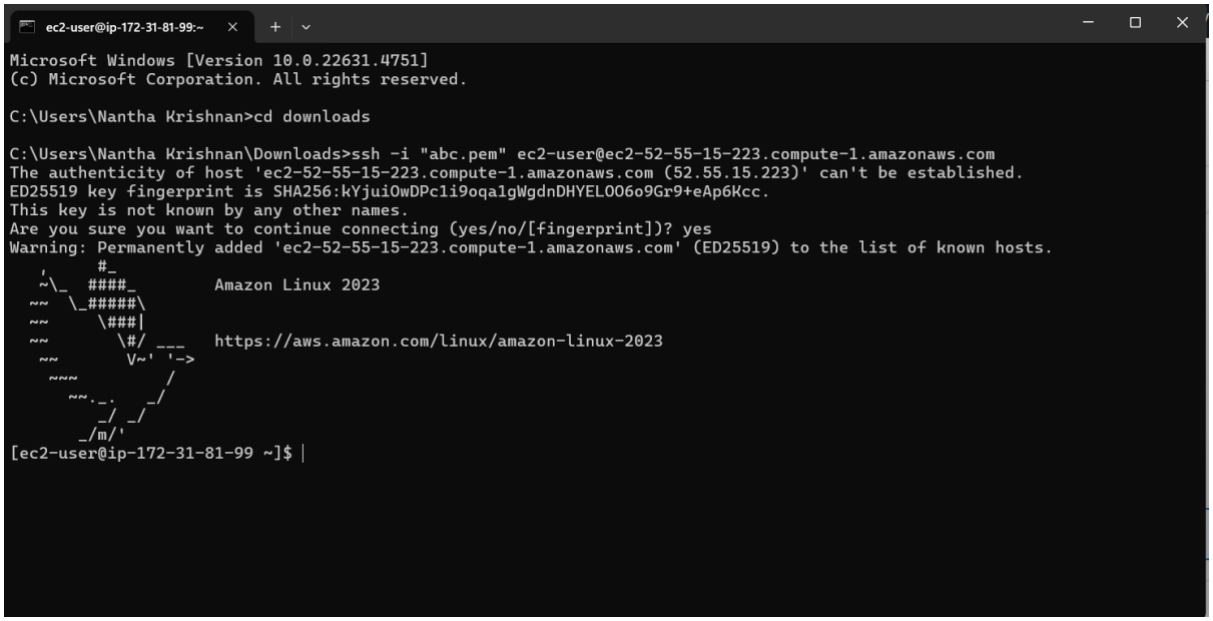


Step 9: In the Modify IAM role window, you should see a dropdown for IAM role. Select the role you created earlier (e.g., My-EC2-S3-Access-Role). Click Update IAM role to apply the changes.



Step 10: Open your terminal (if you're using Linux or macOS) or Command Prompt (Windows). Use SSH to log in to your EC2 instance. For example:ssh -i "your-key-pair.pem" ec2-user@your-ec2-public-ip

Step 11: [ec2-user@ip-172-31-80-54 ~]$ aws ec2 describe-regions --query "Regions[*].RegionName" The error confirms that your IAM role (My-EC2-S3-Access-Role) does not have permissions to perform the ec2:DescribeRegions action. The role currently only has S3-related permissions (e.g., AmazonS3ReadOnlyAccess) and doesn't include broader EC2 permissions.



Outcome

By completing this PoC of setting up IAM roles and permissions with an EC2 instance, you will:

1. Create an IAM role and attach policies to control access to specific AWS services.

2. Launch and configure an EC2 instance for testing purposes.

3. Assign the IAM role to the EC2 instance securely without using access keys.

4. Verify permissions by interacting with AWS services (e.g., listing S3 buckets) from the EC2 instance.

5. Demonstrate the principle of least privilege by ensuring only necessary permissions are granted.