 

**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

Set up IAM Roles and permission : Create a IAM Role on your cloud platform . Assign the role to your VM to restrict / allow specific action.

Name: Mohammed Aneeq M Department: CSE



**Introduction:**

In cloud computing, managing access and permissions is crucial to ensure security, proper resource usage, and operational efficiency. Amazon Web Services (AWS) provides a powerful mechanism for access control through **Identity and Access Management (IAM)**. IAM enables you to securely control access to AWS services and resources. With IAM roles, you can define a set of permissions that grant access to various AWS resources, such as S3 buckets, EC2 instances, and more.

IAM roles can be assigned to AWS resources like EC2 instances, enabling these resources to interact with other services (such as accessing data stored in an S3 bucket) without needing to embed AWS credentials in the application running on those resources.

In this tutorial, we will explore how to create an IAM role in AWS, assign specific permissions to that role, and then attach it to an EC2 instance (VM). We will test the permissions to verify the role's effect on actions allowed and denied based on the IAM policy attached to the role.

**Objective:**

The objective of this exercise is to:

1. Understand the basics of IAM roles and permissions in AWS.
2. Create an IAM role with specific permissions (e.g., allowing S3 access).
3. Attach the IAM role to an EC2 instance.
4. Verify the permissions by testing allowed and denied actions on the EC2 instance.

By the end of this tutorial, you will be able to configure IAM roles and permissions effectively for managing access to AWS resources.

**Prerequisites:**

* An AWS account and basic familiarity with AWS services.
* An existing EC2 instance that you want to assign an IAM role to.
* Basic understanding of IAM concepts like policies, roles, and permissions.

**Steps to Set Up IAM Roles and Permissions in AWS:**

**1. Creating an IAM Role:**

1. **Sign in to the AWS Management Console** and navigate to the **IAM Dashboard**.
2. In the **left sidebar**, click on **Roles**, then click the **Create Role** button.
3. Select **AWS service** as the trusted entity type (since you’ll be assigning the role to an EC2 instance).
4. Choose **EC2** as the service that will use this role.
5. Attach permissions to the role:
   * For this example, we will grant the role **AmazonS3ReadOnlyAccess** permission. This will allow the EC2 instance to read objects from any S3 bucket.
   * Search for **AmazonS3ReadOnlyAccess** in the permissions list and check it to attach the policy.
6. Click **Next: Tags** (you can skip adding tags if not needed).
7. On the **Review** page, give your role a meaningful name (e.g., EC2-S3-ReadOnly-Role).
8. Click **Create Role** to create the IAM role.

**2. Attaching the IAM Role to an EC2 Instance:**

1. **Navigate to the EC2 Dashboard** and select **Instances**.
2. Choose the EC2 instance you want to assign the IAM role to.
3. In the instance details, scroll down to the **IAM role** section and click on **Actions**.
4. Under **Security**, select **Modify IAM Role**.
5. In the **IAM role** dropdown, select the role you created earlier (e.g., EC2-S3-ReadOnly-Role).
6. Click **Update IAM role** to attach the role to the instance.

**3. Verifying IAM Role Permissions:**

To test whether the IAM role works as expected, you can log in to your EC2 instance and attempt actions based on the permissions granted by the IAM role.

1. **SSH into the EC2 Instance** using the private key and public IP (similar to how you connected earlier):

ssh -i "your-key.pem" ubuntu@your-ec2-public-ip

1. **Test Allowed Actions (Accessing S3):**
   * Install the AWS CLI on your EC2 instance if it’s not already installed:

sudo apt-get install awscli

* + Test if the EC2 instance has permission to access S3 (since we attached the AmazonS3ReadOnlyAccess policy):

aws s3 ls

This command lists all the S3 buckets that the instance has permission to access. If the IAM role is correctly set up, it should list the S3 buckets.

1. **Test Denied Actions (Trying to Write to S3):**
   * Attempt to upload a file to an S3 bucket:

aws s3 cp testfile.txt s3://your-bucket-name/

Since the role only has **read-only** access, this action should fail, and you should receive an **Access Denied** error.

**4. Troubleshooting and Adjustments:**

* If the EC2 instance can perform actions it shouldn't, check that the permissions and policies are set up correctly in the IAM role.
* If the EC2 instance cannot perform an allowed action, ensure that the role is properly attached and the policies are correctly configured.

**Best Practices for IAM Roles and Permissions:**

1. **Principle of Least Privilege:** Always grant the minimum permissions required for an entity to perform its tasks. In this example, we granted **read-only** access to S3, which is a good practice unless write permissions are absolutely needed.
2. **Use Managed Policies:** AWS provides pre-built IAM policies like AmazonS3ReadOnlyAccess for common use cases. You can create custom policies as well if needed.
3. **Review Policies Regularly:** Periodically review IAM roles and policies to ensure they align with your current security requirements.

**Conclusion:**

In this tutorial, you have learned how to create an IAM role in AWS, attach permissions to that role, and assign it to an EC2 instance. You also verified the role's permissions by testing actions allowed and denied on the EC2 instance. IAM roles are a key component of securing and managing cloud resources, and mastering IAM policies and permissions is essential for efficient and secure cloud infrastructure management.