



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



Placement Empowerment Program

Cloud Computing and DevOps Centre

Implement Role-Based Access Control in the Cloud:
Create different IAM roles for accessing cloud resources
(e.g., read-only, admin). Test their permissions.

Name: SUBASHINI P

Department: IT



Introduction

In modern cloud environments, **security and access control** are crucial for managing resources effectively. **Role-Based Access Control (RBAC)** in AWS Identity and Access Management (IAM) ensures that users, applications, and services **only have the permissions they need**, reducing security risks.

This PoC demonstrates how to **create, assign, and test IAM roles** with different permissions for AWS resources. We will implement **least privilege access** by assigning:

- **Read-only access to S3** for a user.
- **Full access to EC2** for another user.

Overview

This PoC focuses on **configuring IAM roles with specific permissions** and validating their effectiveness. The key steps include:

1. **Creating IAM Roles**

S3ReadOnlyRole (Grants read-only access to S3).
EC2FullAccessRole (Grants full control over EC2).

2. **Assigning IAM Roles to Users**

Attach S3ReadOnlyRole to User A. Attach EC2FullAccessRole to User B.

3. Testing Permissions

Validate that User A can only list S3 buckets but cannot create/delete them. Verify that User B can launch and manage EC2 instances but cannot access S3.

Objectives

1. Implement **IAM roles with least privilege access**.
2. Demonstrate **secure access control** using AWS IAM.
3. Ensure **users can only perform authorized actions**.
4. Improve **security posture** by restricting unnecessary permissions.

Importance

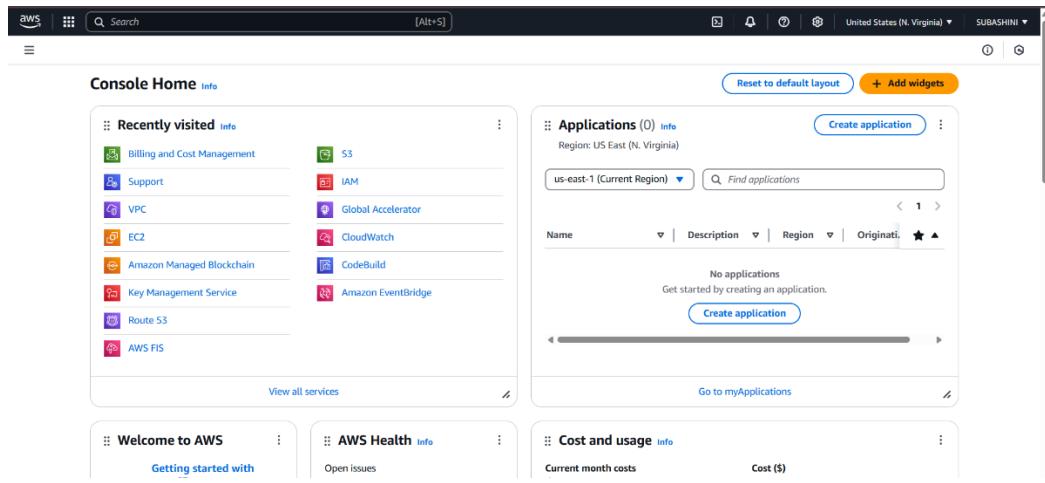
1. **Enhances Cloud Security** – Prevents unauthorized access and enforces least privilege.
2. **Simplifies Permission Management** – IAM roles reduce manual policy management.
3. **Ensures Compliance** – Helps meet security and governance requirements.
4. **Prevents Costly Mistakes** – Avoids accidental resource modifications/deletions.

5. Encourages Best Practices – Follows AWS security guidelines for IAM.

Step-by-Step Overview

Step 1:

1. Go to [AWS Management Console](#).
2. Enter your username and password to log in.



Step 2:

1. Sign in to AWS Management Console.
2. Go to IAM → Roles → Create Role.

The screenshot shows the AWS Identity and Access Management (IAM) service interface. On the left, there's a navigation sidebar with options like 'Identity and Access Management (IAM)', 'Dashboard', 'Access management' (with 'Roles' selected), 'Policies', 'Identity providers', 'Account settings', and 'Root access management'. Below that is 'Access reports' with 'Access Analyzer', 'External access', 'Unused access', 'Analyzer settings', 'Credential report', and 'Authorization artifacts'. The main content area is titled 'Roles (14) Info' and contains a table with 14 rows, each representing a role with its name, trusted entity, and last activity.

Role name	Trusted entities	Last activity
Amazon_EventBridge_Invoke_Event_Bus_1093280111	AWS Service: events	-
AWSDataSyncS3BucketAccess-aws-easy-website25-af5bb	AWS Service: datasync	-
AWSDataSyncS3BucketAccess-tom25-92577	AWS Service: datasync	-
AWSServiceRoleForAmazonElasticFileSystem	AWS Service: elasticfilesystem (Service-Linked)	41 days ago
AWSServiceRoleForAmazonManagedBlockchain	AWS Service: managedblockchain (Service-Linked)	-
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Linked)	21 days ago
AWSServiceRoleForBackup	AWS Service: backup (Service-Linked)	4 hours ago
AWSServiceRoleForElasticLoadBalancing	AWS Service: elasticloadbalancing (Service-Linked)	26 days ago
AWSServiceRoleForGlobalAccelerator	AWS Service: globalaccelerator (Service-Linked)	-
AWSServiceRoleForServiceQuotas	AWS Service: servicequotas (Service-Linked)	-
AWSServiceRoleForSupport	AWS Service: support (Service-Linked)	17 days ago
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked)	-

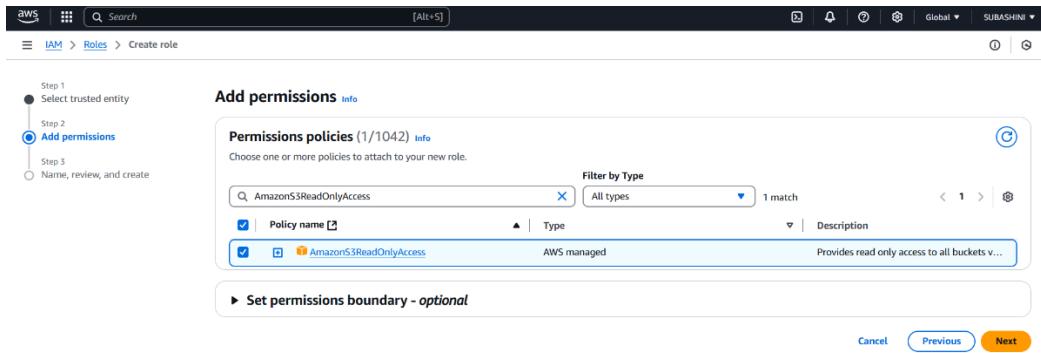
Step 3:

1. **Select trusted entity:** Choose **AWS Service**.
2. **Use case:** Select **EC2** role for an instance.
3. Click **Next**.

This screenshot shows the 'Create role' wizard at 'Step 1: Select trusted entity'. It has three tabs: 'Step 1: Select trusted entity' (selected), 'Step 2: Add permissions', and 'Step 3: Name, review, and create'. Under 'Trusted entity type', the 'AWS service' option is selected, which is described as allowing users from other AWS accounts to perform actions in this account. Other options include 'AWS account' (allowing entities in other AWS accounts to perform actions), 'Web identity' (allowing users federated by a provider to assume this role), 'SAML 2.0 federation' (allowing users federated with SAML 2.0 from a corporate directory to perform actions), and 'Custom trust policy' (creating a custom trust policy). Below this, the 'Use case' section allows selecting a service or use case. The 'Service or use case' dropdown is set to 'EC2', and the 'Use case' dropdown is also set to 'EC2' with the note 'Allows EC2 instances to call AWS services on your behalf'.

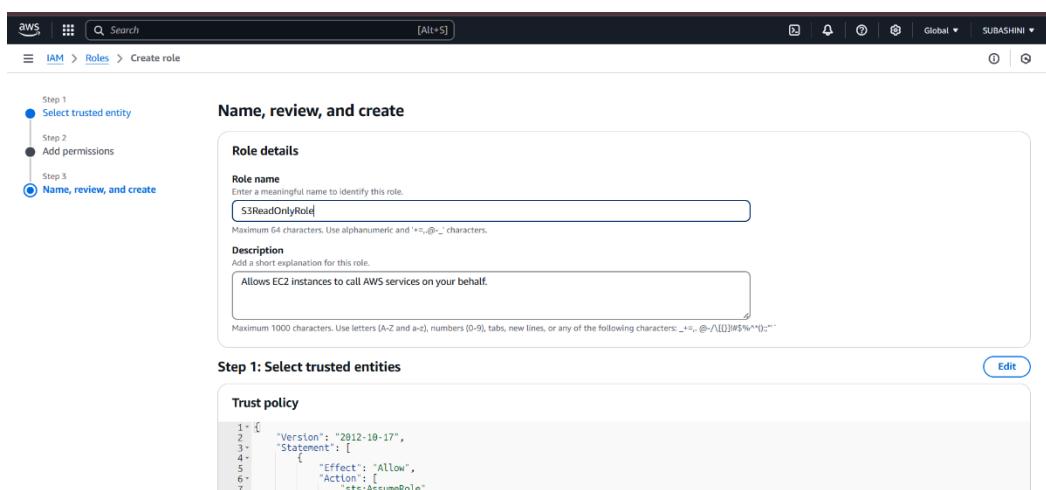
Step 4:

Search for **AmazonS3ReadOnlyAccess** and select it.



Step 5:

1. Click **Next** → Name the role **S3ReadOnlyRole**.
2. Click **Create Role**.



Step 6

1. Go to **IAM** → **Roles** → **Create Role**.
2. **Select trusted entity:** Choose **AWS Service**.

3. Use case: Select EC2.

4. Click Next.

5. Attach permissions:

Search for **AmazonEC2FullAccess** and select it.

6. Click Next → Name the role **EC2FullAccessRole**.

7. Click **Create Role**.

The screenshot shows the 'Add permissions' step of the IAM role creation wizard. On the left, a vertical navigation bar indicates 'Step 1: Select trusted entity', 'Step 2: Add permissions' (which is selected), and 'Step 3: Name, review, and create'. The main area is titled 'Add permissions' with a 'Permissions policies (1/1042)' link. A search bar shows 'AmazonEC2FullAccess'. A table lists one policy: 'AmazonEC2FullAccess' (AWS managed, Type: AWS managed, Description: Provides full access to Amazon EC2 via the AWS Management Console). Below the table is a section for 'Set permissions boundary - optional'. At the bottom right are 'Cancel', 'Previous', and 'Next' buttons, with 'Next' highlighted in orange.

The screenshot shows the 'Name, review, and create' step of the IAM role creation wizard. The vertical navigation bar on the left shows 'Step 1: Select trusted entity', 'Step 2: Add permissions', and 'Step 3: Name, review, and create' (selected). The main area is titled 'Name, review, and create' with a 'Role details' section. It includes fields for 'Role name' (set to 'EC2fullAccessRole') and 'Description' (set to 'Allows EC2 instances to call AWS services on your behalf'). Below this is the 'Step 1: Select trusted entities' section, which contains a 'Trust policy' editor. The trust policy JSON is displayed as follows:

```
1+ [
2+   "Version": "2012-10-17",
3+   "Statement": [
4+     {
5+       "Effect": "Allow",
6+       "Action": [
7+         "sts:AssumeRole"
7+       ]
7+     }
7+   ]
7+ ]
```

Step 7

1. Go to **IAM** → **Users**.

2. Select a user.

The screenshot shows the AWS IAM 'User Details' page for 'iam_user'. The 'Permissions' tab is active, displaying two attached policies: 'AmazonS3ReadOnlyAccess' and 'AmazonEC2ReadOnlyAccess'. Both policies are listed as AWS managed and attached directly. The 'Add permissions' button is visible at the top right of the policy list.

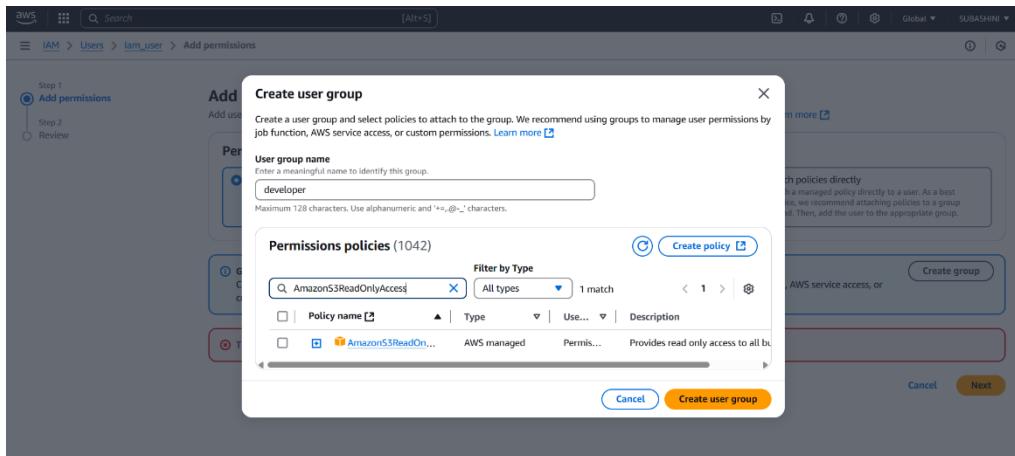
Step 8

1. Assign:

- **S3ReadOnlyRole** to one user.
- **EC2FullAccessRole** to another user.

2. Click **Next** → **Review** → **Add permissions**.

The screenshot shows the AWS IAM 'User Details' page for 'iam_user'. The 'Permissions' tab is active, displaying two attached policies: 'AmazonS3ReadOnlyAccess' and 'AmazonEC2ReadOnlyAccess'. Both policies are listed as AWS managed and attached directly. The 'Add permissions' button is visible at the top right of the policy list.



Summary

User group name: developer

Creation time: March 09, 2025, 18:39 (UTC+05:30)

ARN: arn:aws:iam::253490765722:group/developer

Users (1) **Permissions** **Access Advisor**

Users in this group (1)

User name	Groups	Last activity	Creation time
iam_user	None	31 days ago	

Step 8

1. Go to **EC2** → **Select an Instance**.
2. Click **Actions** → **Security** → **Modify IAM Role**.
3. Attach **EC2FullAccessRole** to the instance.
4. Click **Update IAM Role**.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with sections for EC2, Dashboard, EC2 Global View, Events, Instances (with sub-options like Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots, Lifecycle Manager). The main content area shows a table titled 'Instances (1/1) Info' with one row for 'My instance' (i-0b3b2defa89cf7108). The instance is listed as 'Running' with an 't2.micro' instance type and an 'Initializing' status check. The 'Actions' dropdown menu is open, showing options like 'Connect', 'View details', 'Manage instance state', and 'Public IP'. Below the table, the specific instance details are expanded, showing fields for Instance ID (i-0b3b2defa89cf7108), Public IPv4 address (172.31.231.151), Instance state (Running), Private IPv4 addresses (172.31.231.151), and Public IPv4 DNS.

Step 9

Open Command prompt

1. Run:

```
aws s3 ls
```

It should list S3 buckets.

2. Try creating a bucket: aws s3 mb s3://test-

bucket It should **deny access**.

```
C:\Users\subam>aws s3 ls
2025-02-07 12:49:11 strbucket-1
2025-02-07 12:49:54 strbucket-2

C:\Users\subam>aws s3 mb s3://strbucket-1
make_bucket failed: s3://strbucket-1 An error occurred (AccessDenied) when calling the CreateBucket operation: User: arn:aws:iam::253490765722:user/iam_user is not authorized to perform: s3:CreateBucket on resource: "arn:aws:s3:::strbucket-1" because no identity-based policy allows the s3:CreateBucket action
```

Step 10

1. Sign in as the user with **EC2FullAccessRole**.
2. Try launching an EC2 instance:

aws ec2 run-instances --image-id ami-12345678 --instance-type t2.micro –subnet-id subnet-1234567

3. It should succeed.
4. Try listing S3 buckets:

aws s3 ls

5. It should deny access.

```
C:\Users\subam>aws ec2 run-instances --image-id ami-08b5b3a93ed654d19 --instance-type t2.micro --subnet-id subnet-004337
654f3dd6d99

An error occurred (UnauthorizedOperation) when calling the RunInstances operation: You are not authorized to perform this operation. User: arn:aws:iam::253490765722:user/iam_user is not authorized to perform: ec2:RunInstances on resource: a resource: arn:aws:ec2:us-east-1:253490765722:instance/* because no identity-based policy allows the ec2:RunInstances action. Encoded authorization failure message: dDs03W_AtUdBtPyPwcj1GgqxL8TVmGuNlCjg3dpkcWZm8r58eyvnFJybzdvuyLFMrRug9xUne5uvYW27ZVkQsN0MxFkgAGre6xC-Q6rn6Tf1uLI8PQgTR_gMSL01xpZrintIKViCZ8G9eaYIlg9FGwhxeAyyKyc35aaNm0TlPnoJ4MyHqi3v7Ca0Ng5X8NU8RXZr-z8FtlL6LV6x4VY-hhttq495y0MIui12gATmbVxEY_n41NxwEintGdWTbVEDXbtDW27vjeu9J7Dmk75DYl516MpBdU4UupEvkWhgIrVbEymus9XkzewevevoUqxprEwxOTNNvdap11-4GeLHWHiKosYm7-amC4eCXWjn9u9-k_JuuN4-jpzsETjHcnspotYW4r60B8crCT_H00jj;VRwZTnIvD-uFRogyVhX15qr1X7PswGvL2jTS3xiBIfCmg9Wq4VnHMrGXw05CJZHUpE6pBhGTkaCEhgV7icQ0A3-rgmyanQfEoJ3LHmyn8sizrjKHLwDH3UKjjLouupICTn7dzBbg8M6Hwv7Vjzis_nZtLSH3306wdj6kfd6Vw0stXIotHAyVwAnbxD0B1Rp4eeLG3zhFqGHHk1jmC3Zk4GuXtG36AiYyQaFoZohngfZ3xp3fPhhKZnen7cwS6uQFi38XgXhDmPfce-rhZos90_0LtVWiGR E30GCEB0aWCDUUUGy0ZX38AUbBXzf_kh_3Hlw5Xb3n1jNf-lHdZQ3QUMU4CpswIdVlsSMFLQPXF7cJW0Zx8tg8mE9
```

Outcomes

By completing this **Role-Based Access Control (RBAC) in AWS IAM PoC**, you will:

1. **Understand AWS IAM Roles & Policies** – Gain hands-on experience in creating and managing IAM roles with different levels of access control.
2. **Implement Least Privilege Access** – Learn how to restrict permissions effectively, ensuring users and services only have the minimum access required.
3. **Assign IAM Roles to Users** – Practice attaching predefined IAM policies (AmazonS3ReadOnlyAccess and AmazonEC2FullAccess) to different users securely.
4. **Test & Validate Permissions** – Verify that IAM users can perform only the allowed actions, ensuring security by testing access to S3 and EC2.
5. **Enhance Cloud Security Best Practices** – Improve AWS security posture by reducing the risk of unauthorized access and preventing accidental resource modifications.
6. **Use AWS CLI for IAM Management** – Execute AWS CLI commands to list, create, and verify permissions assigned through IAM roles efficiently.