

## Assignment: IAM Policies Creation and Attachment

Name: *Vikram*

Course: AWS DevOps

---

### Problem Statement

You work for **XYZ Corporation**.

To maintain the security of the AWS account and resources, you have been asked to implement a solution that helps easily **recognize and monitor different users** by assigning proper **IAM policies** to user groups.

---

### Tasks to Be Performed

1. **Create Policy 1** which allows users to:
    - a. Access **S3 completely**
    - b. **Only create EC2 instances**
    - c. Have **full access to RDS**
  2. **Create Policy 2** which allows users to:
    - a. Access **CloudWatch and Billing completely**
    - b. **Only list EC2 and S3 resources**
  3. **Attach Policy 1** to the **Dev Team** (from the previous task).
  4. **Attach Policy 2** to the **Ops Team** (from the previous task).
- 

### Implementation Steps (Using AWS Management Console)

---

#### Step 1 — Open IAM

1. Sign in to the **AWS Management Console**.
  2. In the top search bar, type **IAM** → open the **IAM service**.
  3. You will see the IAM Dashboard.
-

## Step 2 — Create Policy 1

1. In the left sidebar, click **Policies** → **Create policy**.
2. Select the **JSON** tab and paste the following policy:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "s3:*",  
      "Resource": "*"  
    },  
    {  
      "Effect": "Allow",  
      "Action": "ec2:RunInstances",  
      "Resource": "*"  
    },  
    {  
      "Effect": "Allow",  
      "Action": "rds:*",  
      "Resource": "*"  
    }  
}
```

3. Click **Next**, give the policy a name: Policy-1
4. Add description: *Allows full S3 and RDS access, and EC2 creation only*
5. Click **Create policy**.

*Result:* Policy-1 is created successfully.

---

Step 1  
**Specify permissions**

**Specify permissions** Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Step 2  
Review and create

**Policy editor**

```

1▼ {
2  "Version": "2012-10-17",
3▼ "Statement": [
4▼  {
5    "Effect": "Allow",
6    "Action": "s3:*",
7    "Resource": "*"
8  },
9▼  {
10   "Effect": "Allow",
11   "Action": "ec2:RunInstances",
12   "Resource": "*"
13 },
14▼  {
15   "Effect": "Allow",
16   "Action": "rds:/*",
17   "Resource": "*"
18 }
19 ]
20 }
```

**Visual** | **JSON****Edit statement**

Select an existing statement

**Policy details****Policy name**

Enter a meaningful name to identify this policy.

Policy1

Maximum 128 characters. Use alphanumeric and '+,-,.,@-' characters.

**Description - optional**

Add a short explanation for this policy.

Allows full S3 and RDS access and EC2 creation only

Maximum 1,000 characters. Use alphanumeric and '+,-,.,@-' characters.

**Permissions defined in this policy** Info**Edit**

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it.

 Search Show remaining 446 services**Allow (3 of 449 services)**

Service	Access level	Resource	Request condition
EC2	Limited: Write	All resources	None
RDS	Full access	All resources	None
S3	Full access	All resources	None

**Add tags - optional** ...**Policies (1/1400)** Info**C** Actions ▾ Delete Create policy

A policy is an object in AWS that defines permissions.

**Filter by Type** policy1 X

All types

1 match

&lt; 1 &gt;



Policy name	Type	Used as	Description
Policy1	Customer managed	None	Allows full S3 and RDS access and EC2 ...

## Step 3 — Create Policy 2

1. Go to **IAM → Policies → Create policy**.
2. Choose **JSON** tab and paste the following:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "cloudwatch:*",  
        "aws-portal:*"  
      ],  
      "Resource": "*"  
    },  
    {  
      "Effect": "Allow",  
      "Action": [  
        "ec2:Describe*",  
        "s3>ListAllMyBuckets",  
        "s3>ListBucket"  
      ],  
      "Resource": "*"  
    }  
  ]  
}
```

3. Click **Next**, give the policy a name: Policy-2
4. Add description: *Allows full CloudWatch & Billing access and list permissions for EC2 and S3*
5. Click **Create policy**.

**Result:** Policy-2 is created successfully.

---

Policy1 created.

and create

**Policy editor**

```

1 ▼ {
2   "Version": "2012-10-17",
3 ▼   "Statement": [
4 ▼     {
5       "Effect": "Allow",
6 ▼       "Action": [
7         "cloudwatch:*,",
8         "aws-portal:/*"
9       ],
10      "Resource": "*"
11    },
12    {
13      "Effect": "Allow",
14 ▼      "Action": [
15        "ec2:Describe*",
16        "s3>ListAllMyBuckets",
17        "s3>ListBucket"
18      ],
19      "Resource": "*"
20    }
21  ]
22 }

```

[Visual](#) [JSON](#)[Edit statement](#)[Select](#)Select an existing  
add a new action[+ Add](#)**Create policy**

ated.

**Policy details****Policy name**

Enter a meaningful name to identify this policy.

Maximum 128 characters. Use alphanumeric and '+-=,.@-\_` characters.

**Description - optional**

Add a short explanation for this policy.

Maximum 1,000 characters. Use alphanumeric and '+-=,.@-\_` characters.

[Edit](#)**Permissions defined in this policy** Info

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it.

 Search Show remaining 445 services**Allow (4 of 449 services)**

Service	▲   Access level	▼   Resource	Request condition
Billing Console	Full access	All resources	None
CloudWatch	Full access	All resources	None
EC2	Limited: List	All resources	None
S3	Limited: List	All resources	None

**Policy Policy2 created.**[View policy](#)[X](#)**Policies (1/1401)** Info

A policy is an object in AWS that defines permissions.

[Actions](#) ▾[Delete](#)[Create policy](#)**Filter by Type** policy2

All types

1 match

&lt; 1 &gt;



Policy name	▲   Type	▼   Used as	▼   Description
<input checked="" type="radio"/> <a href="#">Policy2</a>	Customer managed	None	Allows full CloudWatch and Billing acc...

## Step 4 — Attach Policy 1 to Dev Team

1. Go to **IAM → User groups**.
2. Click on **Dev-Team** (created in previous assignment).
3. Open the **Permissions** tab → click **Add permissions** → **Attach policies directly**.
4. Search for Policy-1, select it → click **Attach policy**.

**Result:** Policy-1 attached to Dev-Team.

The screenshot shows the AWS IAM User Groups page for the 'Dev-Team' group. At the top, a green banner states 'Policies attached to this user group.' Below this, the 'Summary' section displays the user group name 'Dev-Team', creation time 'October 17, 2025, 13:05 (UTC+05:30)', and ARN 'arn:aws:iam::062250062838:group/Dev-Team'. The 'Permissions' tab is selected, showing one attached policy: 'Policy1' (Customer managed). Other tabs include 'Users (2)' and 'Access Advisor'. A 'Permissions policies' table lists the attached policy with columns for Policy name, Type, and Attached entities.

## Step 5 — Attach Policy 2 to Ops Team

1. Go to **IAM → User groups** → click **Ops-Team**.
2. Open the **Permissions** tab → **Add permissions** → **Attach policies directly**.
3. Search for Policy-2, select it → click **Attach policy**.

**Result:** Policy-2 attached to Ops-Team.

The screenshot shows the AWS IAM User Groups page for the 'Ops-Team' group. At the top, a green banner states 'Policies attached to this user group.' Below this, the 'Summary' section displays the user group name 'Ops-Team', creation time 'October 17, 2025, 13:06 (UTC+05:30)', and ARN 'arn:aws:iam::062250062838:group/Ops-Team'. The 'Permissions' tab is selected, showing one attached policy: 'Policy2' (Customer managed). Other tabs include 'Users (3)' and 'Access Advisor'. A 'Permissions policies' table lists the attached policy with columns for Policy name, Type, and Attached entities.

## Step 6 — Verification

- Go to **User groups** → **Dev-Team** → **Permissions tab**
  - You should see **Policy-1** attached.
- Go to **User groups** → **Ops-Team** → **Permissions tab**
  - You should see **Policy-2** attached.

*Result:* Both policies are correctly attached to their respective groups.

---

## Final Output

Successfully created and attached IAM policies as per the requirements.

This ensures each team has the correct access level to AWS services, improving **security, resource management, and monitoring efficiency**.