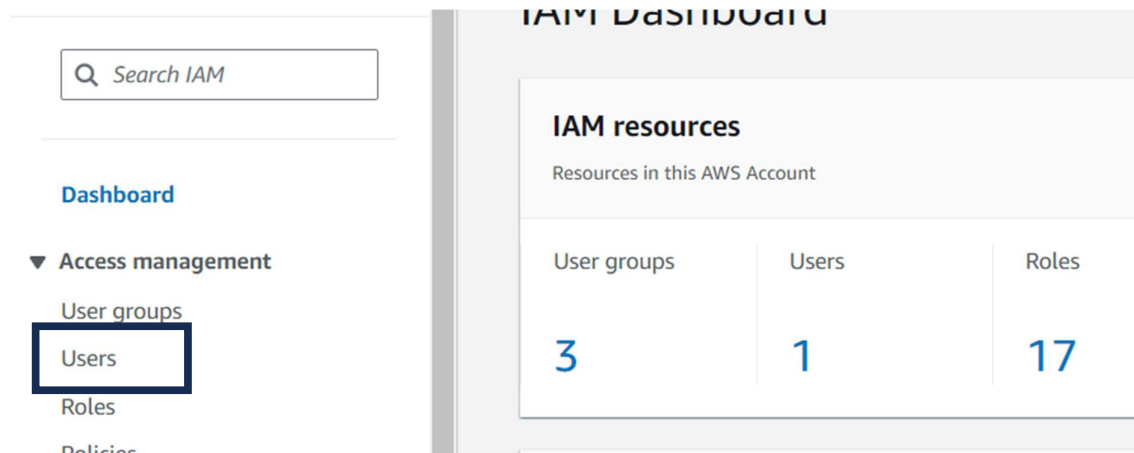


1. Create IAM Users

1. **Login to AWS Management Console.**
2. **Navigate to IAM** (Identity and Access Management).
3. In the left navigation pane, click on **Users**.



4. Click on **Add user**.
5. For each user:
 - Enter the user name (Dev1, Dev2, Test1, Test2).
 - Select **AWS Management Console access**.
 - Set a custom password or let AWS auto-generate it.
 - Uncheck the option for "User must create a new password at next sign-in" if you do not want them to change the password at the first login.

The screenshot shows the 'User details' form in the AWS IAM console. The 'User name' field is filled with 'Dev1'. The 'Provide user access to the AWS Management Console - optional' checkbox is checked. The 'Autogenerated password' radio button is selected. The 'Custom password' radio button is unselected. The 'Show password' checkbox is unselected. The 'Users must create a new password at next sign-in - Recommended' checkbox is unselected. A blue box at the bottom contains a note: 'If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)'

- Click **Next: Permissions**.
- On the Set permissions page, select **Attach policies directly** or **Add user to group**.
- Click **Next: Tags**.

- Add any tags if needed (optional).
- Click **Next: Review**.
- Review the details and click **Create user**.

Repeat these steps for each user: Dev1, Dev2, Test1, and Test2.

<input type="checkbox"/>	Dev1	/	0	-	-	-	-
<input type="checkbox"/>	Dev2	/	0	-	-	-	-
<input type="checkbox"/>	Test1	/	0	-	-	-	-
<input type="checkbox"/>	Test2	/	0	-	-	-	-

2. Create IAM Groups

1. In the IAM Dashboard, click on **User groups** in the left navigation pane.
2. Click on **Create group**.
3. For **Group name**, enter **Dev Team**.
4. Click **Next step**.
5. Attach policies as needed (optional).
6. Click **Create group**.

Repeat these steps to create the **Ops Team** group.

<input type="checkbox"/>	DevTeam	⚠ 0	⚠ Not defined
<input type="checkbox"/>	OpsTeam	⚠ 0	⚠ Not defined

3. Add Dev1 and Dev2 to the Dev Team

1. Go to the **Dev Team** group.
2. Click on **Add users to group**.
3. Select **Dev1** and **Dev2**.
4. Click **Add users**.

Users in this group (2)						Remove	Add users
An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.							
<input type="text" value="Search"/>					< 1 > ⚙		
<input type="checkbox"/>	User name	▲	Groups	Last activity ▼	Creation time ▼		
<input type="checkbox"/>	Dev1		1	None	6 minutes ago		
<input type="checkbox"/>	Dev2		1	None	5 minutes ago		

4. Add Dev1, Test1, and Test2 to the Ops Team

1. Go to the **Ops Team** group.
2. Click on **Add users to group**.
3. Select **Dev1**, **Test1**, and **Test2**.

4. Click **Add users**.

Users in this group (3)

Remove

Add users

An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.

Q Search

< 1 > ⚙

<input type="checkbox"/>	User name	Groups	Last activity	Creation time
<input type="checkbox"/>	Dev1	2	None	7 minutes ago
<input type="checkbox"/>	Test1	1	None	6 minutes ago
<input type="checkbox"/>	Test2	1	None	5 minutes ago

5. Create Policy Number 1

1. **Login to AWS Management Console.**
2. **Navigate to IAM** (Identity and Access Management).
3. In the left navigation pane, click on **Policies**.
4. Click on **Create policy**.

Q Search IAM

Dashboard

▼ Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

Step 1: Add S3 Full Access

1. Under the **Service** dropdown, type and select **S3**.
2. Under **Actions**, check the box for **All S3 actions**.
3. Under **Resources**, check the box for **All resources**.

Step 2: Add EC2 RunInstances Permission

1. Click on **Add additional permissions**.
2. Under the **Service** dropdown, type and select **EC2**.
3. Under **Actions**, check the box for **Specific actions**.

4. Expand the **Write** section and check the box for **RunInstances**.
5. Under **Resources**, check the box for **All resources**.

Step 3: Add RDS Full Access

1. Click on **Add additional permissions**.
2. Under the **Service** dropdown, type and select **RDS**.
3. Under **Actions**, check the box for **All RDS actions**.
4. Under **Resources**, check the box for **All resources**.
5. Click on **Next: Tags** (optional).
6. Click on **Next: Review**.
7. Enter a **Name** (e.g., PolicyNumber1) and **Description** (optional).
8. Click on **Create policy**.

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				
<input type="text" value="Search"/>				
Allow (3 of 420 services)				<input type="checkbox"/> Show remaining 417 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	

Create Policy Number 2

1. In the left navigation pane, click on **Policies**.
2. Click on **Create policy**.
3. Click on the **Visual editor** tab.

Step 1: Add CloudWatch Full Access

1. Under the **Service** dropdown, type and select **CloudWatch**.
2. Under **Actions**, check the box for **All CloudWatch actions**.
3. Under **Resources**, check the box for **All resources**.

Step 2: Add Billing Full Access

1. Click on **Add additional permissions**.
2. Under the **Service** dropdown, type and select **Billing**.
3. Under **Actions**, check the box for **All Billing actions**.

Step 3: Add EC2 List Permission

1. Click on **Add additional permissions**.
2. Under the **Service** dropdown, type and select **EC2**.
3. Under **Actions**, check the box for **Specific actions**.
4. Expand the **List** section and check the boxes for **DescribeInstances** and **DescribeVolumes**.
5. Under **Resources**, check the box for **All resources**.

Step 4: Add S3 List Permission

1. Click on **Add additional permissions**.
2. Under the **Service** dropdown, type and select **S3**.
3. Under **Actions**, check the box for **Specific actions**.
4. Expand the **List** section and check the box for **ListBucket**.
5. Under **Resources**, check the box for **All resources**.
6. Click on **Next: Tags** (optional).
7. Click on **Next: Review**.
8. Enter a **Name** (e.g., PolicyNumber2) and **Description** (optional).
9. Click on **Create policy**.

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				
<input type="text" value="Search"/>				
Allow (4 of 420 services)				<input type="checkbox"/> Show remaining 416 services
Service ▲	Access level ▼	Resource	Request condition	
Billing	Full access	All resources	None	
CloudWatch	Full access	All resources	None	
EC2	Limited: List	All resources	None	
S3	Limited: List	All resources	None	

6. Attach Policy Number 1 to the Dev Team

1. In the IAM Dashboard, click on **User groups** in the left navigation pane.
2. Select **Dev Team**.
3. Click on the **Permissions** tab.
4. Click on **Add permissions**.
5. Select **Attach policies**.
6. Search for PolicyNumber1.
7. Select the policy and click on **Next: Review**.

8. Click on **Add permissions**.

The screenshot shows the AWS IAM console for the 'DevTeam' user group. The 'Permissions' tab is selected, showing one attached policy: 'Policy1'. The interface includes a summary section with details like 'User group name: DevTeam', 'Creation time: August 01, 2024, 08:51 (UTC+05:30)', and 'ARN: arn:aws:iam::016877529802:group/DevTeam'. Below the summary, there are tabs for 'Users (2)', 'Permissions', and 'Access Advisor'. The 'Permissions policies (1)' section includes a search bar, a 'Filter by Type' dropdown set to 'All types', and a table listing the attached policy.

<input type="checkbox"/>	Policy name ↗	Type	Attached entities
<input type="checkbox"/>	Policy1	Customer managed	1

7. Attach Policy Number 2 to the Ops Team

1. In the IAM Dashboard, click on **User groups** in the left navigation pane.
2. Select **Ops Team**.
3. Click on the **Permissions** tab.
4. Click on **Add permissions**.
5. Select **Attach policies**.
6. Search for PolicyNumber2.
7. Select the policy and click on **Next: Review**.
8. Click on **Add permissions**.

The screenshot shows the AWS IAM console for the 'OpsTeam' user group. The 'Permissions' tab is selected, showing one attached policy: 'Policy2'. The interface includes a summary section with details like 'User group name: OpsTeam', 'Creation time: August 01, 2024, 08:51 (UTC+05:30)', and 'ARN: arn:aws:iam::016877529802:group/OpsTeam'. Below the summary, there are tabs for 'Users (3)', 'Permissions', and 'Access Advisor'. The 'Permissions policies (1)' section includes a search bar, a 'Filter by Type' dropdown set to 'All types', and a table listing the attached policy.

<input type="checkbox"/>	Policy name ↗	Type	Attached entities
<input type="checkbox"/>	Policy2	Customer managed	1

Steps to Create a Role and Test It

Steps to Create the Role

1. Login to AWS Management Console.
2. Navigate to IAM (Identity and Access Management).
3. In the left navigation pane, click on Roles.
4. Click on Create role.

Step 1: Select Trusted Entity

1. Under Trusted entity type, select AWS account.
2. Select This account.
3. Check the box for Require external ID and enter a unique identifier (optional).
4. Click on Next: Permissions.

Step 2: Attach Policies

Create a Custom Policy for Complete Access to VPCs and DynamoDB

1. In the IAM Dashboard, click on Policies in the left navigation pane.
2. Click on Create policy.
3. Click on the Visual editor tab.
4. Under Service, type and select VPC.
5. Under Actions, check the box for All VPC actions.
6. Under Resources, check the box for All resources.
7. Click on Add additional permissions.
8. Under Service, type and select DynamoDB.
9. Under Actions, check the box for All DynamoDB actions.
10. Under Resources, check the box for All resources.
11. Click on Next: Tags (optional).
12. Click on Next: Review.
13. Enter a Name (e.g., VPCDynamoDBFullAccess) and Description (optional).
14. Click on Create policy.

Attach the Custom Policy to the Role

1. Go back to Create role.
2. Click on Refresh in the Attach permissions policies step.
3. Search for VPCDynamoDBFullAccess.
4. Select the policy and click on Next: Tags (optional).
5. Click on Next: Review.

Step 3: Role Name and Description

1. Enter a Role name (e.g., VPCDynamoDBRole) and Description (optional).
2. Click on Create role.

2. Allow Only Specific Users to Assume the Role

Edit the Trust Relationship

1. In the IAM Dashboard, click on Roles in the left navigation pane.
2. Select the role you created (e.g., VPCDynamoDBRole).
3. Go to the Trust relationships tab.
4. Click on Edit trust relationship.
5. Replace the existing policy with the following JSON, modifying AccountID with your AWS account ID and replacing User1 and User2 with the actual user names:

json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::AccountID:user/User1",
          "arn:aws:iam::AccountID:user/User2"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

6. Click on Update Trust Policy.

3. Test the Role with User1

Login as User1

1. Sign out from your current AWS session.

2. Login as User1.

Assume the Role

1. In the AWS Management Console, navigate to IAM.
2. In the left navigation pane, click on Roles.
3. Find and click on the role you created (e.g., VPCDynamoDBRole).
4. Click on Assume role.
5. Confirm the switch.

Test Access

1. Navigate to VPC and DynamoDB services in the AWS Management Console.
2. Ensure that User1 has complete access to both VPCs and DynamoDB.