

Module 3: IAM Users Assignment

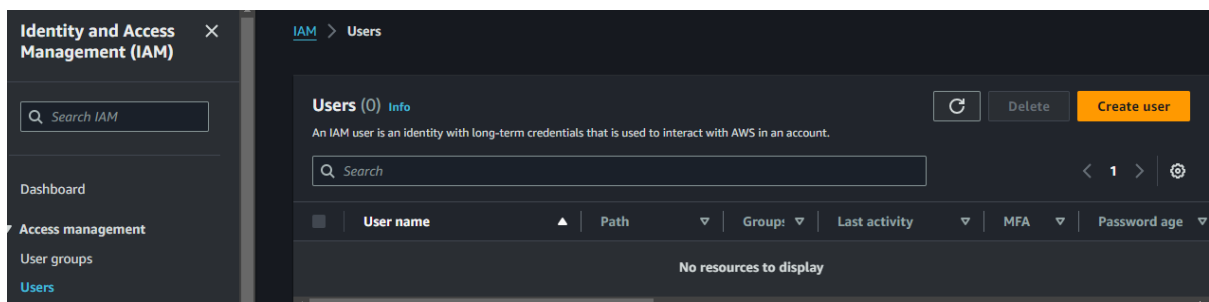
Problem Statement:

You work for XYZ Corporation. To maintain the security of the AWS account and the resources you have been asked to implement a solution that can help easily recognize and monitor the different users.

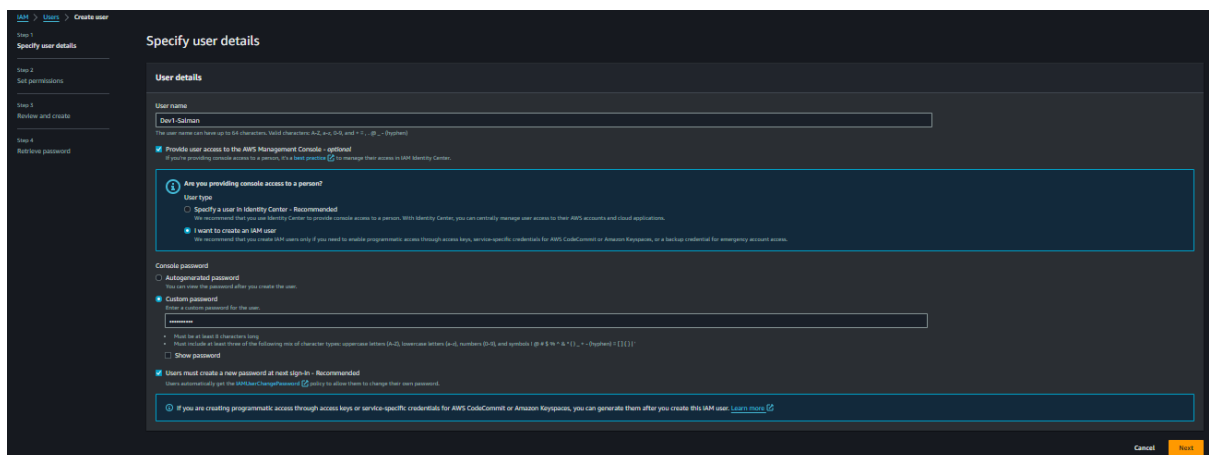
Tasks To Be Performed:

1. Create 4 IAM users named "Dev1", "Dev2", "Test1", and "Test2".
2. Create 2 groups named "Dev Team" and "Ops Team".
3. Add Dev1 and Dev2 to the Dev Team.
4. Add Dev1, Test1 and Test2 to the Ops Team.

1. Creating 4 IAM Users So Need to go IAM Console > Users > Create User



2. Select Name As per the Assignment and Provide Console Access and Select Custom or Autogenerated Password as your wish.



3. Add User to Group and Create Group From Here Or Click Next, later also can Add in Group

IAM > Users > Create user

Step 1
[Specify user details](#)

Step 2
Set permissions

Step 3
Review and create

Step 4
Retrieve password

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

☒ **Add user to group**
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ **Copy permissions**
Copy all group memberships, attached managed policies, and inline policies from an existing user.

☐ **Attach policies directly**
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Get started with groups

Create a group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. [Learn more](#)

Create group

► **Set permissions boundary - optional**

Cancel Previous **Next**

4. Created a Group As Per Assignment Dev-Team and Click Next

Step 1
[Specify user details](#)

Step 2
Set permissions

Step 3
Review and create

Step 4
Retrieve password

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

☒ **Add user to group**
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ **Copy permissions**
Copy all group memberships, attached managed policies, and inline policies from an existing user.

☐ **Attach policies directly**
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

User groups (1)

↻ **Create group**

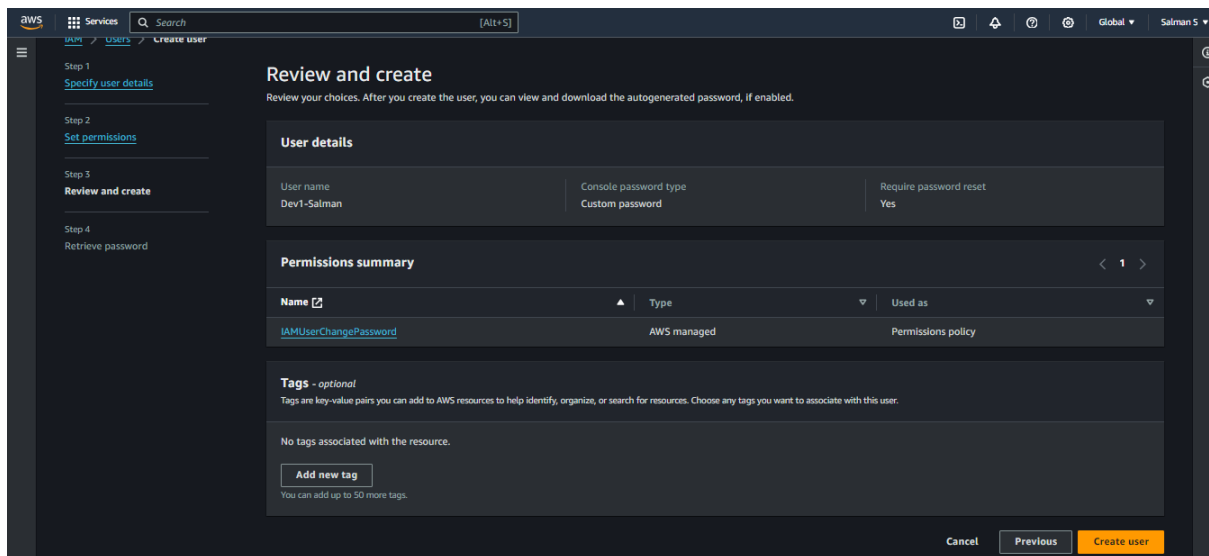
◀ 1 ▶ ⚙

<input type="checkbox"/>	Group name	Users	Attached policies	Created
<input type="checkbox"/>	Dev-Team	0	-	2024-03-22 (Now)

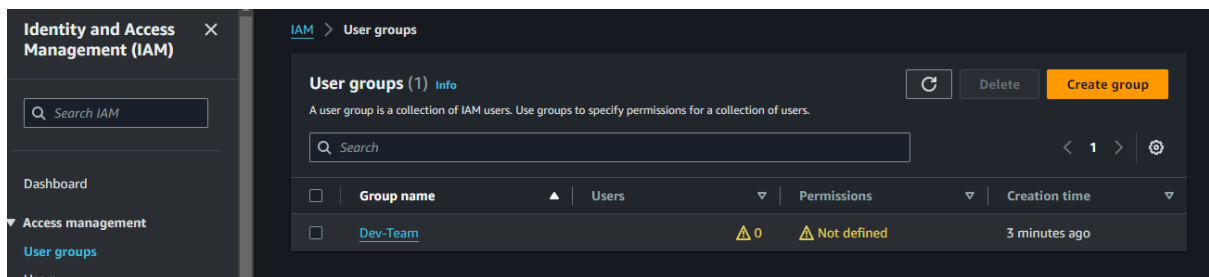
► **Set permissions boundary - optional**

Cancel Previous **Next**

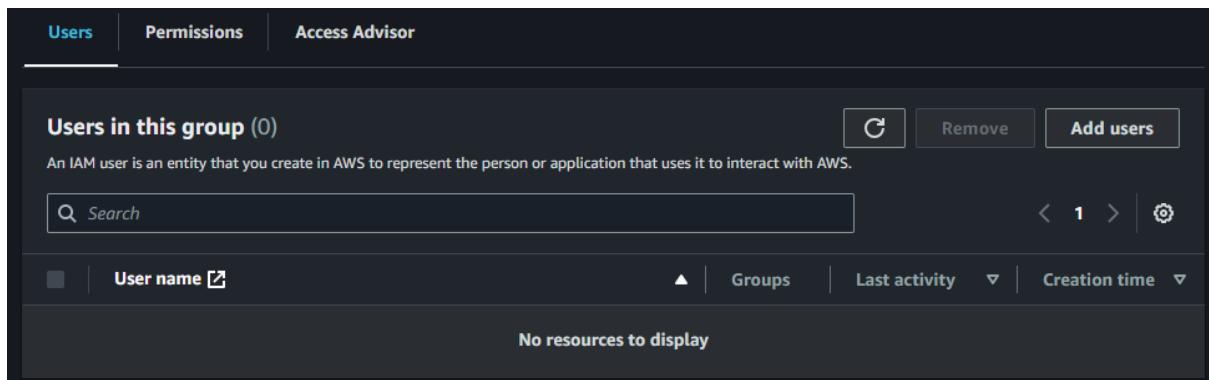
5. Review and Click **Create User**



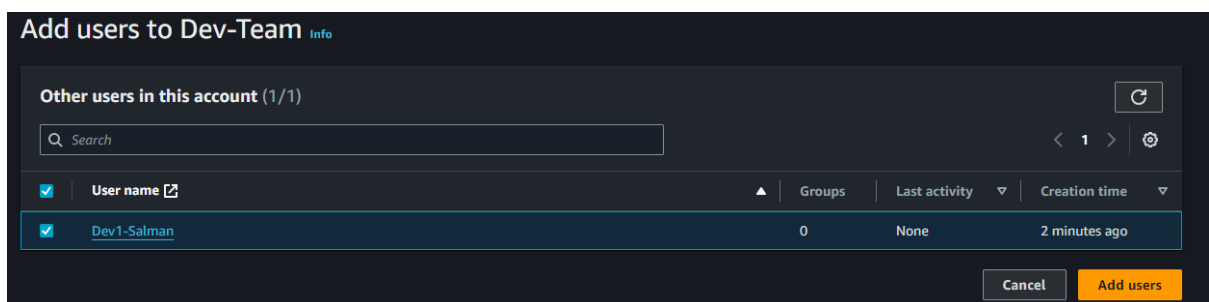
6. I **Created Group** while Creating Users



7. Click **Dev-Team** and **Go Users** > Add Users on **right side end**



8. And Add **Dev1-Salman** in **Dev-Team**



9. Create another 3 Users From Following Above the Steps and also Create Another Group

The screenshot shows the AWS IAM console 'Users' page. The left sidebar contains navigation links: Dashboard, Access management (User groups, Users, Roles, Policies), Identity providers, Account settings, and Access reports. The main content area is titled 'Users (4) Info' and includes a search bar and a table of users.

<input type="checkbox"/>	User name	Path	Group:	Last activity	MFA	Password age
<input type="checkbox"/>	Dev1-Salman	/	1	-	-	9 minutes
<input type="checkbox"/>	Dev2-Salman	/	1	-	-	-
<input type="checkbox"/>	Test1-Salman	/	0	-	-	-
<input type="checkbox"/>	Test2-Salman	/	0	-	-	-

10. Add Dev2-Salman in Dev-Team

The screenshot shows the AWS IAM console 'User groups' page. The left sidebar contains navigation links: Identity and Access Management (IAM), Dashboard, Access management (User groups, Users, Roles, Policies), Identity providers, Account settings, and Access reports. The main content area is titled 'User groups (1) Info' and includes a search bar and a table of user groups.

<input type="checkbox"/>	Group name	Users	Permissions	Creation time
<input type="checkbox"/>	Dev-Team	2	Not defined	28 minutes ago

11. Now I'm Adding 3 Users in Ops-Team As per the Assignment

The screenshot shows the 'Add users to the group - Optional (3/4)' page in the AWS IAM console. The left sidebar contains navigation links: Dashboard, Access management (User groups, Users, Roles, Policies), Identity providers, Account settings, and Access reports. The main content area is titled 'Name the group' and includes a search bar and a table of users to be added to the group.

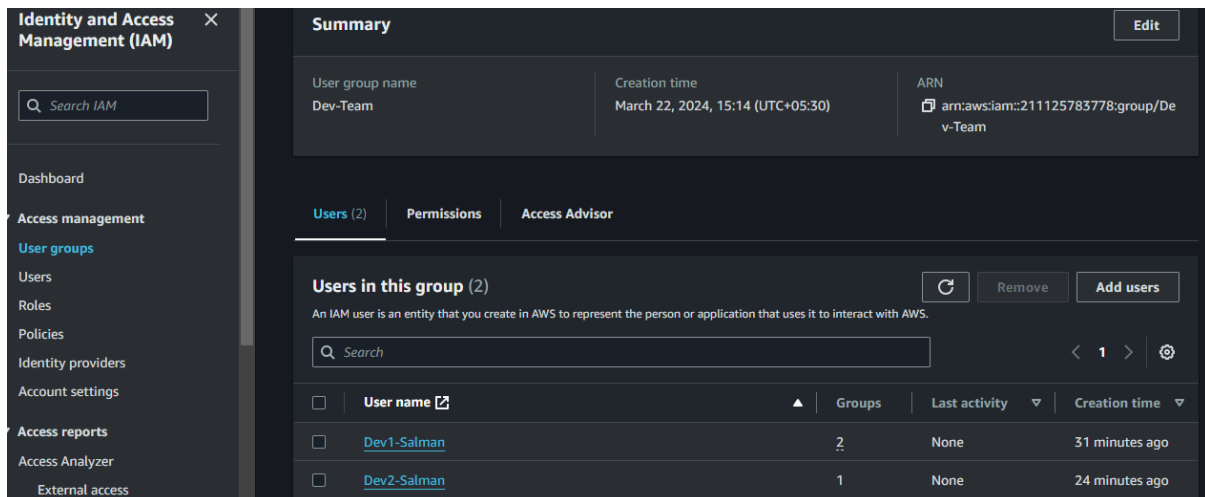
<input type="checkbox"/>	User name	Groups	Last activity	Creation time
<input checked="" type="checkbox"/>	Dev1-Salman	1	None	30 minutes ago
<input type="checkbox"/>	Dev2-Salman	1	None	23 minutes ago
<input checked="" type="checkbox"/>	Test1-Salman	0	None	21 minutes ago
<input checked="" type="checkbox"/>	Test2-Salman	0	None	20 minutes ago

12. Now in the Dev-Team Having 2 Users and Ops-Team Having 3 users

The screenshot shows the AWS IAM console 'User groups' page. The left sidebar contains navigation links: Dashboard, Access management (User groups, Users, Roles, Policies), Identity providers, Account settings, and Access reports. The main content area is titled 'User groups (1) Info' and includes a search bar and a table of user groups.

<input type="checkbox"/>	Group name	Users	Permissions	Creation time
<input type="checkbox"/>	Dev-Team	2	Not defined	32 minutes ago
<input type="checkbox"/>	Ops-Team	3	Not defined	Now

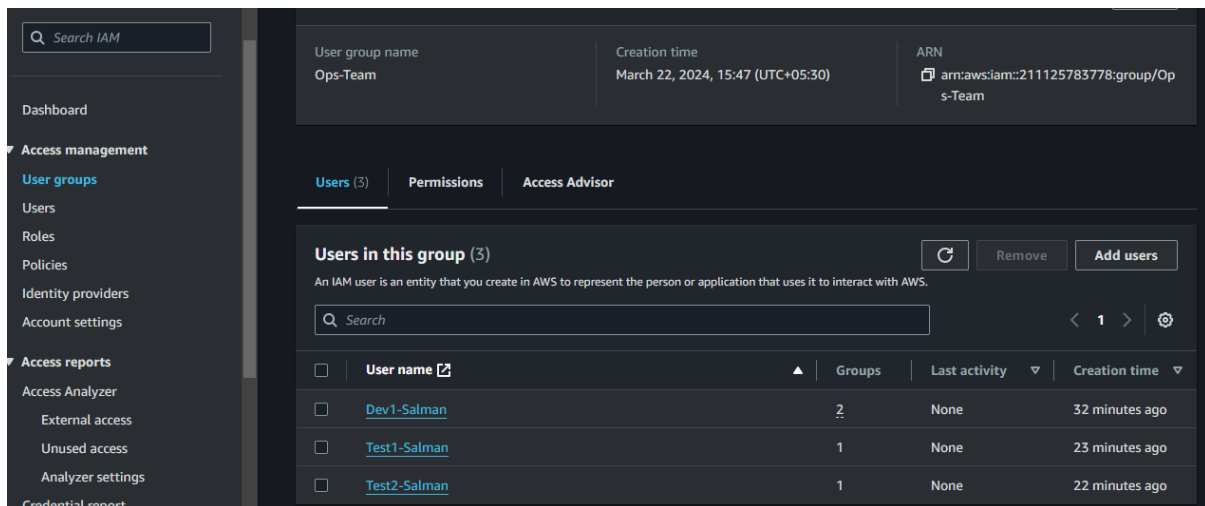
13. Now we can see As per the Assignment **Dev1** in **2 Groups** and This **Dev-Team**



The screenshot shows the AWS IAM console interface. On the left is a navigation sidebar with options like Dashboard, Access management, User groups, Users, Roles, Policies, Identity providers, Account settings, Access reports, Access Analyzer, and External access. The main content area is titled 'Summary' for the 'Dev-Team' user group. It displays the group name, creation time (March 22, 2024, 15:14 UTC+05:30), and ARN. Below this, there are tabs for 'Users (2)', 'Permissions', and 'Access Advisor'. The 'Users (2)' tab is active, showing a list of users in the group. The list has columns for checkboxes, User name, Groups, Last activity, and Creation time. Two users are listed: 'Dev1-Salman' and 'Dev2-Salman', both with 'None' last activity and creation times of 31 and 24 minutes ago respectively.

	User name	Groups	Last activity	Creation time
<input type="checkbox"/>	Dev1-Salman	2	None	31 minutes ago
<input type="checkbox"/>	Dev2-Salman	1	None	24 minutes ago

14. Now this is **Ops-Team** Creating and Adding the **Users in Groups** Successfully Completed



The screenshot shows the AWS IAM console interface for the 'Ops-Team' user group. The navigation sidebar is the same as in the previous screenshot. The main content area shows the 'Summary' for 'Ops-Team', with creation time (March 22, 2024, 15:47 UTC+05:30) and ARN. The 'Users (3)' tab is active, showing a list of three users: 'Dev1-Salman', 'Test1-Salman', and 'Test2-Salman'. The 'Groups' column shows 2, 1, and 1 group respectively. The 'Last activity' is 'None' for all, and 'Creation time' is 32, 23, and 22 minutes ago respectively.

	User name	Groups	Last activity	Creation time
<input type="checkbox"/>	Dev1-Salman	2	None	32 minutes ago
<input type="checkbox"/>	Test1-Salman	1	None	23 minutes ago
<input type="checkbox"/>	Test2-Salman	1	None	22 minutes ago

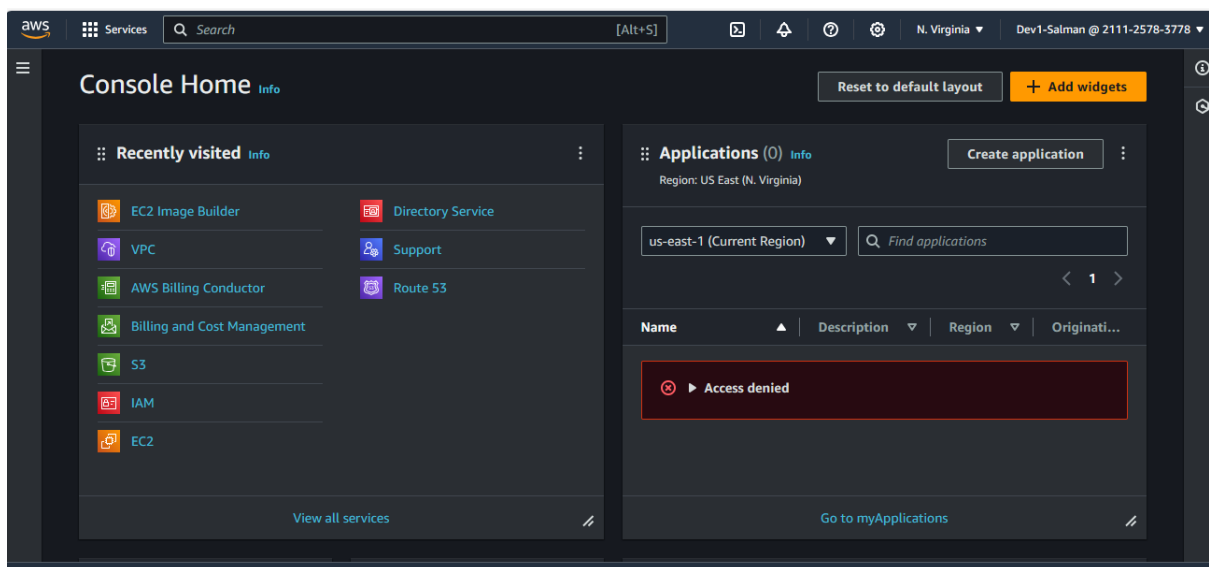
Module 3: IAM Policies Assignment

Problem Statement:

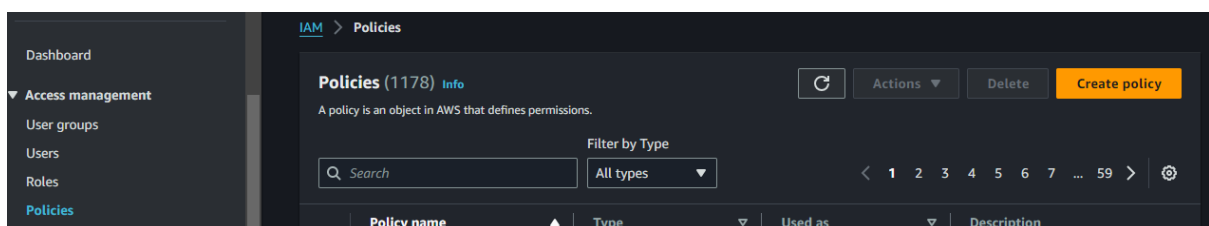
You work for XYZ Corporation. To maintain the security of the AWS account and the resources you have been asked to implement a solution that can help easily recognize and monitor the different users.

Tasks To Be Performed:

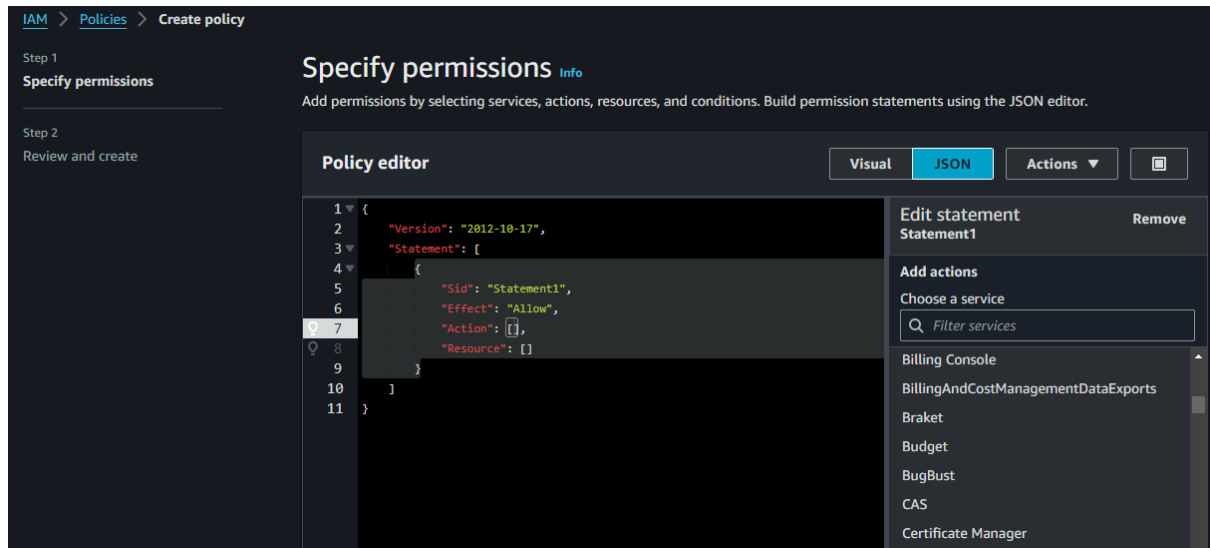
1. Create policy number 1 which lets the users to:
 - a. Access S3 completely
 - b. Only create EC2 instances
 - c. Full access to RDS
 2. Create a policy number 2 which allows the users to:
 - a. Access CloudWatch and billing completely
 - b. Can only list EC2 and S3 resources
 3. Attach policy number 1 to the Dev Team from task 1
 4. Attach policy number 2 to Ops Team from task 1
15. Now I am Logged in as Dev1-User Checking that have Permission to Use? Now we see No Permission



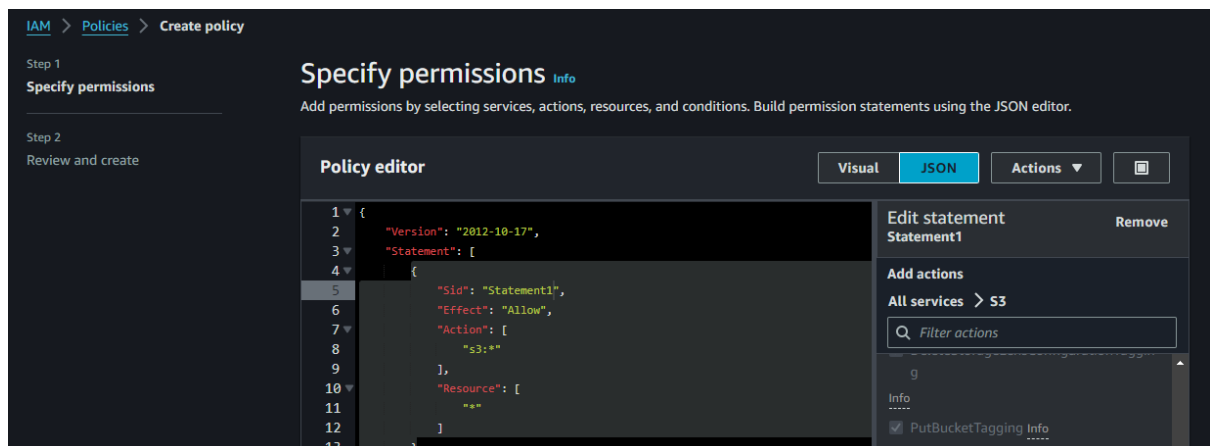
16. Now Assignment-2 Create Policies Go to IAM Dashboard > Policies > Create Policy



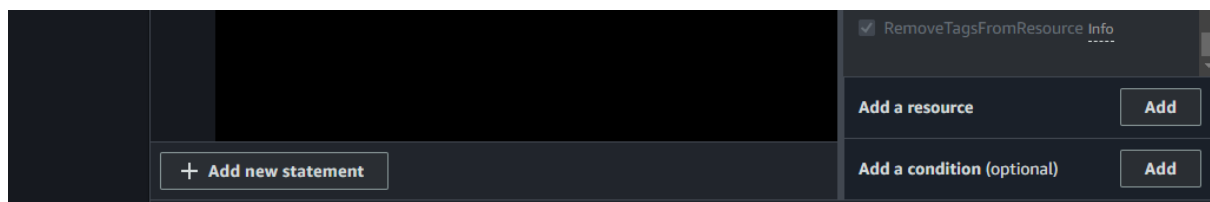
17. Now to Add Permissions using **visual** or **Json** and I am Using **Json**



18. So In the **Statement1**, I Add **s3** and **Allow All Services** for **s3** and Click **All Resources** and **Allow** it



19. We can **Add New statement** and **Add a Resource Option** and Add a **Condition Option** too



20. Added Statement2 and Allowed RDS All Services and Allowed All Resources

```
{
  "Sid": "Statement2",
  "Effect": "Allow",
  "Action": [
    "rds:*"
  ],
  "Resource": [
    "*"
  ]
},
```

21. Adding Statement 3 only Allow to Create EC2 Instances EC2 but, find Below the Services which allowed to EC2 as per the task

```
{
  "Sid": "Statement3",
  "Effect": "Allow",
  "Action": [
    "ec2:RunInstances",
    "ec2:CreateTags",
    "ec2:DescribeImages",
    "ec2:DescribeInstanceTypes",
    "ec2:DescribeKeyPairs",
    "ec2:CreateKeyPair",
    "ec2:DescribeVpcs",
    "ec2:DescribeSubnets",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeSecurityGroupRules",
    "ec2:DescribeVolumes",
    "ec2:CreateVolume",
    "ec2:AttachVolume",
    "ec2:DescribeNetworkInterfaces",
    "ec2:CreateNetworkInterface",
    "ec2:AttachNetworkInterface"
  ],
  "Resource": [
    "*"
  ]
}
```


22. Give a Policy Name and Check your Permissions which defined to policy

[IAM](#) > [Policies](#) > Create policy

Step 1
[Specify permissions](#)

Step 2
Review and create

Review and create [info](#)

Review the permissions, specify details, and tags.

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.

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.

Allow (3 of 405 services) [Show remaining 402 services](#)

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

23. Now go to Groups Section and Go To Dev Team

[IAM](#) > [User groups](#)

Dashboard

Access management

[User groups](#)

[Users](#)

[Roles](#)

[Policies](#)

User groups (2) [Info](#) [Refresh](#) [Delete](#) [Create group](#)

A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

< 1 > [Settings](#)

<input type="checkbox"/>	Group name	Users	Permissions	Creation time
<input type="checkbox"/>	Dev-Team	2	⚠ Not defined	1 hour ago
<input type="checkbox"/>	Ops-Team	3	⚠ Not defined	37 minutes ago

24. As per Task Click on Dev-Team > Permissions > Right Side Add Permissions > Attach Policies

Dashboard

Access management

[User groups](#)

[Users](#)

[Roles](#)

[Policies](#)

[Identity providers](#)

[Account settings](#)

Access reports

[Access Analyzer](#)

[External access](#)

[Unused access](#)

[Analyzer settings](#)

[Credential report](#)

[Organization activity](#)

Dev-Team [Info](#) [Delete](#)

Summary [Edit](#)

User group name	Creation time	ARN
Dev-Team	March 22, 2024, 15:14 (UTC+05:30)	arn:aws:iam::211125783778:group/Dev-Team

[Users \(2\)](#) [Permissions](#) [Access Advisor](#)

Permissions policies (0) [Info](#) [Refresh](#) [Simulate](#) [Remove](#) [Add permissions](#)

You can attach up to 10 managed policies.

Filter by Type [All types](#)

< 1 > [Settings](#)

<input type="checkbox"/>	Policy name	Type	Attached entities
--------------------------	-------------	------	-------------------

25. Select Your Created Policy and Attach Policies

IAM > User groups > Dev-Team > Add permissions

Attach permission policies to Dev-Team

► Current permissions policies (0)

Other permission policies (1/913)

You can attach up to 10 managed policies to this user group. All of the users in this group inherit the attached permissions.

Filter by Type

Q policy X Customer managed 1 match < 1 >

<input checked="" type="checkbox"/>	Policy name	Type	Used as	Description
<input checked="" type="checkbox"/>	PolicyNumber1-Salman	Customer managed	None	-

Cancel Attach policies

26. Now See that Permissions Defined

Dashboard

Access management

User groups

Users

Roles

Policies

A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

Search

<input type="checkbox"/>	Group name	Users	Permissions	Creation time
<input type="checkbox"/>	Dev-Team	2	Defined	1 hour ago
<input type="checkbox"/>	Ops-Team	3	Not defined	38 minutes ago

27. Now Check with Dev1-User use to see and edit s3 bucket because permission defined

aws Services Search [Alt+S] Global Dev1-Salman @ 2111-2578-3778

Amazon S3

Buckets

Access Grants

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

Storage Lens groups

AWS Organizations settings

Last updated: Mar 21, 2024 by Storage Lens. Metrics are generated every 24 hours. Metrics don't include directory buckets. [Learn more](#)

Total storage	Object count
6.2 MB	15
Average object size	You can enable advanced metrics in the "default-account-dashboard" configuration.
421.0 KB	

General purpose buckets Directory buckets

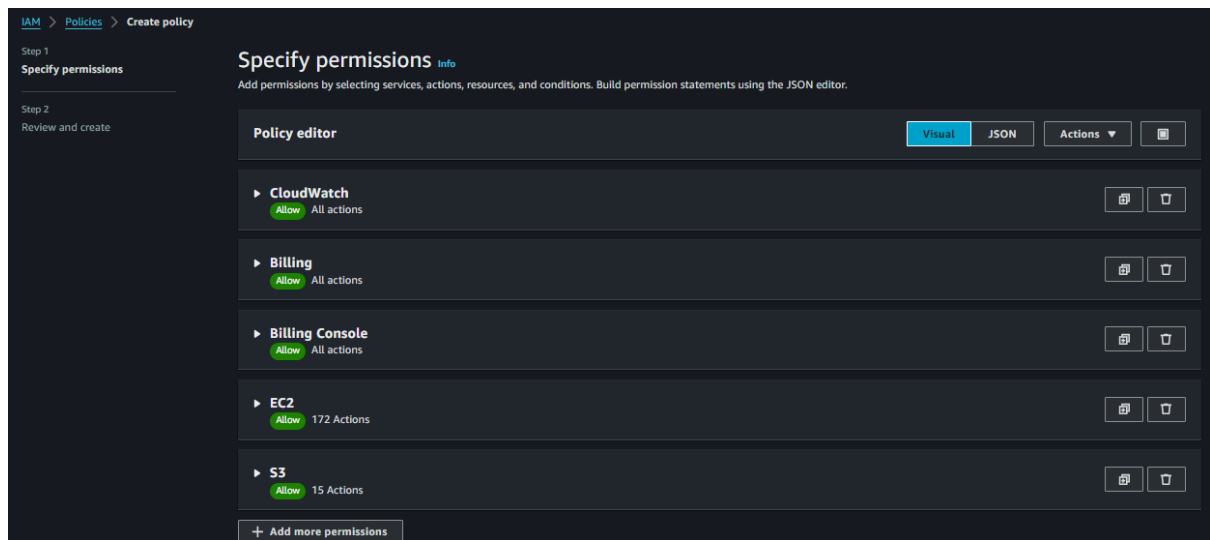
General purpose buckets (1) Info Copy ARN Empty Delete Create bucket

Buckets are containers for data stored in S3.

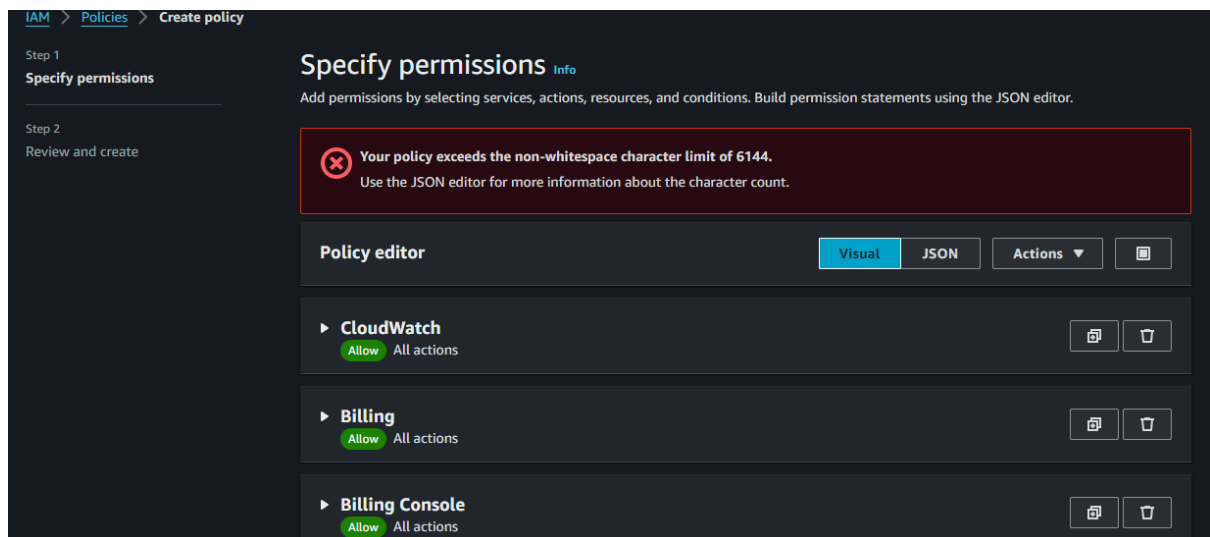
Find buckets by name < 1 >

<input type="radio"/>	Name	AWS Region	Access	Creation date
<input type="radio"/>	salman-march-12-assignment	Asia Pacific (Singapore) ap-southeast-1	Bucket and objects not public	March 12, 2024, 12:04:57 (UTC+05:30)

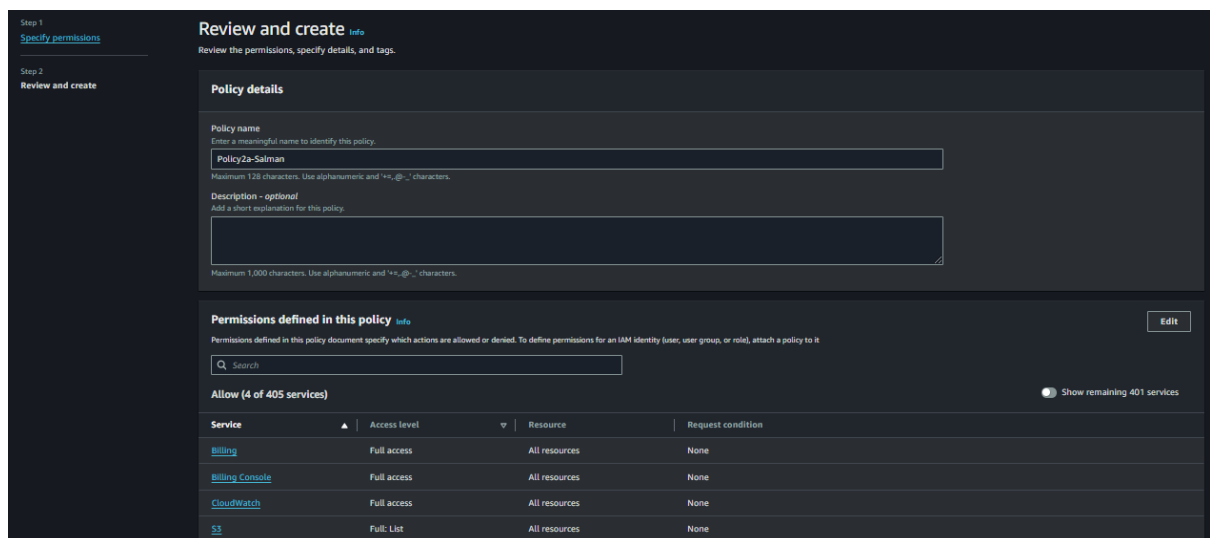
28. Now to go Policies and Create another Policies add Services as per the Task and Go Ahead



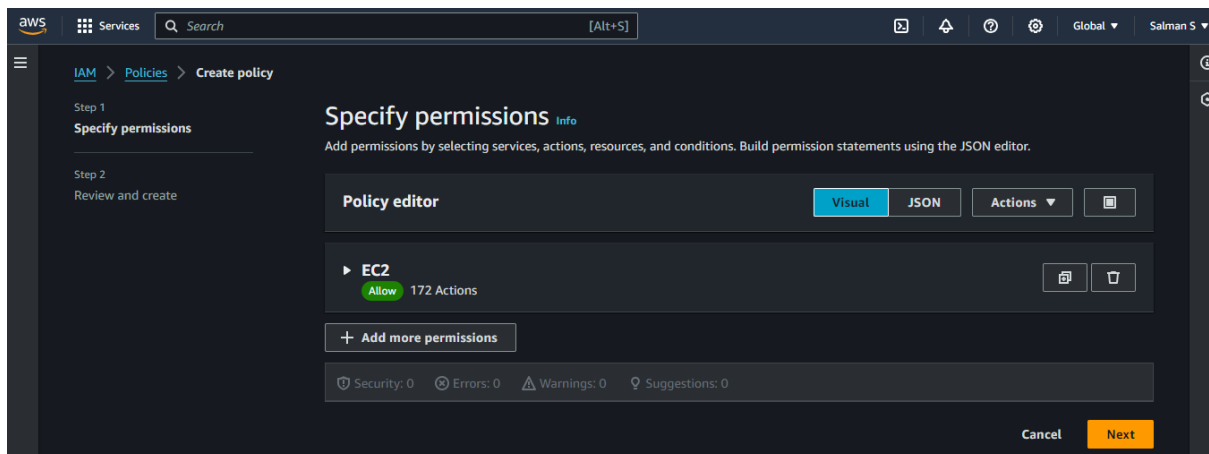
29. Triggered Error Because Non-Whitespace Character Limit of 6144



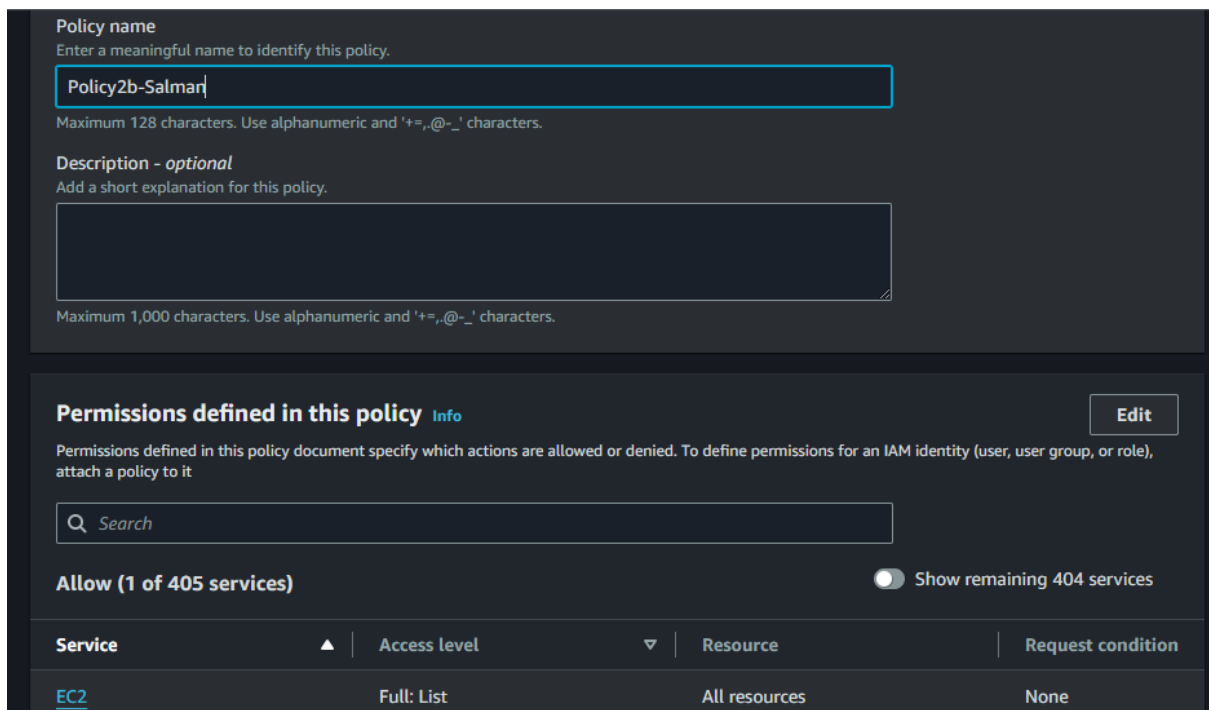
30. So Removed EC2 All Lists and Creating a Policy2a-Salman



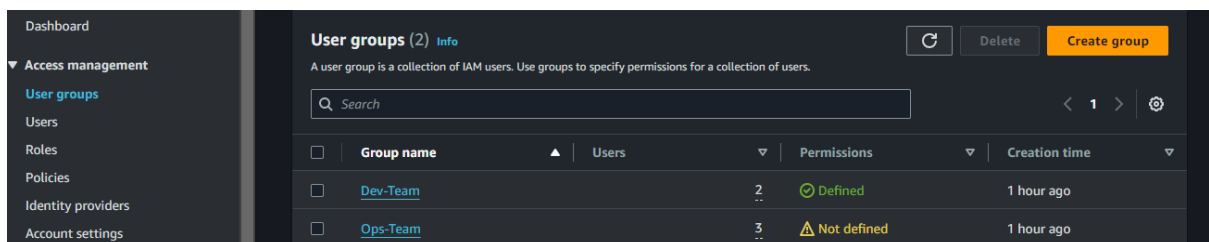
31. Now Again Creating Another Policies **added EC2 All Lists** and Click Next



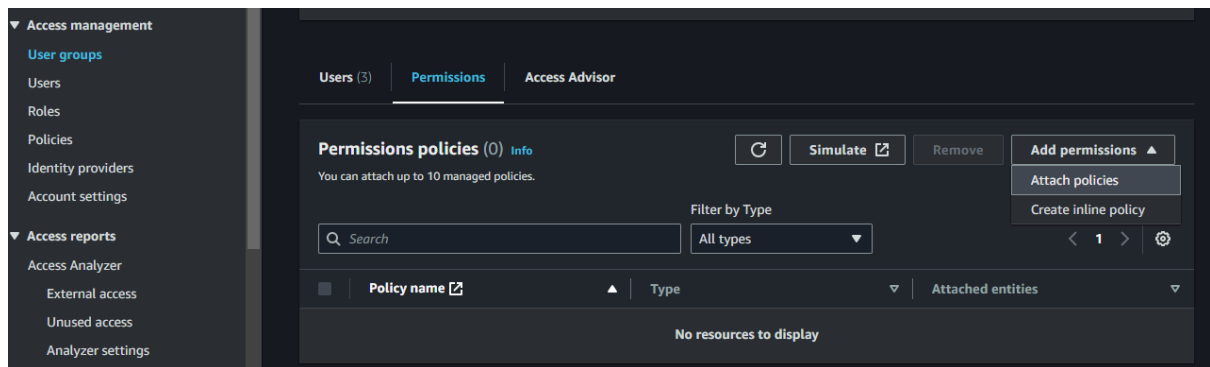
32. Policy2b-Salman



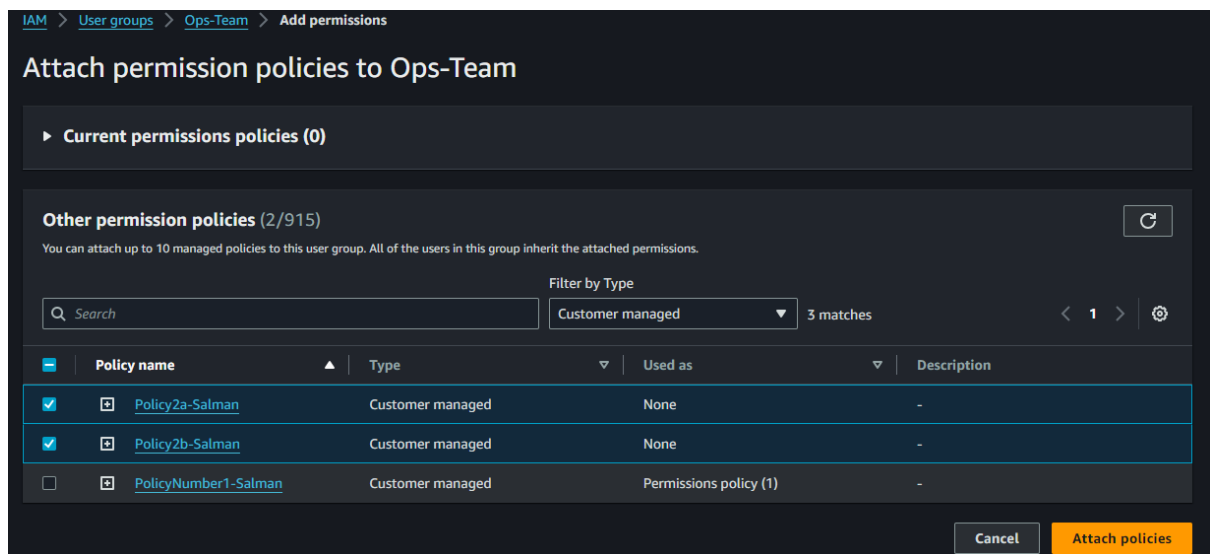
33. Now This Time go to **Ops-Team** As per the Task and Click



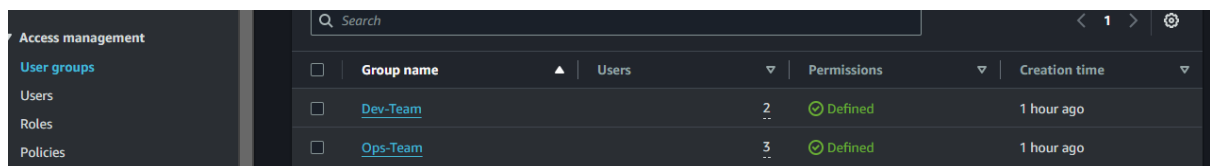
34. Same Click **Attach Policies**



35. Select **2 policies** which we **created for Ops-Team** and Attach it



36. Now its **Defined** to **Ops-Team** and we Created Successfully **Users and Policies**



Module 3: IAM Roles Assignment

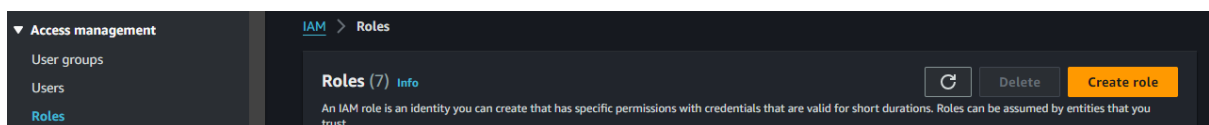
Problem Statement:

You work for XYZ Corporation. To maintain the security of the AWS account and the resources you have been asked to implement a solution that can help easily recognize and monitor the different users.

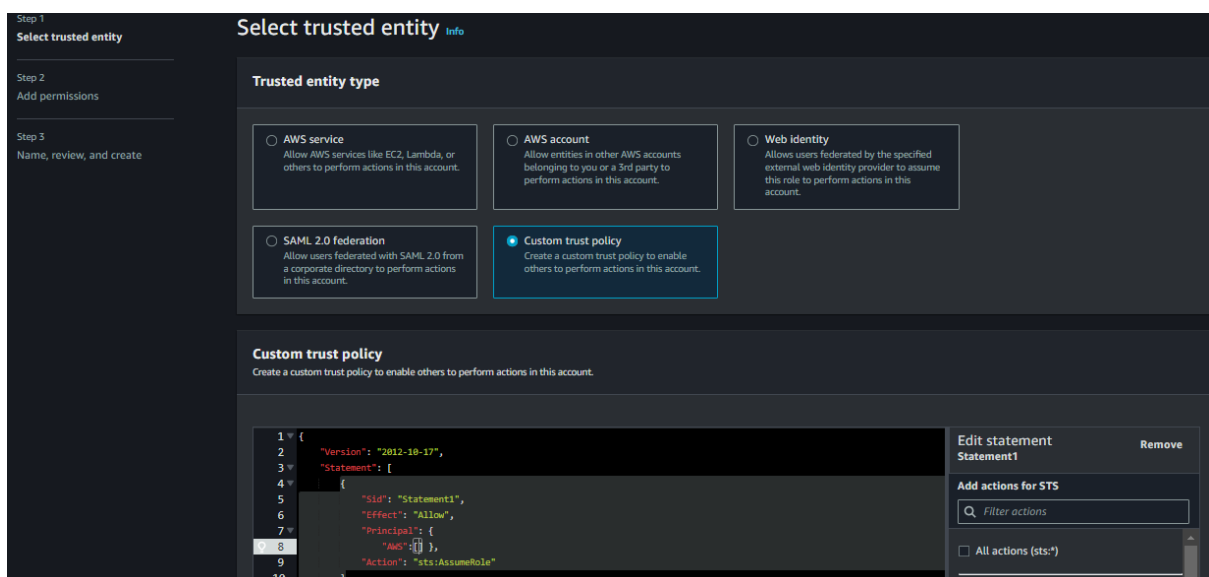
Tasks To Be Performed:

1. Create a role which only lets user1 and user2 from task 1 to have complete access to VPCs and DynamoDB.
2. Login into user1 and shift to the role to test out the feature.

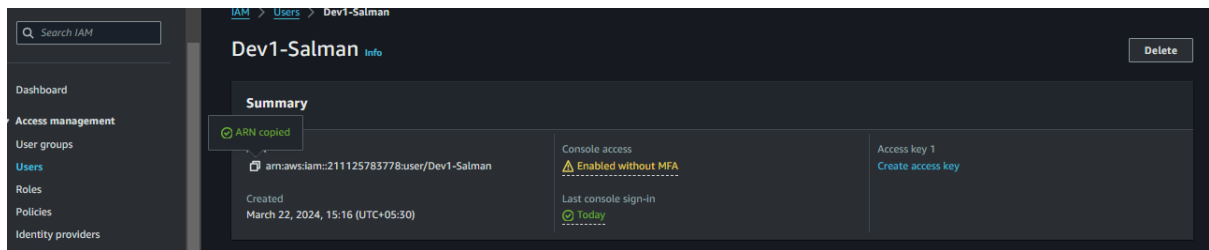
37. Go to [IAM Dashboard](#) > [Roles](#) > [Create Role](#)



38. Select [Custom Trust Policy](#) and [Go to Code](#) > [Principal](#)



39. Go to Users **Copy ARN** of **Dev1-Salman** as **User1** in the Task



Summary

ARN copied

arn:aws:iam::211125783778:user/Dev1-Salman

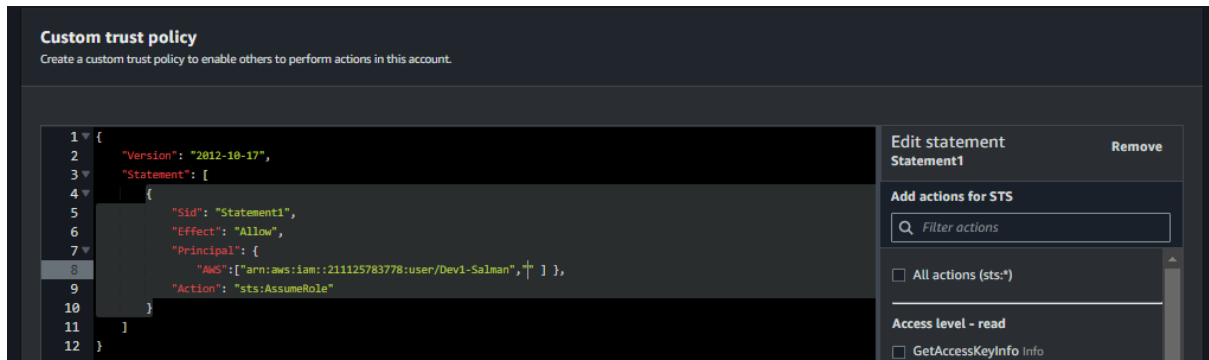
Created: March 22, 2024, 15:16 (UTC+05:30)

Console access: Enabled without MFA

Last console sign-in: Today

Access key 1: Create access key

40. In the Principal part **paste copied user1 ARN**



Custom trust policy

Create a custom trust policy to enable others to perform actions in this account.

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "Statement1",
6       "Effect": "Allow",
7       "Principal": {
8         "AWS": [ "arn:aws:iam::211125783778:user/Dev1-Salman", " " ] },
9       "Action": "sts:AssumeRole"
10    }
11  ]
12 }
```

Edit statement: Statement1

Add actions for STS

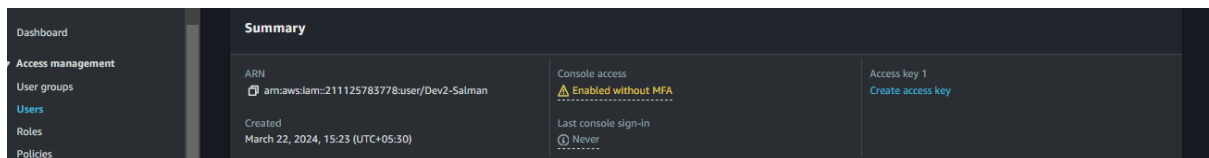
Filter actions

All actions (sts:)

Access level - read

GetAccessKeyInfo

41. Go to **User2** and **Copy ARN**



Summary

ARN: arn:aws:iam::211125783778:user/Dev2-Salman


Created: March 22, 2024, 15:23 (UTC+05:30)

Console access: Enabled without MFA

Last console sign-in: Never

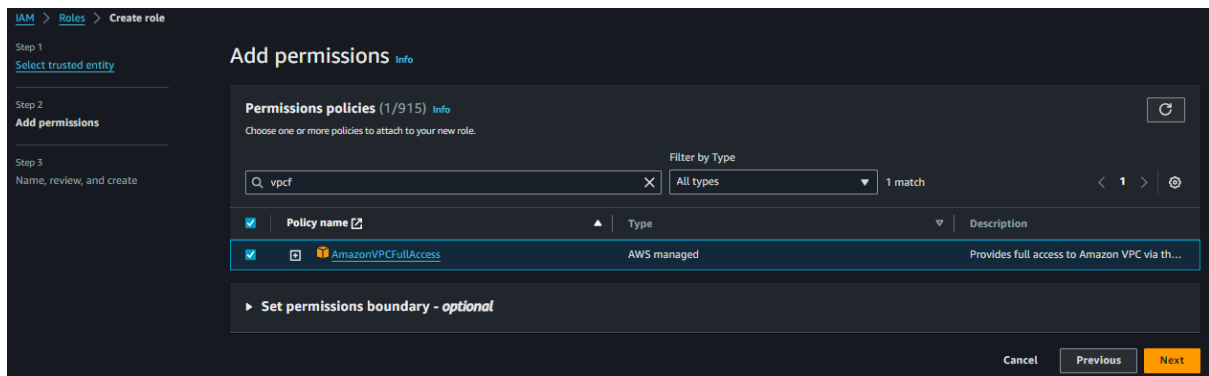
Access key 1: Create access key

42. Again Come **Paste Here**

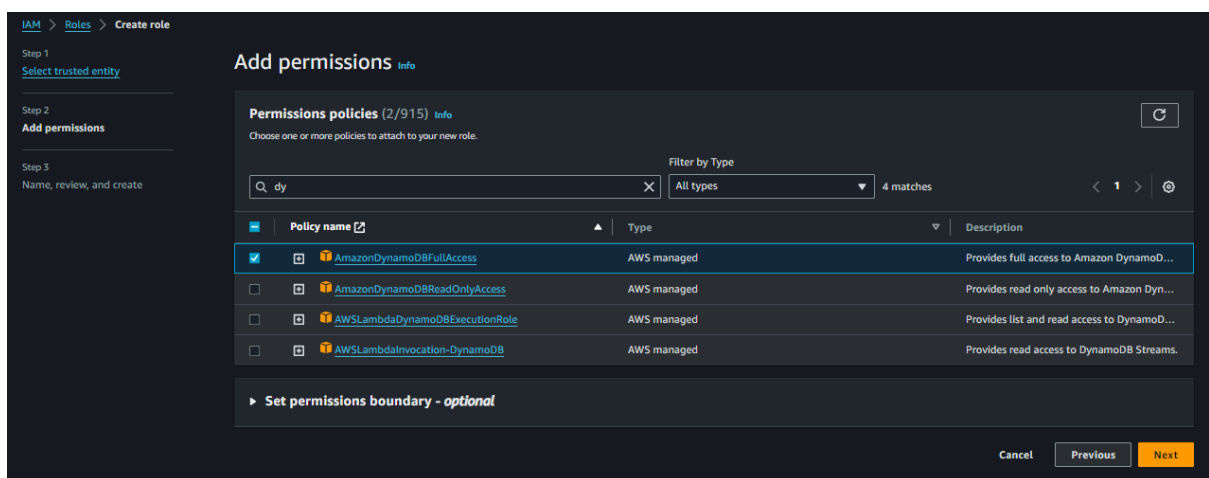


```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "Statement1",
6       "Effect": "Allow",
7       "Principal": {
8         "AWS": [ "arn:aws:iam::211125783778:user/Dev1-Salman", "arn:aws:iam::211125783778:user/Dev2-Salman" ] },
9       "Action": "sts:AssumeRole"
10    }
11  ]
12 }
```

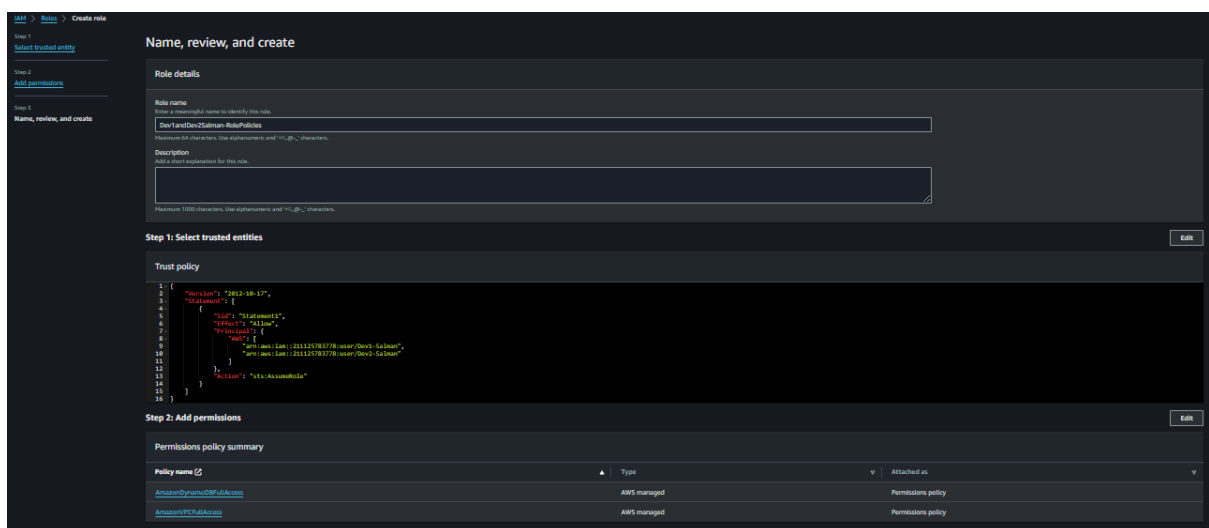
43. After Pasted the ARN and Clicking Next and finding Permissions Polices and Attach VPCfullAccess



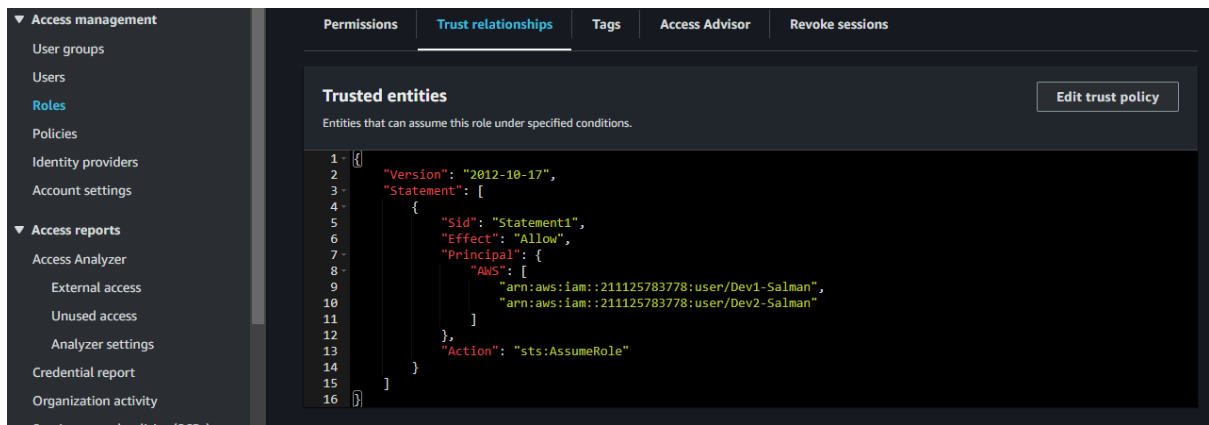
44. And Select Another Permission As per the Assignment DynamoDBFullAccess



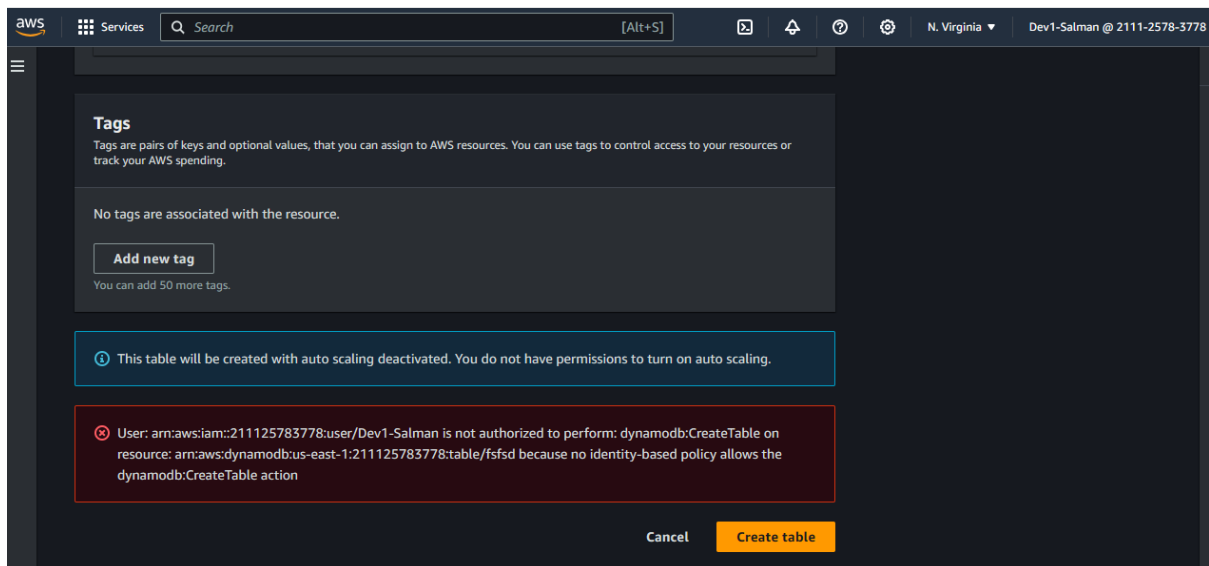
45. Name, Review and Create



46. Created Role and Able to **See the Code here** and Also **We can Edit it**



47. Check ones **Login As a Dev1-Salman** User able to Create **DynamoDB Table** see its saying your **not authorized to create** and Top Right Side Click on Account Id u will **get a Option Switch** role



48. Paste the All Details and Role Name Which U Given and Login

Switch Role

Switching roles enables you to manage resources across Amazon Web Services accounts using a single user. When you switch roles, you temporarily take on the permissions assigned to the new role. When you exit the role, you give up those permissions and get your original permissions back. [Learn more](#)

Account ID

The 12-digit account number or the alias of the account in which the role exists.

211125783778

IAM role name

The name of the role that you want to assume. You can get this from the end of the role's ARN. For example, ARN: arn:aws:iam::111111111111:role/RoleName

Dev1andDev2Salman-RolePolicies

Display name - optional

This name will appear in the console navigation bar when active. Choose a name to help identify the permission set assigned to the role.

Display color - optional

The selected color displays in the console navigation when this role is active

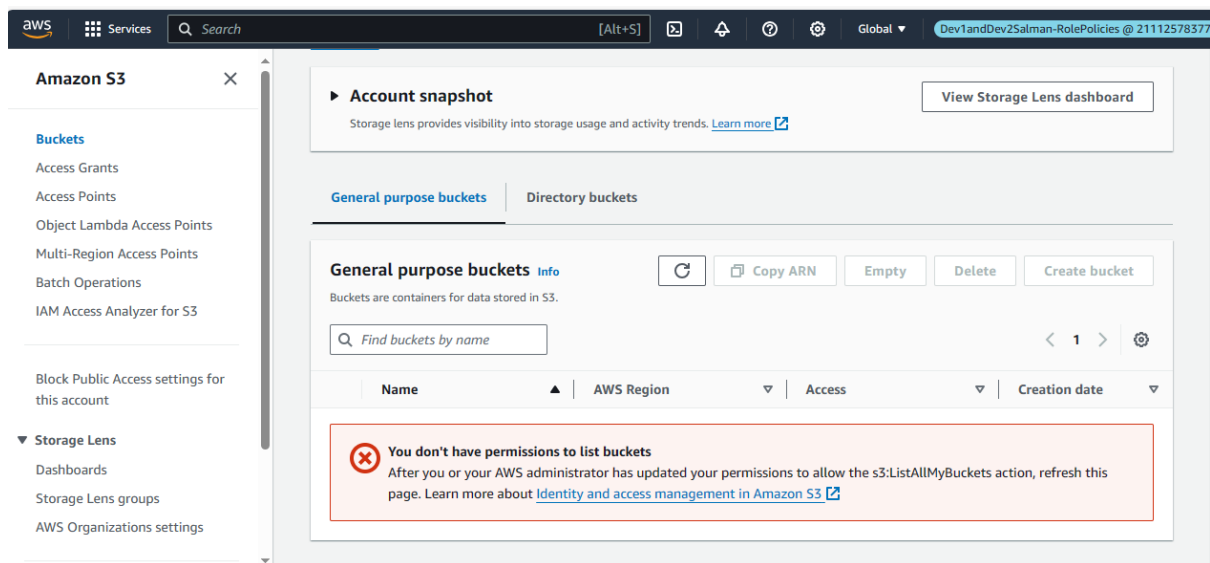
Blue

49. Now Logged as IAM-Role and See Now we able create table in Dynamodb

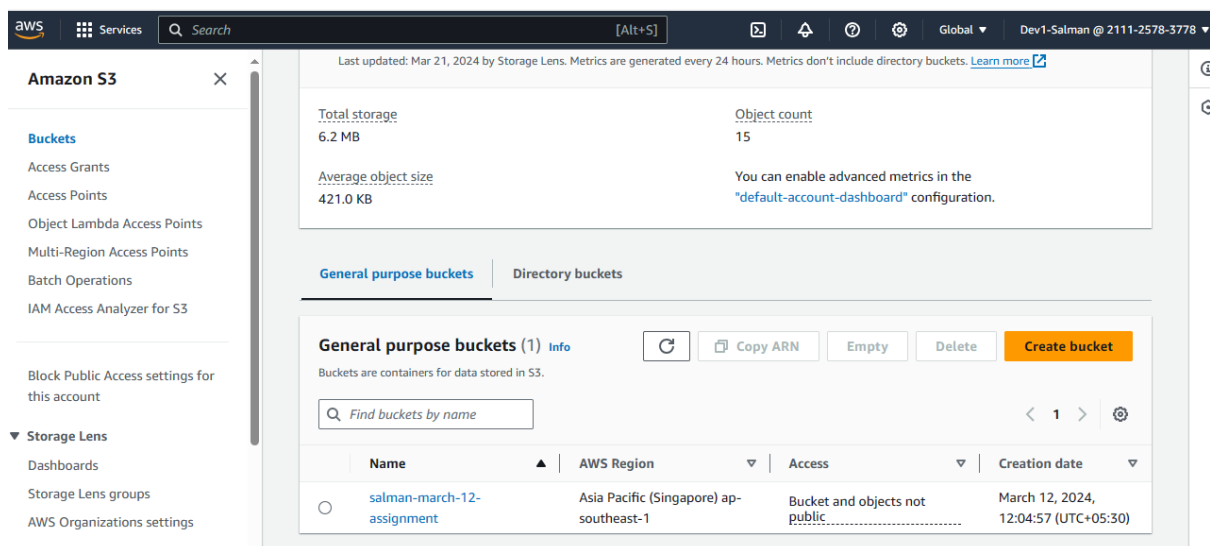
The screenshot shows the AWS Management Console interface for DynamoDB. At the top, a blue banner indicates that the 'sdfs' table is being created and will be available shortly. The left sidebar contains navigation links for Dashboard, Tables, Explore items, PartitQL editor, Backups, Exports to S3, Imports from S3, Integrations, Reserved capacity, and Settings. The main content area is titled 'DynamoDB > Tables' and shows a table with one entry named 'sdfs'. The table's status is 'Creating'. The partition key is 'fsdf (S)' and the sort key is 'fsd (S)'. There are no indexes, and deletion protection is turned off. The read capacity mode is 'Provisioned (5)'. A 'Create table' button is located in the top right corner of the table list.

	Na...	Status	Partition key	Sort key	Indexes	Deletion protection	Read capacity mode
<input type="checkbox"/>	sdfs	Creating	fsdf (S)	fsd (S)	0	Off	Provisioned (5)

50. Previously As a **IAM-User** we Authorize to use s3 but not in **IAM-Role**



51. **Switch Again IAM-User** and Able and See and Create S3 bucket As a **Dev1-user**



***IAM-User :**

IAM users represent individual users who can interact with AWS resources using their own long-term credentials (username and password, access keys, etc.),

***IAM-Role :**

while IAM roles are a way to delegate permissions to entities within or outside your AWS account, such as applications or AWS services. Roles are temporary and can be assumed by users, services, or resources, granting them specific permissions for a limited duration.