POC : IAM (Terraform)

- Identity Access Management is used to manage access to users and resources.
- IAM is universal system. IAM is a free Service.
- A root a/c is the account initially created when AWS is setup.
- New IAM accounts have no Permissions by default until granted.
- New users get assigned an Access Key ID and Secret Key when first created when you give programmatic access.
- Access Keys are used only with CLI and SDK.

- **IAM Identities** as User, Groups and roles.
- **IAM Users** are End Users who log in to the console or interacts with AWS resources
- **IAM Groups** : Group of users, so they all share permission levels of the group.
- **IAM Roles:** Associate permissions to a role and assign to this to an users or groups.
- **IAM Policies** : JSON documents which grant permissions for specific users or groups or a role to access services. Policies are attached to IAM identities.

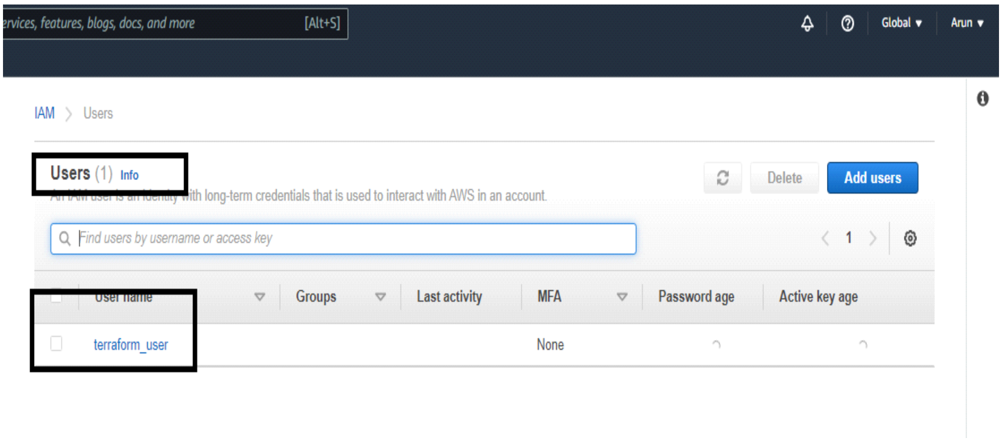
This Documentation includes the Terraform code for the following resources

- **Create IAM user**
- **Create group**
- **Add user to group**
- **Create Policies**
- **Attach Policy to user**
- **Attach Policy to group**

- **Create an IAM user**

```
resource "aws_iam_user" "user1" {
  name = "terraform_user"
 tags = {
          tag-key = "user1"
          }
}
```

In this block We used **aws_iam_user** for creation of new user and give any logical name Say for example **user1** and give name of the user you want, Here **terraform**_user has been given.


ervices, features, blogs, docs, and more
[Alt+S]
Global ▾
Arun ▾
IAM
Users
Info
Delete
Add users
Find users by username or access key
1

Users (1)

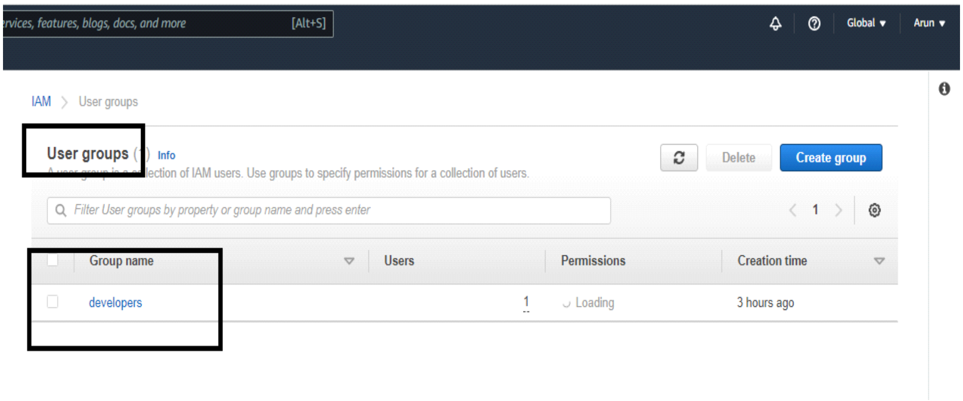
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

□	User name	Groups	Last activity	MFA	Password age	Active key age
□	terraform_user			None		

- **To Create groups**

```
resource "aws_iam_group" "developers" {
          name = "developers"
          }
```

In this block We used **aws_iam_group** for creation of new group and give any logical name, say for example **developers** and give name of the group you want, Here **developers** has been given


ervices, features, blogs, docs, and more
[Alt+S]
Global ▾
Arun ▾
IAM
User groups
Info
Delete
Create group
Filter User groups by property or group name and press enter
1

User groups (1)

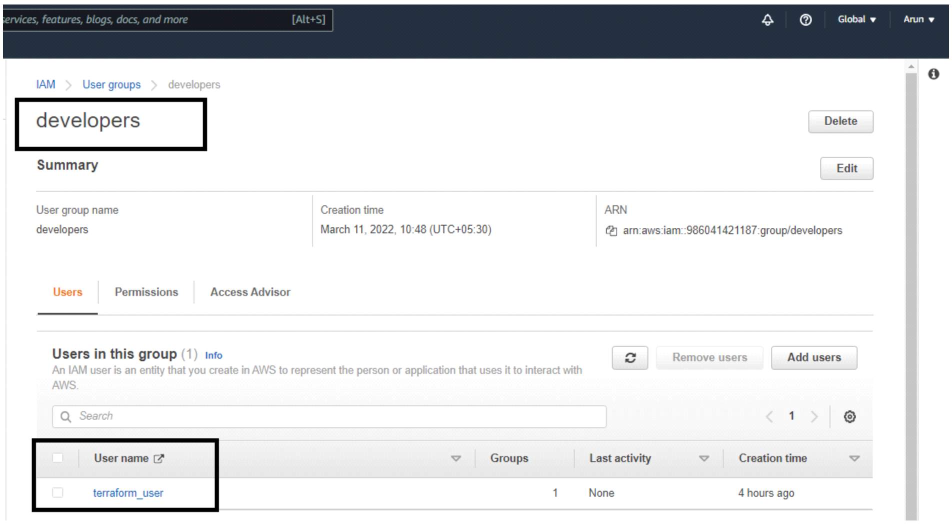
A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

□	Group name	Users	Permissions	Creation time
□	developers	1	Loading	3 hours ago

- **Add User to group**

```
resource "aws_iam_user_group_membership" "user1membership" {

        user = aws_iam_user.user1.name

        groups = [

           aws_iam_group.developers.name,

        ]

        }
```

In this block We used **aws_iam_user_group_membership** for adding a member to the group and give any logical name, say for example **user1membership** and give user name and the group ids of the groups for which user to be added.



developers

Summary

User group name: developers

Creation time: March 11, 2022, 10:48 (UTC+05:30)

ARN: arn:aws:iam::986041421187:group/developers

Users in this group (1) Info

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

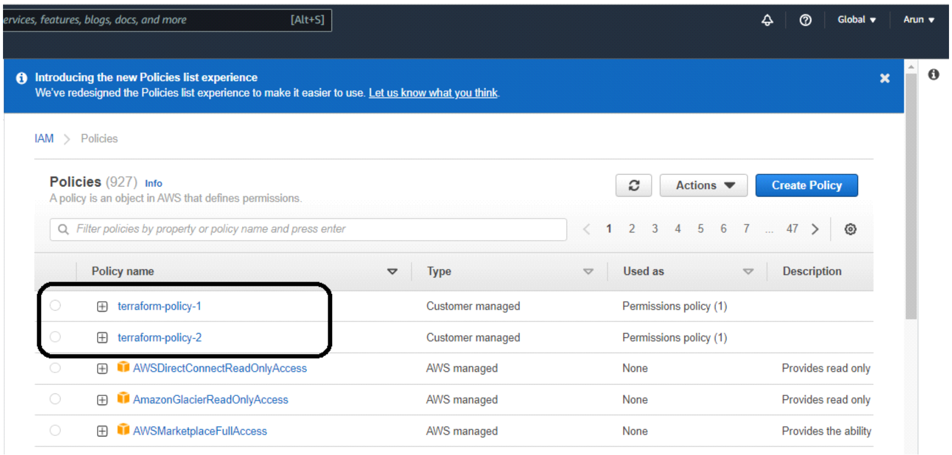
User name	Groups	Last activity	Creation time
terraform_user	1	None	4 hours ago

- **Create Policies**

```
resource "aws_iam_policy" "policy_one" {
        name = "terraform-policy-1"
        policy = jsonencode
        (
                {
                Version = "2012-10-17"
                Statement = [
                {
                        Action   = ["ec2:Describe*"]
                        Effect   = "Allow"
                        Resource = "*"
                 },
        ]
        }
```

In this block We used **aws_iam_policy** for creation of new policy and give any logical name, say for example **policy_one** and **policy_two** and give name of the policy you want, Here **terraform-policy-1** and **terraform-policy-2** have been given. And write the policy in json format.

```
            )
            }


resource "aws_iam_policy" "policy_two" {
            name = "terraform-policy-2"
            policy = jsonencode
            (
            {
             Version = "2012-10-17"
            Statement = [
            {
                        Action   = ["s3:ListBucket*"]
                        Effect   = "Allow"
                         Resource = "*"
                        }
            ]
             }
            )
            }
```



ervices, features, blogs, docs, and more [Alt+S]
Global ▾
Arun ▾

Introducing the new Policies list experience
We've redesigned the Policies list experience to make it easier to use. Let us know what you think.

IAM > Policies

Policies (927) Info
A policy is an object in AWS that defines permissions.

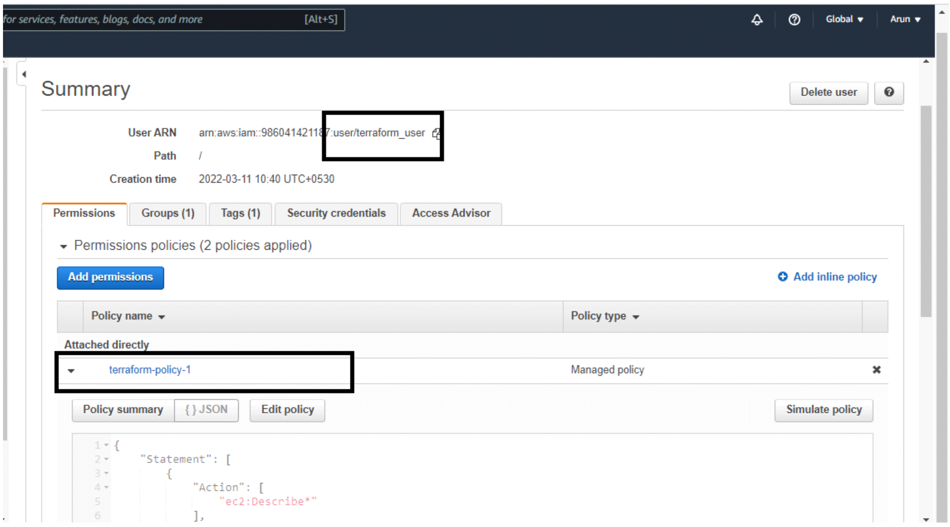
Actions ▾
Create Policy
Filter policies by property or policy name and press enter
< 1 2 3 4 5 6 7 ... 47 >

	Policy name	Type	Used as	Description
○	⊞ terraform-policy-1	Customer managed	Permissions policy (1)	
○	⊞ terraform-policy-2	Customer managed	Permissions policy (1)	
○	⊞ AWSDirectConnectReadOnlyAccess	AWS managed	None	Provides read only
○	⊞ AmazonGlacierReadOnlyAccess	AWS managed	None	Provides read only
○	⊞ AWSMarketplaceFullAccess	AWS managed	None	Provides the ability

- **Attach Policy to user**

```
resource "aws_iam_user_policy_attachment" "ec2-attach" {

            user       = aws_iam_user.user1.name

            policy_arn = aws_iam_policy.policy_one.arn

            }
```

In this block We used **aws_iam_user_policy_attachement** for attaching policy to the **user** and give any logical name, say for example **ec2-attach** and give **id of the user** you want, and **policy_arn id** have been given.



Summary

User ARN arn:aws:iam::986041421187:user/terraform_user

Path /

Creation time 2022-03-11 10:40 UTC+0530

Permissions policies (2 policies applied)

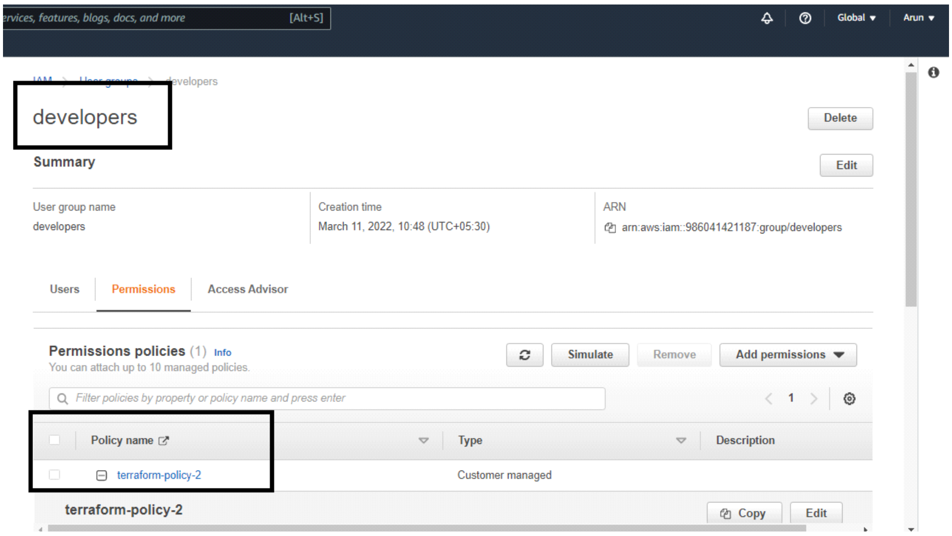
Policy name	Policy type
Attached directly	
terraform-policy-1	Managed policy

```
{
    "Statement": [
        {
            "Action": [
                "ec2:Describe*"
            ],
```

- **Attach Policy to group**

In this block We used **aws_iam_group_policy_attachement** for attaching policy to the **group** and give any logical name, say for example **s3-attach** and give **id of the group** you want, and **policy_arn id** have been given.

```
resource "aws_iam_group_policy_attachment" "S3-attach" {
    group      = aws_iam_group.developers.name
    policy_arn = aws_iam_policy.policy_two.arn
}
```



ervices, features, blogs, docs, and more [Alt+S]
Global
Arun
IAM > User groups > developers
Delete
Edit

developers

Summary

User group name
developers

Creation time
March 11, 2022, 10:48 (UTC+05:30)

ARN
arn:aws:iam::986041421187:group/developers

Users
Permissions
Access Advisor

Permissions policies (1) Info

You can attach up to 10 managed policies.

Simulate
Remove
Add permissions
Filter policies by property or policy name and press enter
1

Policy name	Type	Description
terraform-policy-2	Customer managed	

terraform-policy-2