Identity Access Management (IAM)

Learn all about AWS IAM and what it entails. IAM is at the very center of AWS it is something that you absolutely need to understand as you build your applications on AWS.

IAM

Allows you to manage users and their levels of access to the AWS resources. Here are some key points to remember when thinking about AIM:-

1. IAM is universal, it is not specific to a region or AZ
2. Centralized control to the AWS account
3. Shared Access to your AWS account
4. Granular Permissions
5. Identity federation (like Active Directory)
6. MFA – Multifactor Authentication – 2 factor Auth
7. Temporary access for users
8. Allows you to set up your own password rotation policy
9. Integrates with many different AWS services supports PCI DSS compliance

In IAM there are 4 types of entities :-

1. **Users** – End-users
2. **Groups** – A collection of users under one set of permissions.

Defined by the role trust policy

Defined by IAM permissions policies

3. **Roles**, you can create roles and can then assign them to AWS resources
4. **Policy** is a document that defines one or more permissions.

You can apply **policies** to users, groups, and roles. Users, groups, and roles can all share the same policy documents.

Below is an example of a JSON policy document.

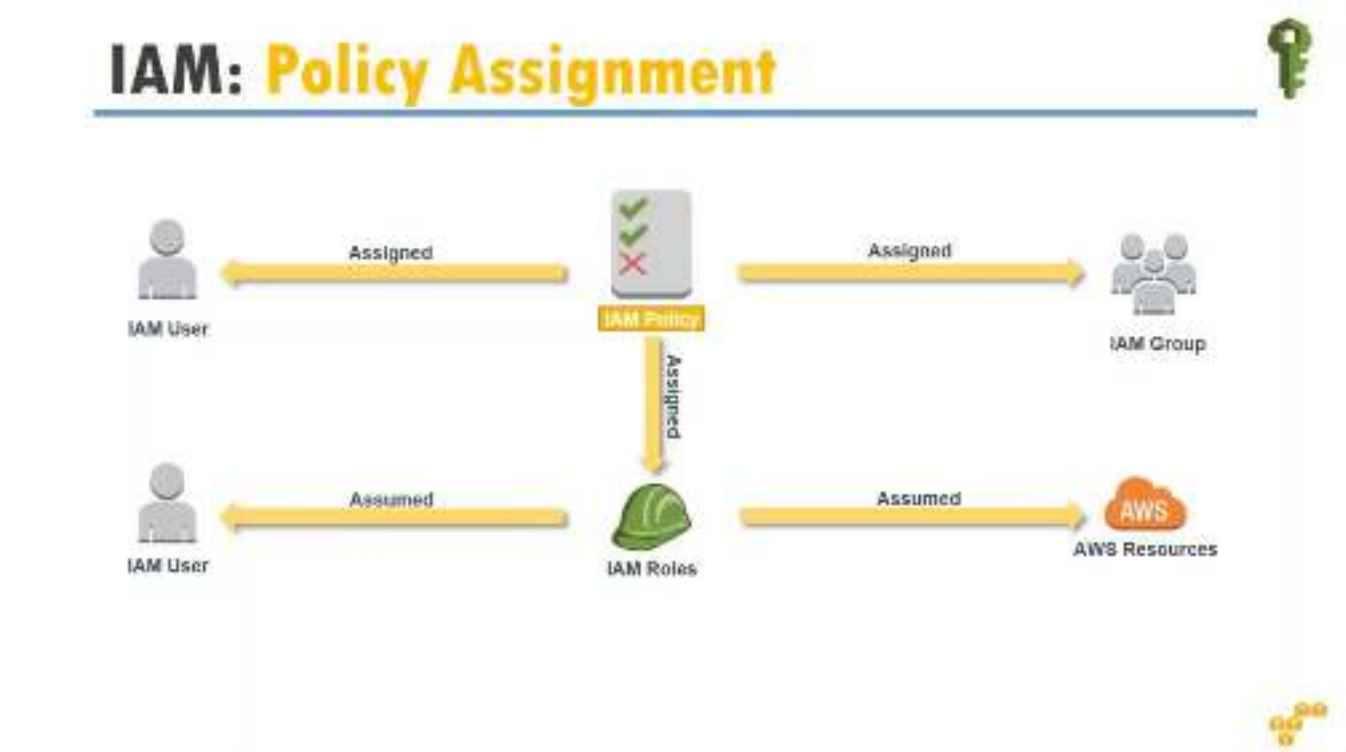
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aws-portal:ViewBilling",
                "aws-portal:ViewPaymentMethods",
                "aws-portal:ModifyPaymentMethods",
                "aws-portal:ViewAccount",
                "aws-portal:ModifyAccount",
                "aws-portal:ViewUsage"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": "203.0.113.0/24"
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "arn:aws:s3:::customer",
                "arn:aws:s3:::customer/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:GetConsoleScreenshots"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "codedploy:*",
```

```
                "codecommit:*"
            ],
            "Resource": [

                "arn:aws:codedeploy:us-west-2:123456789012:deploymentgroup:*",
                "arn:aws:codebuild:us-east-1:123456789012:project/my-demo-project"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:GetObject",
                "s3:DeletObject",
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::developer_bucket",
                "arn:aws:s3:::developer_bucket/*",
                "arn:aws:autoscling:us-east-2:123456789012:autoscalgrp"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": [
                        "public-read"
                    ],
                    "s3:prefix": [
                        "custom",
                        "other"
                    ]
                }
            }
        }
    ]
}
```

“A root account” is the account created when one first sets up the AWS account. It has complete Admin Access.

1. New Users have no permissions when first created.
2. New Users are assigned access **key** and a secret access **ID** when & first created.
3. Access ID & Secret Access keys are used for access via the API and CLI.



Key Points to Note :-

1. An AIM Policy can be assigned to an **User** or a **Group** or a **Role**
2. An AIM is assumed by an **User** or an AWS Resource