# AMAZON IDENTITY AND ACCESS MANAGEMENT

*By VIGNESH*

## Why IAM?

Cloud security is the highest priority in AWS. When you host your environment in the cloud, you can be assured that it’s hosted in a data center or in a network architecture that’s built to meet the requirements of the most security-sensitive organization.

## What is IAM?

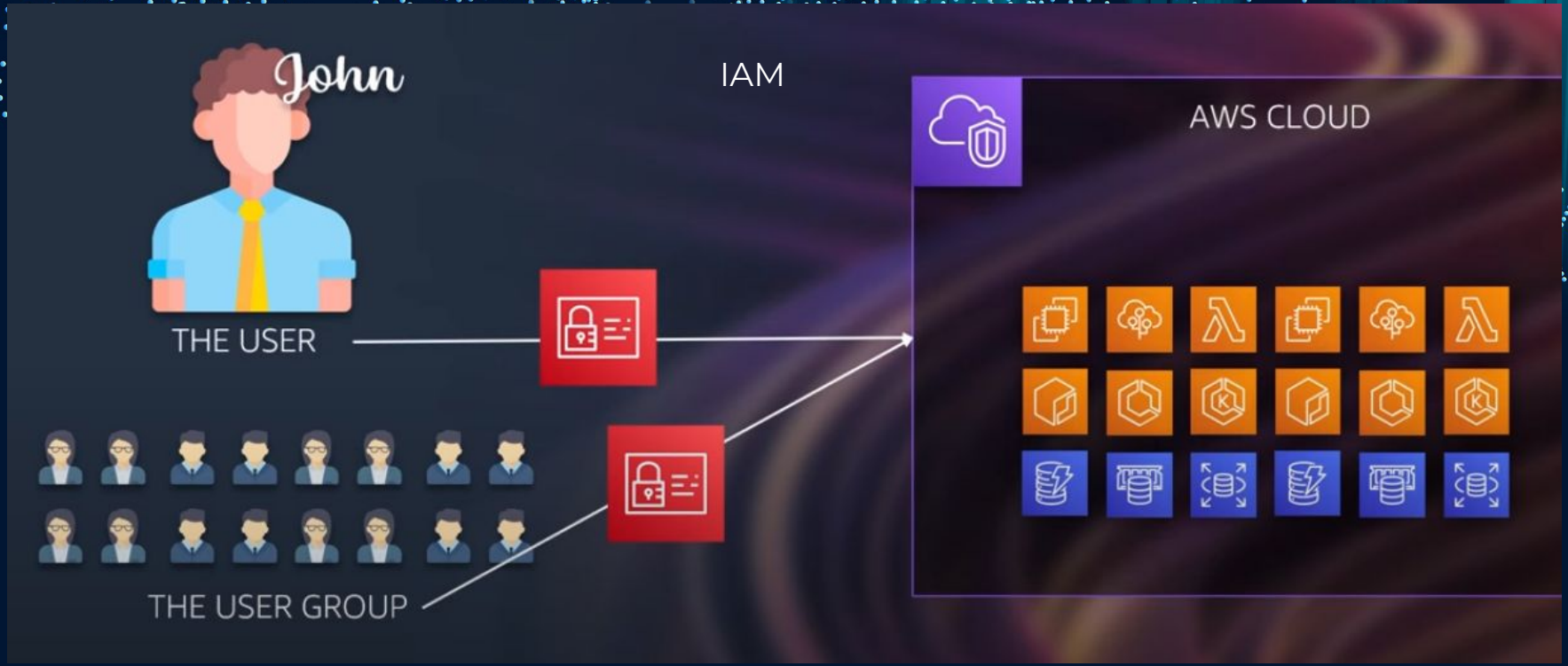
AWS Identity and Access Management (IAM) is a web service for securely controlling access to AWS resources. It enables you to create and control services for user authentication or limit access to a certain set of people who use your AWS resources.

Fine Grained Access Control across all AWS services.

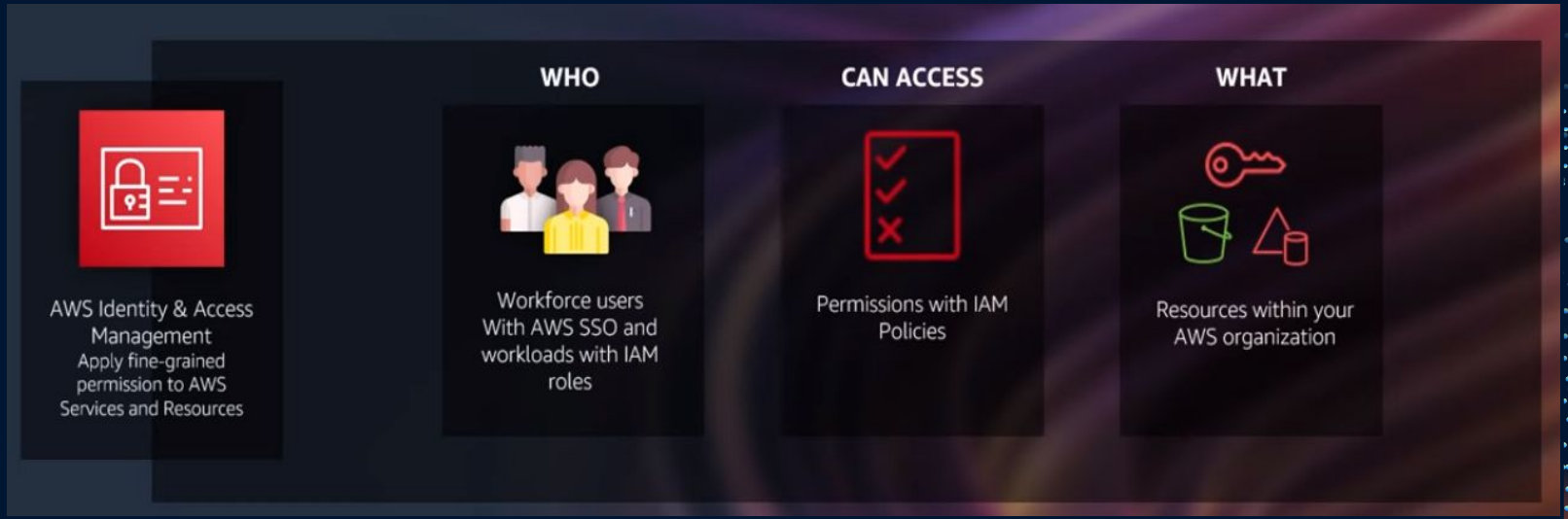
You manage permissions to your Workforce.

Who can access which service under which condition.

IAM is offered at no additional cost

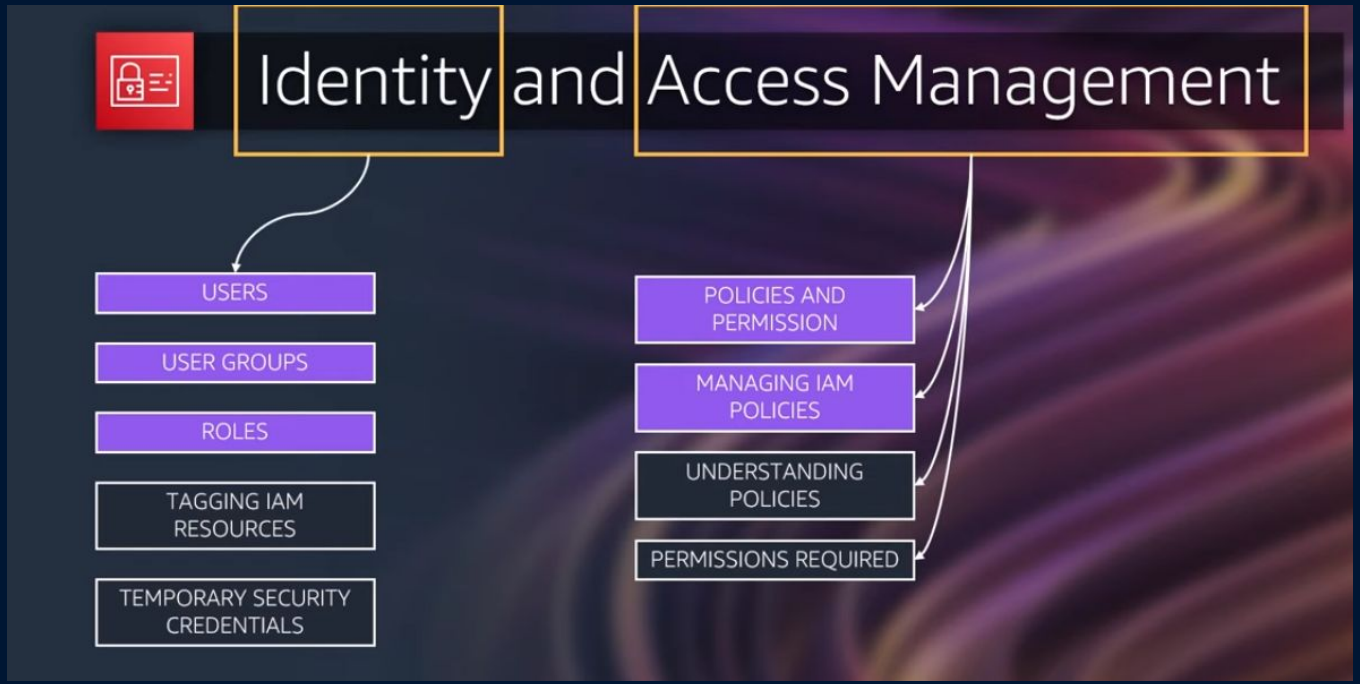

John
IAM
AWS CLOUD
THE USER
THE USER GROUP

# WHO CAN ACCESS WHAT !



# Access control is a selective restriction of access to a place or other resource.

# IAM


Identity and Access Management
USERS
USER GROUPS
ROLES
TAGGING IAM RESOURCES
TEMPORARY SECURITY CREDENTIALS
POLICIES AND PERMISSION
MANAGING IAM POLICIES
UNDERSTANDING POLICIES
PERMISSIONS REQUIRED

**USER:**

**An IAM user is a fundamental entity within IAM. It represents an individual or application that interacts with AWS services. IAM users are separate from your AWS account root user, and they provide a way to grant specific permissions to different individuals or processes without sharing the main account credentials.**

**AM users can also be provided access to the AWS Management Console, allowing them to interact with AWS services through a web-based user interface.**

## GROUP

(IAM) User Groups are a way to simplify the management of permissions for multiple users within your AWS account. Instead of individually assigning permissions to each user, you can create groups and attach policies to those groups. This makes it easier to manage and maintain consistent access control across your AWS resources.

If you need to change permissions for a set of users, you can modify the permissions attached to the group's policies

# ROLE

IAM Role provide a Temporary permissions. Roles promote the principle of least privilege by allowing you to assign only the necessary permissions to a resource or user.

IAM Roles allow you to define a set of permissions that can be assumed by AWS resources. Instead of attaching permissions directly to a user or resource, you define a role with specific permissions and then assign that role to entities like AWS EC2 instances, Lambda functions, or even other roles.

## POLICY

A JSON document that defines permissions and access control rules within an AWS account.

Users: To grant specific permissions to individual AWS users.

Groups: To assign a set of permissions to a collection of users.

Roles: To define permissions for entities (like EC2 instances or Lambda functions) that assume the role temporarily.

Resource-based policies: To control access to AWS resources, like S3 buckets, from external accounts or services.

## Identity Policy:

An identity policy, also known as an AWS Identity and Access Management (IAM) policy, is used to define permissions for individual AWS IAM users, groups, or roles.

For instance, you can use an identity policy to grant a specific IAM user permission to read objects from an S3 bucket

## Resource Policy:

A resource policy, on the other hand, is used to control access to AWS resources, such as S3 buckets, Lambda functions, or Amazon API Gateway APIs.

For example, you can use a resource policy on an S3 bucket to allow public read access to all objects in the bucket, even if the requester is not authenticated with your AWS account.

**Statement:**

**Effect: This specifies whether the permissions in the statement are granted ("Allow") or denied ("Deny").**

**Action: This lists the AWS service actions that the policy allows or denies. Actions can represent specific API operations, such as s3:GetObject or ec2:RunInstances.**

**Resource: This specifies the AWS resources (e.g., S3 buckets, EC2 instances) to which the actions apply. The resource can be specified using Amazon Resource Names (ARNs).**

Creating a Role to access an S3 bucket

Modifying IAM role for that ec2-instance

SSH into the EC2 instance

Aws s3 ls

Aws s3 cp <sample.txt> s3://<Bucket Name>/<object name> (U)

aws s3 cp s3://bucket-name/object-key . (D)

# Giving a user S3 full access (Identity Policy) and Modifying Resource policy for a specific object.

## Resource policy

```
{
	"Version": "2012-10-17",
	"Statement": [
		{
			"Sid": "Statement1",
			"Effect": "Deny",
			"Principal": {
				"AWS": "arn:aws:iam::YOUR_ACCOUNT_ID:user/YOUR_ACCOUNT_USER_NAME"
			},
			"Action": "S3:*",
			"Resource": [
				"arn:aws:s3:::YOUR_S3_BUCKET_NAME",
				"arn:aws:s3:::YOUR_S3_BUCKET_NAME/*"
			]
		}
	]
}
```