# CHAP-1 IAM (Identity and Access Management)

**Identity and Access Management (IAM)** is a web service that helps you securely control access to AWS resources. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

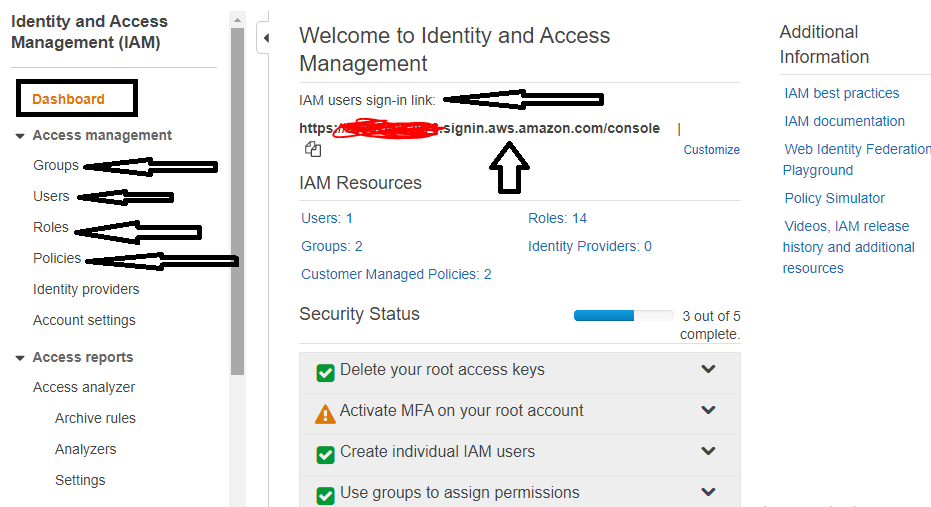
IAM is a global service. IAM is free of cost.

Any AWS customer can use IAM. The service is offered at no additional charge. You will be charged only for the use of other AWS services by your users.

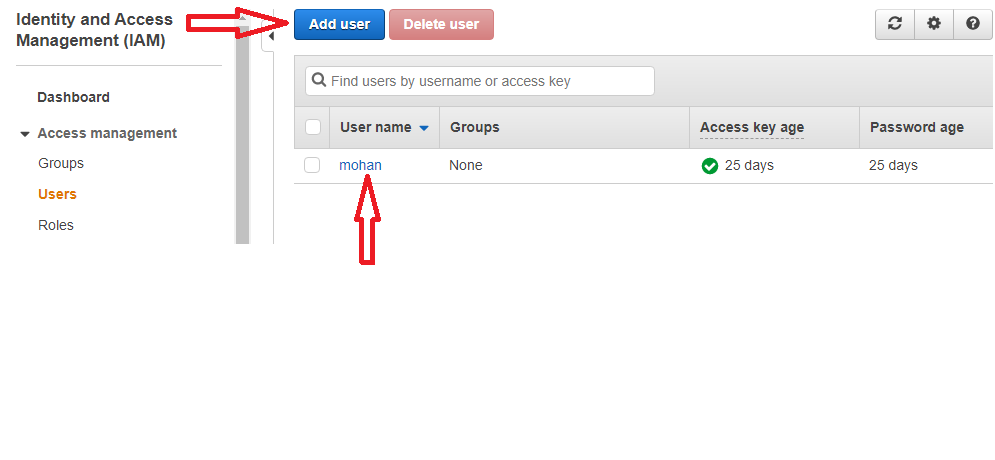
There are 3 basic components in IAM. They are-

1. **User**- A user can be an individual, system, or application requiring access to AWS services. A user can place requests to web services such as Amazon S3 and Amazon EC2. A user's ability to access web service APIs is under the control and responsibility of the AWS account under which it is defined.
2. **Group**- A group is a collection of IAM users.
3. **Roles**

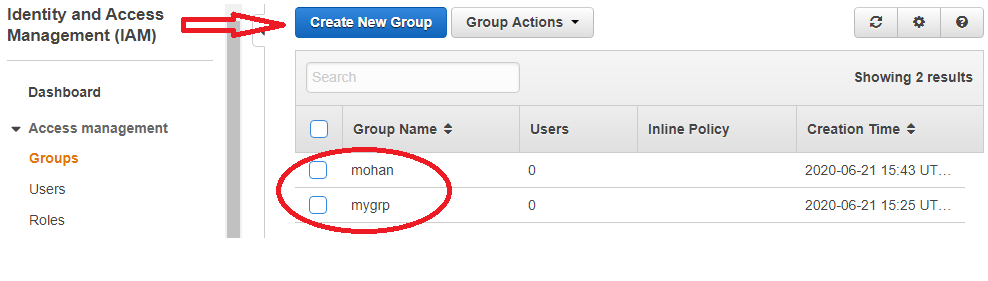
The below diagram shows IAM dashboard. The components of IAM have been shown with arrows. The IAM sign-in link is also marked with arrow. This link will be used by IAM users to login to AWS management console.



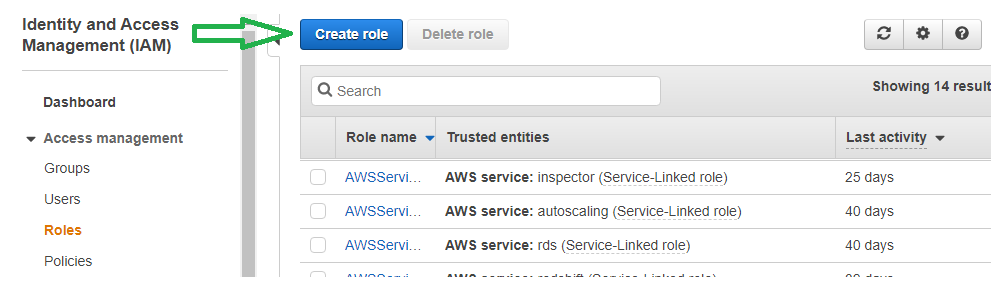
The user page inside IAM is shown as below. You can click “Add User” button to add new users. You can also view existing user in this section.



The group page inside IAM is shown as below. Here you can create new group and view existing groups.



The role page inside IAM is shown as below. Here you can create roles and can also view roles created by AWS for multiple purposes.



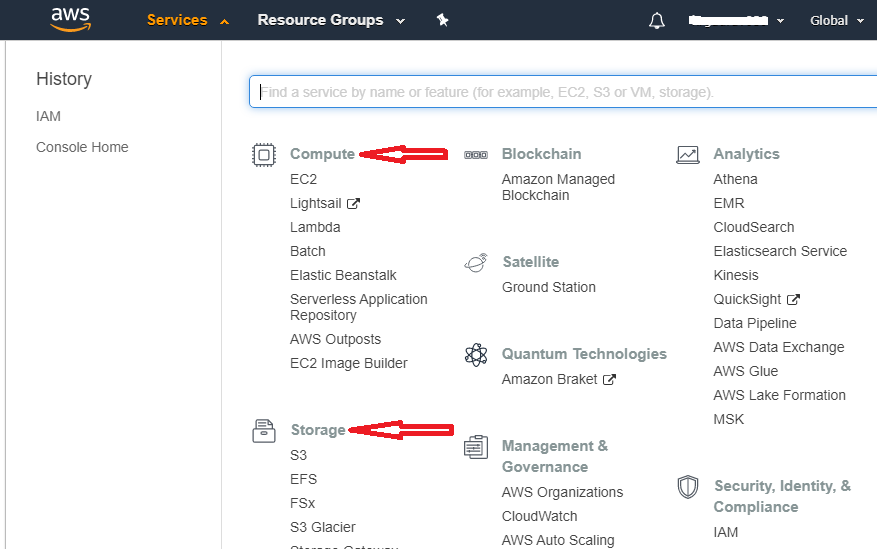
**Root Account**

The AWS account holder (also called Root Account) can manage users, groups, security credentials, and permissions. Root account is the one who has created AWS account and who will be paying all the bills.

**Interaction with AWS**

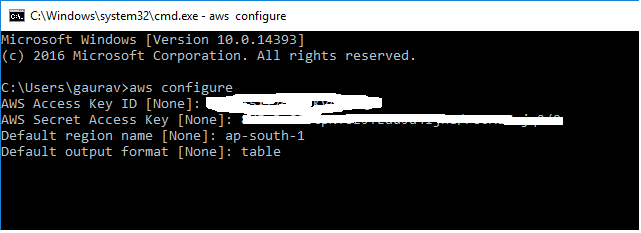
You can work with AWS Identity and Access Management in any of the following ways.

1. **AWS Management Console-** A GUI based interface from where you can manage your AWS infrastructure. It is shown as below.



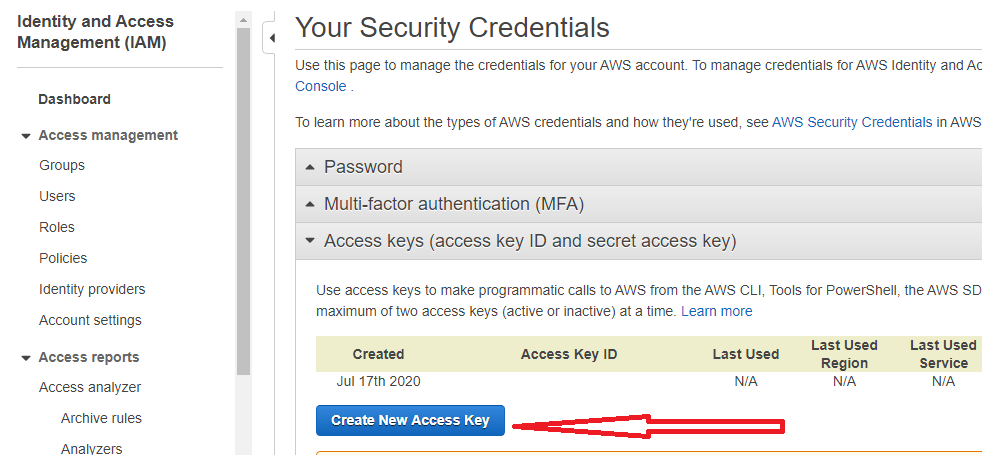
1. **AWS Command Line Tools-** the AWS Command Line Interface (AWS CLI) is an open source tool that enables you to interact with AWS services using commands in your command-line shell. You can download AWS CLI from Internet and then install on your computer.

After installation of CLI, go to Command Prompt and type “aws configure”. You will have to pass below details there to configure your AWS account.

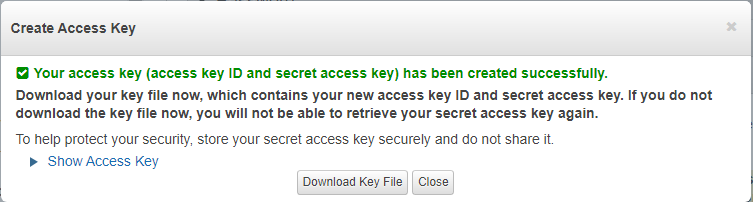
****

**How to get Access Key and Secret Access Key?**

Go to “Your Security Credentials” page, then on “Access keys (access key ID and secret access key)”, there you will find “Create New Access Key” button.

****

This is the next window from where you can get your access key (access key ID and secret access key).

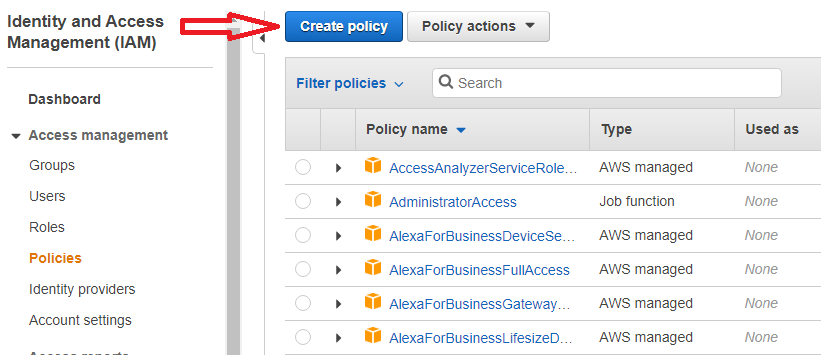


**IAM role**- An IAM role is an IAM entity that defines a set of permissions for making AWS service requests. IAM roles are not associated with a specific user or group. Instead, trusted entities assume roles, such as IAM users, applications, or AWS services such as EC2.

Roles are needed when one service wants to access another service. For example- EC2 accessing S3 bucket to write a text file inside bucket.

**Policies-** Set of rules governing permission boundary in AWS. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when an IAM principal (user or role) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents.

The policy page inside IAM is shown below. Here you can create policies and view existing policies.



There are two types of policies. They are-

1. **Inline policies** – Policies that you create and manage and that are embedded directly into a single user, group, or role. In most cases, we don't recommend using inline policies.
2. **Managed policies** –policies that you can attach to multiple users, groups, and roles in your AWS account. There are two types of managed policies:
3. **AWS managed policies** – Managed policies that are created and managed by AWS.
4. **Customer managed policies** – Managed policies that you create and manage in your AWS account. Customer managed policies provide more precise control over your policies than AWS managed policies.

AWS managed policies are the recommended policies to be attached with a principal (user/role).

**What does a policy look like?**

The following policy grants access to add, update, and delete objects from a specific folder, *folder1234*, in a specific bucket, *bucket123*.

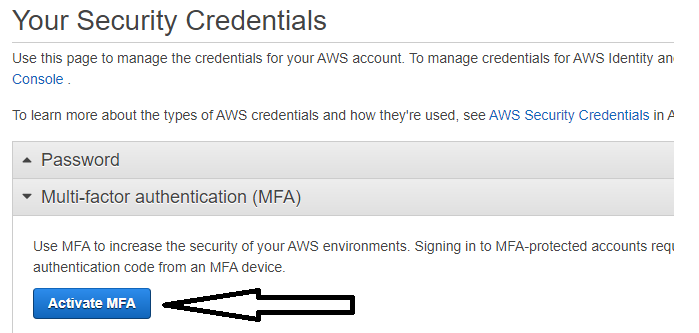
{  
  "Version":"2012-10-17",  
  "Statement":[  
   {  
     "Effect":"Allow",  
     "Action":[  
       "s3:PutObject",  
       "s3:GetObject",  
       "s3:GetObjectVersion",  
       "s3:DeleteObject",  
       "s3:DeleteObjectVersion"  
     ],

     "Resource":"arn:aws:s3::: bucket123/ folder1234/\*"  
    }  
  ]  
}

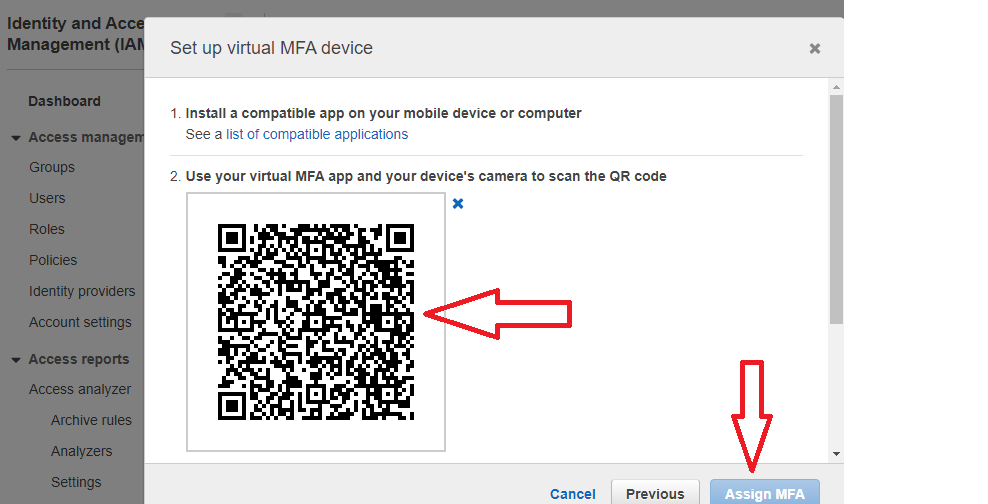
**MFA in AWS**

Multi-Factor Authentication (MFA) is a practice that adds an extra layer of protection on top of your user name and password. With MFA enabled, when a user signs in to an AWS Management Console, they will be prompted for their user name and password (the first factor—what they know), as well as for an authentication code from their AWS MFA device (the second factor—what they have). Taken together, these multiple factors provide increased security for your AWS account settings and resources.

You can activate MFA on your account by clicking on “Activate MFA” as shown below.



Then you will land into this page where you can configure MFA device. You can use Google Authenticator in your mobile as virtual MFA device.



**Federated Access in AWS**

IAM supports identity federation for delegated access to the AWS Management Console. With identity federation, external identities are granted secure access to resources in your AWS account without having to create IAM users.

These external identities can come from your corporate identity provider (such as Microsoft Active Directory or from the AWS Directory Service) or from a web identity provider (such as Amazon Cognito, Login with Amazon, Facebook, Google, or any OpenID Connect-compatible provider).