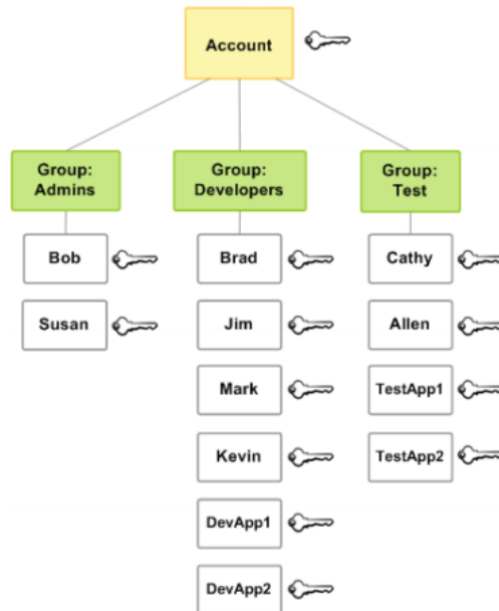


# AWS Identity and Access Management (IAM)

## What is AWS IAM ?

IAM is a web service which is used for securely controlling and managing access to all the AWS services. With IAM you can centrally manage users, security credentials such as Access and Secret Keys and permissions that control which users or services can access what AWS services.



## Difference Between Root User and Admin/IAM User :

Mostly both of them are the same access, however Admin/IAM user has no access over Root while Root has full access over Admin. Generally Root has full access to do anything on an account where a Admin/IAM user is limited with the defined access.

## IAM Users :

An IAM user that we create in AWS is basically represent a person or an application that uses to interact with various services within AWS. An user in AWS basically consists of a name and credentials.

- **Users and Credentials :**

Depending on the User credentials you can access AWS in different ways depend on the need.

- **Console Password :**

- With an User Name and Console Password you can sign into AWS Management Console for an interactive session.

- **Access Keys:**

- These are combination of Access Key ID and Secret Access Key which can be assigned to a user to access programmatically AWS services using API code, AWS CLI or PowerShell tools.
- **SSH Keys for CodeCommit :**
  - This is a SSH Public key in the OpenSSH format which can be used to authenticate with CodeCommit.
- **Server Certificates :**
  - SSL/TLS certificates can be used to authenticate with some of the AWS services, the recommended one is AWS ACM (Certificate Manager) to provision, deploy and manage the server certificates.
- **Users and Permissions :**

When a new user get created under AWS IAM, by default user has no permission to do anything and even not authorized to perform any operation or access to any of the AWS services. If you want to make the user authorized for any of the services you have add the policies accordingly.

### **IAM Group :**

IAM Group is basically a collection of IAM Users. Group make easier to manage permissions for a large number of users by grouping them and specifying the required type of permissions. Just by adding the users to the groups permissions will be automatically inherited from the groups.

- **Some Important Characteristics of Group :**
  - A group can contain many users and an user can belongs to multiple groups.
  - Groups can't be nested, A group can contain only users, but not the groups.
  - Groups are not automatically includes all users in the AWS account, each users need to assign to the group.
  - As per the default limit you can create 300 groups and a group can contain 5000 users. There are some more limitations available, please follow the link for more details. [https://docs.aws.amazon.com/IAM/latest/UserGuide/reference\\_iam-limits.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_iam-limits.html)

### **IAM Roles :**

IAM role is an IAM identity which is similar to an IAM User and it contains some specific permissions which determine what the identity can and cannot do in AWS as per the attached policies. A role doesn't have a credential such as Password or access keys associated with it, instead it provides a temporary security credentials for the Role session.

We can use Roles to delegate access to Users, Applications or Services to access the AWS services. Some of the common use cases are below.

- Roles can be used to grant access to an User to access the AWS services.
- It can be used to grant access to an user of an AWS Account to access service from another AWS account.
- You can also grant access to the AWS account for the user who already have identities defined outside of AWS, such as Corporate AD Directory.

- Also it helps to grant access to some third party during security Audit.
- It also help to avoid hardcoding the Access Keys and Secret keys on a server while the server is deployed in AWS itself.

### **AWS Policies :**

Policies are used to manage access in AWS by attaching them to IAM identities like (Users, Groups and Roles) and AWS Services. Permissions in the policies determine whether the request is allowed or denied.

### **AWS Supports 6-Types of Policies :**

- **Identity-Based Policy :**
  - You can attach Managed and Inline policies to the IAM identities (Users, Groups and Roles) which grants permissions to an identity.
- **Resource-Based Policy :**
  - It grants permissions to the principal that is specified in the policy. Principals can be in the same account as the resource or in other accounts. You can attach inline policies to the resources such as S3 bucket policy and IAM Role policy.
- **Permissions Boundaries :**
  - Managed policies can be used as the permission boundaries for an IAM entity (User or Role). It defines maximum permissions that the identity-based policy can grant to an entity, but doesn't grant permissions.
- **Organizations SCPs :**
  - SCPs are used to define the maximum permissions for account members of an Organization or Organizational Unit (OU). SCP do not grant permissions.
- **Access Control Lists (ACLs) :**
  - ACLs are cross account permissions policies that grant permissions to the specified principal. ACLs cannot grant permissions to entities within the same account. ACLs used to control which principals in other accounts can access the resources to which the ACL is attached.
- **Session Policies :**
  - Session policies limit the permissions that the role or user's identity-based policies grant to the session. Session policies limit permissions for a created session, but do not grant permissions.

### **Account Settings :**

#### **How to Set Password Policy :**

- Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>
- In the navigation pane, click **Account Settings**.
- In the **Password Policy** section, select the options you want to apply to your password policy.
- Click **Apply Password Policy**.

### **Password Policy :**

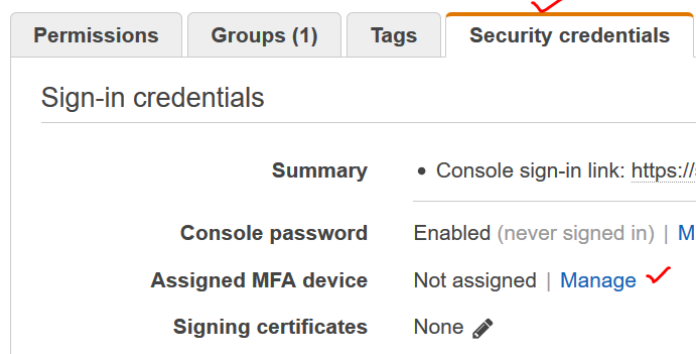
- You can set a password policy on your AWS account to specify complexity requirements and mandatory rotation periods for your IAM users' passwords.
- Following list of options are available which will help you to configure a strong Password Policy and you can modify them as per your requirement.
  - Minimum password length : **(6 to 128)**
  - Require at least one uppercase letter: **(A to Z)**
  - Require at least one lowercase letter : **(a to z)**
  - Require at least one number : **(0 to 9)**
  - Require at least one nonalphanumeric character : **( ! @ # \$ % ^ & \* ( ) \_ + - = [ ] { } | ' )**
  - Allow users to change their own password : **(Users can be permitted to change their password)**
  - Enable password expiration : **(expiration period between 1 and 1095 days, inclusive)**
  - Prevent password reuse : **(You can set the number of previous passwords from 1 to 24, inclusive)**
  - Password expiration requires administrator reset : **(You can prevent IAM users from choosing a new password after their current password has expired)**

#### IAM Multi-Factor Authentication (MFA) :

- AWS MFA is the simple and best practice to add an extra layer of protection to the AWS account on top of User Name and Password. There is no additional charge for enabling and using MFA. The user has to obtain a supported Hardware or Virtual MFA device. Please follow the below links to know how to enable the MFA <https://splabs.in/enable-virtual-mfa-device-for-an-aws-iam-user/> and <https://splabs.in/enable-virtual-mfa-device-for-aws-account-root-user/>

#### How to Enable Virtual MFA Device for an AWS IAM User :

- Sign in to the AWS Management Console and Select IAM service at <https://console.aws.amazon.com/iam/>
- From the left navigation pane select **Users**.
- From the **User Name** list select the intended IAM user.
- Choose the Security Credential tab. Next to **Assigned MFA Device**, choose **Manage**.



- In the **Manage MFA Device** wizard, choose **Virtual MFA Device**, and then choose **Continue**.

Manage MFA device

Choose the type of MFA device to assign:

☒ **Virtual MFA device**  
Authenticator app installed on your mobile device or computer

☐ **U2F security key**  
YubiKey or any other compliant U2F device

☐ **Other hardware MFA device**  
Gemalto token


For more information about supported MFA devices, see [AWS Multi-Factor Authentication](#)

Cancel
Continue

- IAM generates and displays configuration information for the virtual MFA device including the QR Code Graphic.

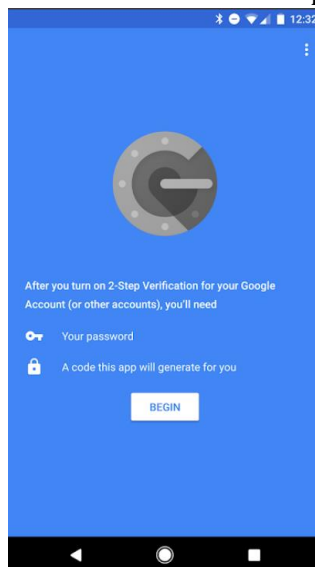
Set up virtual MFA device

1. Install a compatible app on your mobile device or computer  
See a [list of compatible applications](#)
2. Use your virtual MFA app and your device's camera to scan the QR code



Cancel
Previous
Assign MFA

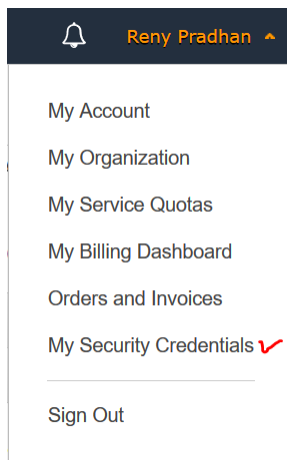
- Install and Open your virtual MFA app (For Example **Google Authenticator**) on your Mobile and choose the option to create a new virtual MFA device or account.



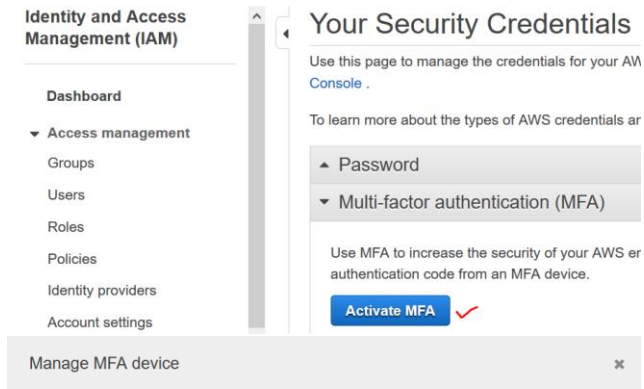
- If the MFA app supports QR codes then From the wizard, choose **Show QR Code**, and then use the app to scan the QR code.
- If the MFA device doesn't support QR Codes then from the **Manage MFA Device** wizard, choose **Show secret key**, and then type the secret key into your MFA app.
- Once the above steps performed correctly MFA Device start generating the One-Time Password.
- In the **Manage MFA Device** wizard, in the **MFA code 1** box, type the one-time password that currently appears in the virtual MFA device. Wait up to 30 seconds for the device to generate a new one-time password. Then type the second one-time password into the **MFA code 2** box. Choose **Assign MFA**.

#### How to Enable Virtual MFA Device for AWS Account Root User :

- Sign in to the AWS Management Console.
- On the right side of the navigation bar, choose your account name, and choose **My Security Credentials**. If necessary, choose **Continue to Security Credentials**. Then expand the **Multi-Factor Authentication (MFA)** section on the page.



- Choose **Activate MFA**.
- From the wizard, choose **Virtual MFA device**, and then choose **Continue**.



Choose the type of MFA device to assign:

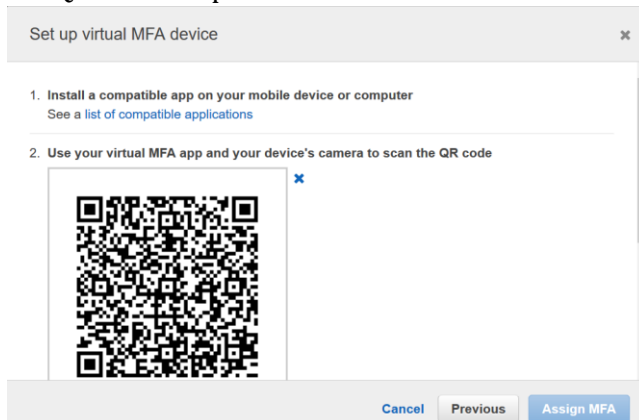
☒ **Virtual MFA device**  
Authenticator app installed on your mobile device or computer

☐ **U2F security key**  
YubiKey or any other compliant U2F device

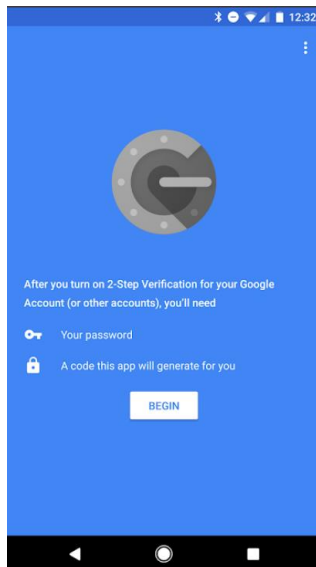
☐ **Other hardware MFA device**  
Gemalto token

For more information about supported MFA devices, see [AWS Multi-Factor Authentication](#)

- IAM generates and displays configuration information for the virtual MFA device including the QR Code Graphic.



- Install and Open your virtual MFA app (For Example **Google Authenticator**) on your Mobile and choose the option to create a new virtual MFA device or account.



- If the MFA app supports QR codes then From the wizard, choose **Show QR Code**, and then use the app to scan the QR code.
- If the MFA device doesn't support QR Codes then from the **Manage MFA Device** wizard, choose **Show secret key**, and then type the secret key into your MFA app.
- Once the above steps performed correctly MFA Device start generating the One-Time Password.
- In the **Manage MFA Device** wizard, in the **MFA code 1** box, type the one-time password that currently appears in the virtual MFA device. Wait up to 30 seconds for the device to generate a new one-time password. Then type the second one-time password into the **MFA code 2** box. Choose **Assign MFA**.
- Choose **Assign MFA**, and then choose **Finish**.