**IAM**

**What is IAM…?**

* IAM stands for Identity and access management.
* IAM is web service which helps you to securely control access to AWS resources. With IAM you can centrally manage permissions that which AWS resources users can access.
* With IAM you can specify that who is authenticated and authorized to access the AWS resources.

**Features of IAM:**

* **Shared access to your AWS account:** With IAM you grant permission to the other people to access the AWS resources without sharing your username and password.
* **Granular Permission:** With IAM you can grant different permissions to the different people.
* **MFA:** With MFA you can provide the extra security to root users and IAM user account.
* **Identity Federation:** You can allow users to access who having the password else where for a limited amount of time.
* **Identity information for assurance:** You receive log records that include information who had accessed the account and who sent the resources request from your account.
* **Integration with AWS services:** You can integrate with the other AWS services.
* **Free to Use:** IAM and IAM TTS both can be used with out extra charges.

**Users:**

* IAM user is an entity created using the IAM. Each IAM user can represent the human who can interact with the AWS resources.
* An IAM user with administrator powers is not equal to the root user.
* An IAM user can login into there user account by entering there username, A/C number of root user and password which has been created.
* Root user can disable the password of the IAM user it means you can no longer login into the IAM user account but it will not effect your IAM user account permission.
* You can change the password of the IAM user account by using the old password.
* You can enable the MFA for the more security. Per each account you have 12,500 IAM user accounts.
* Using AWS IAM user you can find the unused passwords and access keys by that you can remove that IAM user account which has no longer use. You can also download and generate a credential reports of all the IAM users who has listed.
* For newly created AWS IAM user account does not have any permissions to access anything. Having individual IAM user accounts you can individually assign the permissions to each IAM user.
* For having the AWS IAM user in your AWS account you don’t need to pay any extra payment.

**User Groups:**

* User Groups is a collection of the Users. It is hard to assign to each user its permission from get ridding of that you can create the user groups.
* If you specify the permissions to the user groups then you add the permissions to the users to that group the specific user can get all the permissions which the group has.
* For user groups you can attach policies and detach the policies. You can add and remove the users from the user groups. You can rename and delete the User groups.
* **Characteristics of the User Groups:**
* Group can contain many users, a user can belongs to many groups.
* Groups cannot be nested, A user group cannot contain another user group in it.
* There will be no default user group is created.
* The User groups and users in the user group are limited.

**Roles:**

* An IAM role is an IAM identity that you create in your AWS account that has special permission.
* IAM role is similar to IAM user it can also specify the users what to do and what not to do. Usually IAM user is uniquely identified by an unique password and username.
* Like IAM user, Roles does not have any long term credentials. Roles provides you a temporary credential for your role session.

**Policies:**

* Policy is an object which is attached to an identity for specifying the specific permissions. Permission in the policies defines we are allowed or denied to that AWS resources.
* Most of the AWS policies are written in the JSON document. There are six types of AWS policies:
  + - **Identity based policy :** It is an policy which is used to control the permission of the identity ( users, user groups, roles). It was written in the JSON document. Identity based policies are two types:
* **Managed Policies:** These are the policies used to attach to the multiple users, groups, roles. Managed Policies are two types:
* **AWS Managed Policies:** These are the policies managed by the AWS itself.
* **Customer Managed Policies:** These are created and managed by the user itself.
* **Inline-polices:** These policies are attached to single user, user groups. If you want to delete the inline-police then you need to delete identity which it has been attached.
* **Resource Based Policy:** Resource Based Policy are policy document that are attached to a resource. Resource Based Policy are the inline policy. IAM supports only one resource based policy which is called as trust policy which is attached to IAM role. An IAM role is an resource and identity which supports the resource based policy.
* **Permission boundaries:** Permission boundaries is an advanced feature in which the set maximum permission that can be granted by the identity based policy. When you set permission boundaries for an entity, that entity needs the permissions from both identity based policy and permission boundaries.
* **Service control policies:** If you enable all the features in the organization, then you will able to enable the service control policies. SCP are the JSON policies that specify the maximum number of permissions. SCP limits the permissions for both user accounts and root user account.
* **Access control lists:** ACLs are the service policies that allows you to control which principles in another AWS account. ACL will not allows you to control access in your account. ACL are not JSON document polices even though they are similar to resource based policies.
* **Session policies:** Session policies are the AWS document that allows you to control the actions which are performed by the user, user groups, roles while using Resource key management services (RKMS). AWS SDK and clients provides an easy way to programmatically interact with the RKMS.

**MFA:**

* MFA stands for Multi Factor Authentication. MFA provides extra layer security for your AWS root user and AWS IAM user.
* When you enable the MFA for your root user it will not enable for IAM users, you need to enable for the IAM user accounts also. You can register up to 8 devices of any combination of currently supported MFA types.
* It can be done in the three ways:
* **FIDO security key:** Fast Identity Online is a security key that provides the extra layer of security key. FIDO security key is used to protect the users from phishing fraud and identity theft.
* **Virtual MFA device:** It is an application runs on the another devices or phones and emulates physical device. Virtual authenticator apps which generates the time based one time password(TOTP) which is time based, The code need to typed correctly in the given time to sign-in.
* **Hardware TOTP token:** A hardware device which generates a six-digit code on the another device. It follows TOTP algorithm to sign in we need to type the correct code from the hardware device which is assigned. The MFA code which is generated is must be unique.

**Identity Federation:**

* To first learn about identity federation we need to learn about identity provider
* **Identity Provider:** Identity provider is external service provider which follows SSO (Single sign-in On). It uses external credentials of the users to access the AWS resources, with their external credentials by using these we no need to create any IAM user accounts. We can grant the permissions to there accounts.
* Identity Federation means process of connecting external identity provider. We can access the AWS resources with the existing credentials. To accessing the resources of AWS with existing credentials we need to provide trust relationship between AWS and external identity provider.
* For these we need to configuring roles in the IAM that defines permissions and policies for federated users. Once trust relation is established we can use external identity.

**Access Key:**

* Access key is long term credentials for an IAM user account or AWS root account. Access keys are used to make programmatic requests in AWS CLI or AWS API.
* Access key must consists of two parts Access key ID and Secret access key. Once the access key is created you need to store it in a secure location. If you delete the access key, You need to delete the existing and need to create a new one.
* You need to use both access key and secret access key to authenticate requests.