**IDENTITY ACCESS MANAGEMENT(IAM)**

**SECURITY**

* Security is a core requirement for any application whether it is hosted on an on-premises data center or a cloud such as AWS
* It is a fundamental service that protects your applications and data from a variety of cyber-attacks, security breaches, accidental or deliberate data deletions, theft, and much more.
* IAM is free and GLOBAL
* You can use IAM to create users and groups, assigning users specific permissions and policies, and a lot more.
* The best part of all this is that IAM is completely FREE. Yup! Not a penny is required to use it.
* IAM used for security purpose
* With IAM we can control entire AWS resource centrally
* We can share AWS ROOT account by creating IAM users
* It is not all recommended to use root account for daily activities or work, instead

use IAM user

* We should not share IAM Email/password to others
* ROOT user have Full permission and IAM user have limited permission

**MFA (Multi Factor Authentication)**

Example: google authentication

* MFA is highly recommended for ROOT user account and IAM user as well
* We need to setup MFA for every individual IAM user

You can access AWS in different ways depending on the user credentials: 1.Console Access

2.Programitical Access

* **Console Access**
  + - AWS Console website (GUI)
    - A password that the user can type to sign in to interactive sessions such as the AWS Management Console.

Root 🡪 email/password or username/password

* **Programmatical Access (CLI,SDK’s,Developer tools)**
  + - Login with KEYS (Access key and Secrete keys)
    - KEYS are user specific individual IAM users have their own keys
    - It is not recommended to share the KEY to any one and create the KEY’s based on the requirement, don’t create it un-necessarily
    - KEY’s also have same permissions like console
    - Every IAM user can have two set of KEY’s
    - IAM user we can attach and detach the permissions/policies anytime
    - We should not create KEY’s for ROOT account

**KEY TERMINOLOGY FOR IAM**

**USERS**

End Users such as people, employees of an organization

**GROUPS**

* Group is a collection of IAM users that has a particular set of permissions assigned to it

**For example,** a set of users who perform admin tasks can be clubbed under a common group called as administrators.

* Groups under groups are not possible/nested groups are not possible
* It is possible to attach multiple policies to the IAM users and group also, Max limit is 10
* We can attach and detach policies to IAM users and groups
* If you attach any user to the group his/her individual policies remains same and the new permissions will be inherited to the IAM users
* We cant assign/create KEY’s to IAM group
* Keys are only IAM user and not for group
* IAM user can be multiple groups at the same time
* IAM groups are used to assign policies to the bunch of IAM users at the same time

**POLICIES**

A policy is a document that lists one or more permissions. You can attach policies to virtually anything in AWS, from users and groups to individual AWS resources as well.

Policies are made up of documents called policy documents. These documents are in the format of JSON and they give permissions as to what a User/Group/Role is able to do.

**Two types of policies**

* + - **Managed Policy (Created and Managed by AWS)**

EC2FullAccess

S3FullAccess

* + - **Inline Policy (Created and Managed by Customer)**

**IAM Groups (R53 Service Access)**

**IAM Users**

* + - * Varun -> EC2 + R53
      * Rajesh -> S3 + R53
      * Ravi -> EFS + R53

**Note :**

* + - Policies are written in JSON format
    - Policy documents contains permissions

**IAM FEATURES**

* Centralized control of your AWS resources
* Shared access to your AWS account
* Granular permissions
* Identity Federation (Including Active Directory, FB, LinkedIn)
* Multi Factor Authentication
* Provide temporary access for users/devices and services where necessary
* Allows to setup your own password rotation policy
* Integrated with many different AWS Services

**Shared access to a single account:**

You can create and provide users with shared access to your single account with real ease.

**Multi-factor authentication:**

along with your password, you will also have to provide a secret key/pin from a special hardware device, or even from software apps such as Google Authenticator.

**Global reach:**

IAM is the Global, not specific to the Region

**Delete your root keys**

**Delete your root access keys. Now why would you want to do? What are root access keys?**

* Root keys simply consist of an access ID and a secret key
* Can be used to programmatically access any AWS service
* Each user that you create gets its own set of keys.
* The secret key has to be protected and kept under lock and key at all costs.

**AWS CLI VS Console**

* Access key ID and Secret key ID are used with AWS CLI or API’s
* Email ID or username and password is used to login to AWS console.
* You cannot login to AWS Console using access and secret keys
* You cannot login to CLI with Console username and password.

**Permissions**

* + Permissions provide you with access to and control of various AWS resources.
  + They are also responsible for controlling actions that you can perform on the resources.

They are also responsible for controlling actions that you can perform on the resources.

**User-based permissions**

**Resource-based permissions**

**User-based permissions**

These permissions are attached to IAM users and allow them to perform some action over an AWS resource.

User-based permissions can be applied to groups as well.

**Two Categories**

**Inline policies**

* created and managed completely by you

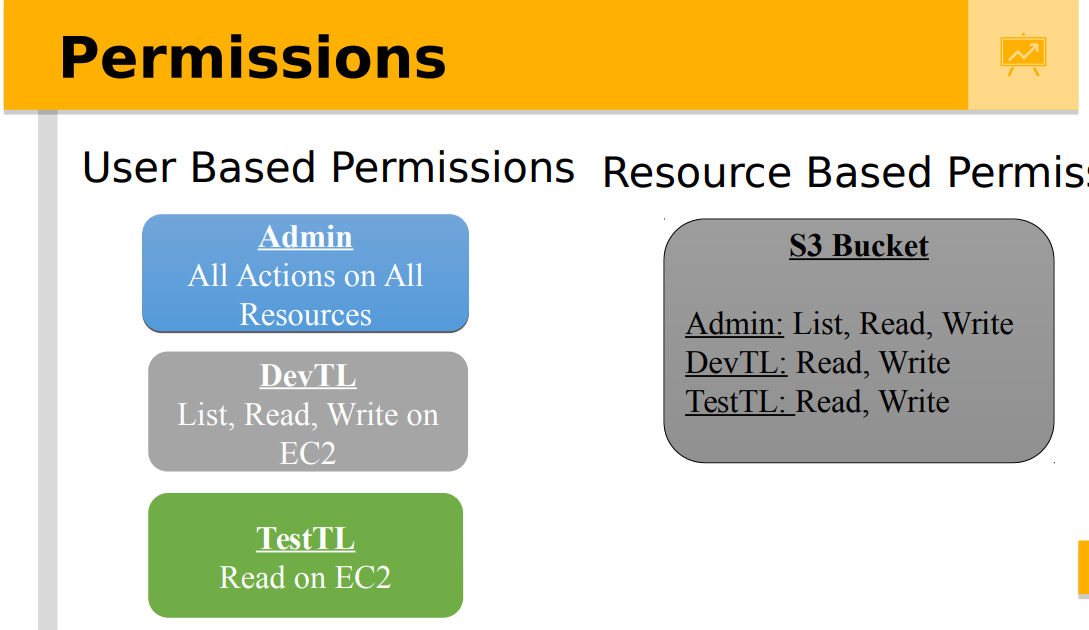
**Managed Policies**

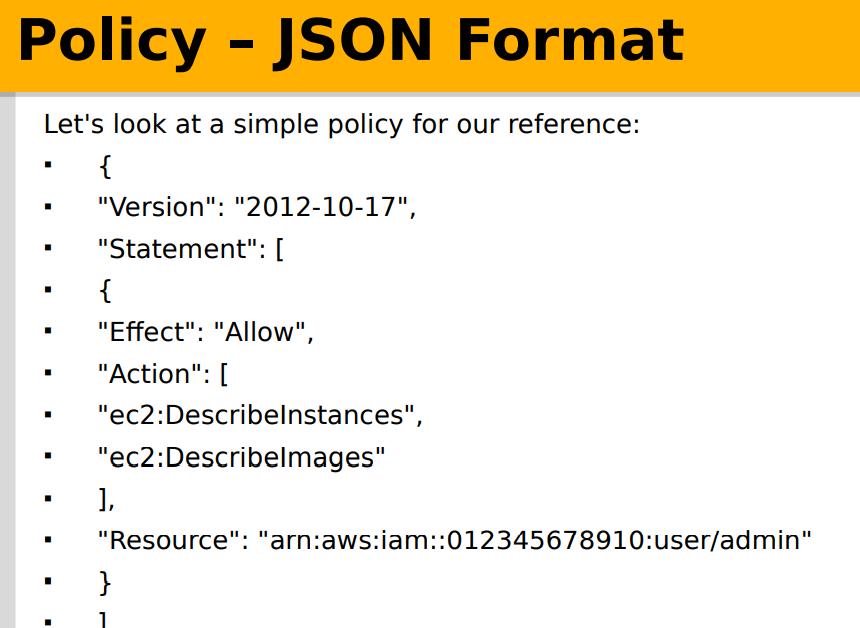
* created and managed more by AWS itself

**Resource Based Permissions**

User has specific level of access to a particular AWS resource along with what actions they can perform on it.

These categories of permissions are only inline-based this means that they are completely managed and created by you.



****

**IAM TAGS**

* TAG’s are Key-value pair
* TAS’s are used for authentication purpose
* TAG’s are also used for Cost optimization
* TAG’s are not IAM specific, it is thought-out AWS
* TAG’s optional but highly recommended
* 50 Tags per resource

Example: **Key -> Value**

IAM User -> Maharshi

Team -> DevOps

Email -> [dlosdevops@discover.com](mailto:dlosdevops@discover.com)

**IAM ROLES**

* **Roles:** Temporary access without credentials
* Roles are a group of permissions that grant users access to some particular AWS resources and services.
* 1 ROLE can be attached to multiple EC2 instances at the same time
* IAM user is used to login to AWS console not SERVER’s
* If you don’t configure KEY’s on Linux machine we cont access AWS services
* If you configure KEY’s on Linux machine KEY’s are stored locally on the machine which is not safe (because it might to hacked)
* If we use the ROLES, we need to configure KEY’s on the machine
* Based on the permissions that you have attached to the ROLE, those permissions are available from the machine
* 1 EC2 instance can have only 1 ROLE attached at the same time

**Difference:**

* Policies are applied to users and groups that belong to a particular AWS account
* Roles are applied to users who are generally not a part of your AWS account
* Use roles to delegate access to users, applications, and services that do not have access to your AWS resources.
* Use roles to create federated identities where a user from your organization's corporate directory gets access to your AWS resources on a temporary basis

**Identity Provider/ Federation**

* SSO (Single Sign On)
* Only one single sing we can login into AWS

**Multi-Factor Authentication**

* AWS Multi-Factor Authentication (MFA) is the practice or requiring two or more forms of authentication to protect AWS resources.
* It is an added security feature available through Amazon Identity and Access Management (IAM) that strengthens username and password credentials.
* **Prerequisites**: Google Authenticator installed on smart phone