IAM (Identity & Access Management)
==========
-> An AWS Identity and Access Management (IAM) user is an entity that you create in AWS to represent the person or
application that uses it to interact with AWS Services.

-> AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources.

We can use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

-> IAM helps protect against security breaches by allowing administrators to automate numerous user account related tasks.

-> Best practice of using the root user only to create your first IAM user.

-> Enable Multi Factory Authentication (MFA) for Root Usr

-> By using Google Authenticator App we can configure "Virtual MFA"

Best Practices:
==========
- When we login AWS using 'email' and 'password', that has complete access to all AWS services and resources in the account (Root account).

- Strongly recommended that you do not use the "root user" for your everyday tasks, even the administrative ones.

- Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

- IAM user is truely global, i.e, once IAM user is created it can be accessible in all the regions in AWS.

- Amazon S3 also considered as Global but, it is not truely global. When we create a bucket in S3 it displays all the buckets of other regions in one place , so that is the reason we are calling AmazonS3 is Global (but partly global).

- But IAM is 100% Global. Once you create IAM user you can use it anywhere in all the regions.

1. Main things in IAM is
    -Roles
    -Users
    -Policies / Permissions
    -Groups

2. IAM users can be accessible by the following 3 ways.
    -through AWS console
    -CLI (Command Line Interface)
    -API (fast glaciers)

3. In MNCs , permissions will not be provided for individual users. Create the Groups and add the users into it.
    Users & Groups are for the Endusers.
    Roles are for the AWS Services.

Steps:
====

1. Create an IAM user
   Services - Security, Identity, & Compliance - IAM
   Users---<Add user>
        User name* = Iamuser1
        Access type = 'select' both "Programmatic Access"
                                "AWS Management Console access"

        Console password = 'select'
                           custom Password = (********somepassword eg:test1234)

                click <NextPermissions>
        (Note: we are not providing any permissions as of now, just <create user>)

        Once the IAM user has been created.
                AccessKeyID =AKIAIEJH7Z3FDKH36YWQ
                Secretaccesskey=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

        (Note: Once you close this window, AccessKeyID and Secret Accesskey has gone, so save it somewhere)

  - Best Practice is never give an individual permissions to the user, as users will be changed frequently, when they left the organization.
    So Need to create the Groups and assign the users to it.

2. Group
        <create new group>
        Groupname =admins
        (Note: no need add any policy now).
        <creategroup>

3. Add user to this group

        click on newly create group 'admins'
        <Add users to Group>

        GroupARN =arn:aws:iam::540105522204:group/admins

  -Always add the permissions to the 'Groups' level not to the 'users' level. Its a Best Practice in the real-time.

****
Policies:
****
 - When we want to add the permissions to the the groups is through the 'Policies'.
 - Default AWS Policies are appear in'Orange color Icons'
 - One disadvantage of AWS Default Policies are , we can't customize the policies to apply to the Groups.
 - To provide customized policies to apply to Groups, we need to create the new one and apply to the Groups.

4. Now, we will add 'Administrator Access' Permissions to the user(Iamuser1) we create.

        Groups -Admins-tab<permissions> ---<AttachPolicy>---'select' Administrator Access----<AttachPolicy>

    -Dashboard -Customize the IAM link replacing the ID with any name. To Hide the ID need to customize.

        IAM user sigin in

        https://4234324234.signin.aws.amazon.com/console

```
        After Customize

        https://classroomuser.signin.aws.amazon.com/console

- Open the new tab in the browser
        https://classroomuser.signin.aws.amazon.com/console

        IAMuser =Iamuser1
        password=test1234

5. Now need to login using the IAM user, which we created.

        Once login , we can launch an EC2 instance.As this user(Iamuser1) is provided with Admin
access.

=====================================================================
Requirement:
=========+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

I got an requirement to create a new user and he should be able to do only 'stop' and 'start' ,
'reboot' select instances only.
He should not have the permissions to terminate the EC2 Instances.
He should not have the permissions to create the new EC2 Instance.

1. Login to your AWS Console with your root login.

2. IAM -Create another user
        User name* = Iamuser2
        Access Type ='select'  "AWS Management Console access"

        'select' CustomPassword ="<somepassword>"
        <NextPermissions>
        Not selecting any group here
        <createuser>

3. Signout and Login using the 'Iamuser2' and its credentials

        Open browser
                https://classroomuser.signin.aws.amazon.com/console
        login with Iamuser2 credentials

        Services ---EC2
        you will get an 'Authorization Error'

4.  To view EC2 instances need to provide read permission to the user 'Iamuser2'.
     - using Tags, we can provide permissions to this user.

     Login using the Root user
        EC2 Instances
        Select the Running Instance
         click on tab <Tags>
                add new tag
                   Key =user
                   Value=Iamuser2
                    <save>

5. Using this we can restrict the user to create EC2 instances. We can allow him to do only 'stop'
and 'start' Instances.
    For this, need to write the custom scripts.

        Open the browser search for ='restrict aws user ec2 instance'
        https://aws.amazon.com/premiumsupport/knowledge-center/restrict-ec2-iam/

        copy the script and open in any editor and customize it.
```

```
        arn:aws:ec2:us-east-1:111122223333:instance/*"
        (Note: For every service we have arn (amazon resource name), but for EC2 there is no arn
naming)

        InterviewQuestion:If anyone ask you , arn is not displaying for the EC2 instances?
        Ans:Simply say that, ARN is not visible for the EC2 instances, but for the other services
like S3, we have                          ARN url.



copy the script

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:StartInstances",
                "ec2:StopInstances",
                "ec2:RebootInstances"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/Owner": "Bob"
                }
            },
            "Resource": [
                "arn:aws:ec2:us-east-1:111122223333:instance/*"
            ],
            "Effect": "Allow"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:Describe*",
            "Resource": "*"
        }
    ]
}



After Customization
===================

        {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:StartInstances",
                "ec2:StopInstances",
                "ec2:RebootInstances"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/user": "Iamuser2"
                }
            },
            "Resource": [
                "arn:aws:ec2:us-east-1:449938344550:instance/*"
            ],
            "Effect": "Allow"
        },
        {
            "Effect": "Allow",
```

```
            "Action": "ec2:Describe*",
            "Resource": "*"
        }
    ]
}

=============
Note:
        449938344550 = Root AccountID

6. copy the script after customization
    IAMUser
      Policies ---<createPolicy>---'select' JSON tab
        paste the customized script.

        <ReviewPolicy>

7. Review Policy
        Name ='UserRestrictEC2Instance'
        <createpolicy>

8. Now, need to add this policy to the user or groups.
        select Users
             'Iamuser2' ---Permissions(tab)-- Add Permissions ---AttachExisting Policies directly
             Filter policies ='UserRestrictEC2Instance'

              Select the policy(UserRestrictEC2Instance') ---<Review>--<AddPermissions>

9. Login to IAM user console
        Iamuser2/password

   - Now Try to Terminate the EC2 Instance. It throws an error
   - Try to Launch an EC2 instance , it throws an error.
     Like this we can restrict the user by creating some policies and apply to it.
     AWS provides the readymade(default) policies we need to customize as per our requirement.



-------------------------------------------------------------
What is IAM ?
What is Root Account ?
How to enable MFA for root account

What is IAM account
How to create IAM account
Programmatic Access Vs Console Access
Attaching Policies to User
Creating Custom Policy

Creating User Group
Adding Users to Group
Adding Policies to User Group

-------------------------------------------------
```