Identity Access Management - IAM:

------------------------------------------------------

--> IAM is a Webservice, which can control the secure access to the AWS Account

--> We can control, who can login to the AWS Account & What actions the users can perform ( Authentication & Authorization)

- Users

- Groups

- Policies

- Roles

--> On AWS, we have 2 types of users

Root user --> Is the owner of the AWS Account, Will be created at the time of AWS Account creation. root user will get unrestricted access to the AWS Account.

Note: AWS recommends not to use the root user for day-to-day activities, Is to SETUP THE aws account.

IAM User: --> is the user account for every employee in the same AWS account.

NOTE: A Default IAM user will get NO Permissions

--> Permissions can be provided to the users using different methods

Policy:

----------

--> Is a document, written in JSON Format and it contains the permissions

--> Policy can be attached to the User or Groups, so that the users and all the group members will be applied with the policy automatically.

User Group:

-------------------

--> is a container, in which we can collect similar type of users.

--> User can be a part of multiple user groups, so that the user will get the permissions from all the groups.

--> There are 2 types of user access available for AWS users

1. Management console access type

2. Programatic access Type

1. Management Console:

---------------------------------------

The user can access the AWS account using Management console, The Browser ( ex: chrome, edge, firefox etc)

--> The user wil use the " UserName & Password " combination to login to the AWS Account

2. Programatic Access:

-----------------------------------

The user can access the AWS account with the help of AWS CLI Tools, SDK, API or any other development tools

--> The user will use "Access Key ID & Secret Access Key" combination to login using Programatic access

Management Console Access Type:

------------------------------------------------------

1. Create a default IAM user "user0"

Console Sign-In Url : https://601594011753.signin.aws.amazon.com/console

UserName : user0

Password : jZ1mfvH7-(

Account Alias --> is an alternative name to the AWS Account ID, which can be used at the time of an IAM user Sign-In

2. Create an Account alias for the AWS Account as "phani32"

3. Login as "user0" by entering the Account alias instead of account ID

4. Create IAM user "user1" and provide full permissions on the AWS account by attaching a policy directly

Note: There are 3 different ways that we can provide permissions to the IAM users

- Attach a policy Directly

- Add the user to a Group

- Copy the permissions from an existing IAM user

5. Login as "user1"

a) Create another user "user2" with no permissions

b) Create a security group in the Service EC2

6. Create an IAM user "user3" and provide Read-Only Permissions on the entire AWS Account

7. Create a new IAM policy "EC2FullAccess" which can provide EC2 Full access

8. Create an IAM user "user4" and attach the policy created "EC2FullAccess"

9. Login as "user4" and verify the permissions

10. Create a User Group "AdminGroup" & Attach a policy which can provide full access on AWS Account.

11. Create an IAM User "user5" and add the user5 into the Group "AdminGroup" ( User5 should get Full access on AWS Account)

12. Login as "user5" and verify the administrator permissions

13. Add an exisiting user "user0" into the group "AdminGroup" and verify that the user0 got Full access on the AWS Account

Note: We can copy the permissions from an existing user to a new user.

14. Create an IAM user "user6" and copy the permissions from "user3" (user3 is having ReadOnlyAccess)

15. Reset the forgotten password for "user0"

- Login as root user or Admin IAM user

Programatic Access Type:

----------------------------------------

--> Programatic Access type of a user can access the AWS account using AWS CLI Tools, SDK, API or any other development tools

--> Access Key & Secret Access Key should be used for authentication

16. Create an IAM user "user11" With full access on AWS account and provide Programmatic access type.

Access Key : AKIAYYEOMEBU2JNSZMKB

Secret Access Key : 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

17. Access the AWS Account as "user11"

- Install AWS CLI Tools --> https://aws.amazon.com/cli

- Configure the AWS Credentials

aws configure

18. get the list of IAM users as "user11" using AWS CLI Tools

aws iam list-users

19. Create a new IAM user "user12" by accessing the AWS Account as "user11"

aws iam create-user --user-name user12

20. Generate a new pair of access Key & Secret Access Key, if the user lost his keys

AKIAYYEOMEBUYYDZLVSW

Yy8v3s4LoAtuMm0pjqlq7n7+KaWMdyUz2LrWfWqY

21. Deactivate the First Access Key of "user11"

22. Reconfigure the CLI tools with the new Pair of access Keys

--> We can provide both type of access to a single user

23. Assign Management console access type to the user "user11" by keeping the programmatic access as it is.

NOTE: Always DENY permissions will have the Highest Priority Over ALLOW Permissions.

24. Create an IAM user "user14" and apply 2 policies which are conflicting with each other

a) Create a IAM Full access - Deny Policy

b) Create an IAM user "user14" and attach the below 2 policies

- Administrator Access -Allow

- IAMFullAccess\_Deny"

c) Login as "user14" and observe the permissions what the user got.