Assignment Interview question

1. What is the need of IAM?

**Ans:**

In IAM we can create Groups by which we can manage teams and we can define the policies.

AWS Identity and Access Management (IAM) is a web service which helps securely control access to AWS resources. We can use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

1. If I am a non tech person, how will you define policies in IAM.

**Ans:**

Policies are used to set Boundaries. An AWS account would charge some amount for their cloud services, in order to control or restrict the budget we use policies.

So that only authorized users have access to certain number of services.

An AWS IAM Policy defines the permissions of an identity (users, groups, and roles) or resource within the AWS account. An AWS IAM policy regulates access to AWS resources to help ensure that only authorized users have access to specific digital assets.

Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents.

1. Please define a scenario in which you would like to create your own IAM policy.

**Ans:**

AWS apply a default policy which is called full AWS access, nothing is restricted here. Hence deny policies will be created to restrict the permission.

Example: Multiple policies are involved in this example.

Thor is a developer and trying to access a resource inside AWS. Thor has two policies which apply to him. Policy one and policy two. Thor is also a member of group Developers group. Developers group also has policy associated with it.

So Thor attempts to access an AWS resource, that AWS resource might also have a policy associated with it. All these policies have one or more statements inside them, and AWS collects all of the statements in all of the policies which apply.

So the users directly and the groups the users are in and any resource policies on the resource that they are attempting to access, collects them together and evaluate them at the same time. It applies the Rule deny, allow, deny.

Thor--🡪 Policy one and Policy two



Thor → Developers → Resource

↗ ↗ ↖ ↖ ↑↑ ↑↑

↑↑ ↑ ↑ ↑↑ ↑↑

✓

✓

✓

X

✓

✓

X

✓

✓

X

✓

X

Policy one Policy two Policy three Resource policy

4. Why do we prefer not using root account?

**Ans:**

Account root user has full access to everything. It’s like a GOD, which can’t be restricted.

Account root user has full control over one specific AWS account and any resources which we created within it. Account root user can’t be restricted. This is the reason why we need to be very careful with the account root user because if the user name and password ever become known the results can be disastrous, because the details can be used to delete everything within the AWS account.

Therefore we don’t prefer not using root account.

1. How to revoke policy for an IAM user?

**Ans:**

We can delete a customer managed policy to remove it from the AWS account. But cannot delete AWS managed policies.

We can delete IAM policies using the AWS Management Console, the AWS Command Line Interface (AWS CLI), or the IAM API.

To delete a customer managed policy (console):

Choose **Policies** to delete.

1. Select the check box next to the customer managed policy to delete. Choose **Actions**, and then choose **Delete**.
2. Confirm that you want to delete the policy, and then choose **Delete**.

6. Can a single IAM user be a part of multiple policy via group and root? how?

**Ans:**

A single IAM user can be part of multiple policy via group and root.

An IAM user can be a member of 10 groups and there are 5,000 IAM users limit for an account.

For example: Inside an AWS account we have got 2 groups

Group 1: Developers ----> Tina, Rina

Group 2: QA -----🡪Mike, Tina

In Developers group – →Tina and Rina are users

In QA group – →Mike and Tina are users

User Tina is the member of both Developers and QA groups.

An IAM user can be a part of multiple groups. We can make groups which represents teams or projects. Groups have policies attached to them.

In the above example Developers and QA groups have their own policies attached to them. And an IAM user Tina has its own policy. So Tina has got policies attached to Developers and QA groups as well as her own user policy.

Effectively Tina has got 3 policies, so collect all of the policy allows and denies from user directly and from 2 groups as a collection. It applies the Rule deny, allow, deny.