1. What is the need of IAM?

Ans. IAM is the security discipline that incorporates identification, authentication, and authorization by defining the right policies for right people to use the right resources like applications or data when they need to, without any interference.

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

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

Ans. Let’s assume that there are 10 test user accounts out of 5 users’ password should expire in a month based on their account type and the remaining 5 users’ password should expire in 6 months since these 5 have different account type. So, in this scenario I would create 2 customized policies.

4. Why do we prefer not using root account?

Ans. If the root account credentials are hacked, they will be able to access or change anything in the account giving them the ability to misuse any data or resources in the account. They could even make you incur unnecessary costs by creating resources in your account. And also, the IAM users in that root account can also be affected. Hence it is always recommended to prefer IAM account for all the tasks.

5. How to revoke policy for an IAM user?

Ans. You can revoke policy for an IAM user using the AWS Management Console, the AWS Command Line Interface (AWS CLI), or the IAM API.

Using the AWS Management Console:

* Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.
* In the navigation pane, choose **Policies**.
* Select the check box next to the customer managed policy to delete. You can use the search box to filter the list of policies.
* Choose **Actions**, and then choose **Delete**.
* 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. Yes, a single IAM user can be a part of multiple policy only via group but not via root.

Via Group:

* Create a user and attach ‘x’ policy to that user.
* Now create a group and assign some ‘y’ policy to that group.
* Add that user to the created group so that the user can be a part of both ‘x’ and ‘y’ policies

Via Root: You cannot attach identity-based policies to the root user, and you cannot set the permissions boundary for the root user.