What is IAM in AWS?

* IAM stands for Identity and access management.
* It is a web service provided by Amazon Web Services (AWS) that helps us to securely control access to AWS services and resources.
* Using IAM, we can manage users and their access by creating and managing AWS users and groups.
* Using permissions, we are able to allow and deny user access to AWS resources.

(or)

Amazon Web Services, Identity and Access Management (AWS IAM).

In other words, AWS IAM has a specific way of dealing with giving access and consent control within the organization.

IAM has overall control over who can use the assets and under what conditions they can be used. These are also classified as Authentication and Authorization.

What is AWS IAM, and why is it important in a DevOps environment?

Answer: AWS IAM is a web service that allows you to control access to AWS services and resources. In DevOps, IAM is critical for securing and managing access to AWS resources, ensuring that only authorized individuals or services can interact with AWS services and perform actions

Key Features of IAM:

1. Users

2. Groups

3. Roles

4. Policies

5. Multi-Factor Authentication (MFA)

**Explain the concept of IAM users, groups, and roles in AWS IAM.**

Answer:

— IAM Users: IAM users are individual entities with unique credentials used for authentication. Each user can have specific permissions assigned to them.

— IAM Groups: IAM groups are collections of IAM users. Permissions are assigned to groups, and users added to those groups inherit the permissions.

— IAM Roles: IAM roles are used by AWS resources or external services to obtain temporary security credentials. They do not have long-term access keys and are ideal for granting permissions to applications and services running on AWS.

Let’s discuss about one by one

**Users**: User is nothing but a person who interacts with AWS resource. Each user has a unique name within the AWS account.

**Group**: Group is nothing but it is a collection of IAM users. You can use groups to provide permissions to users who are in that group, which can make it easier to manage the permissions for those users.

**Roles**: Role is nothing but set of permissions, by using role we can provide set of permissions to user or users in a group at a time.

(Or)

Defines a set of permissions for making AWS service requests. IAM roles are intended to be assumable by anyone who needs them (including users outside of your organization, under certain conditions).

**Permission**: one particular access providing to user at a time.

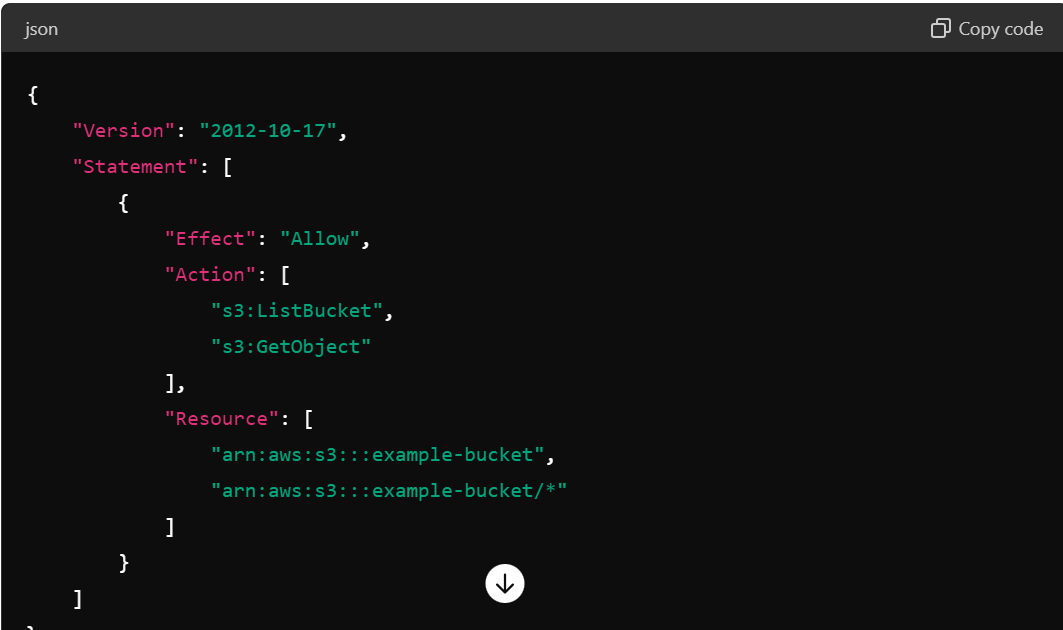
**Policy**: Set of permissions we can call it as policy.

**Role**: Set of permissions defined by policy.

Example:

Here s3:listBucket is one permission in S3 resource

s3:GetObject is one permission in s3 resource.



**Policy**:

Policy is nothing but set of permissions. Policies are attached to users, groups, and roles to define what actions they can perform on which resources.

Policies are of two types:

1. Managed policy

2. Inline policy

**Managed Policies**: Standalone policy or standard policy or default policy

The policy which comes with aws that you can attach to multiple users, groups, and roles.

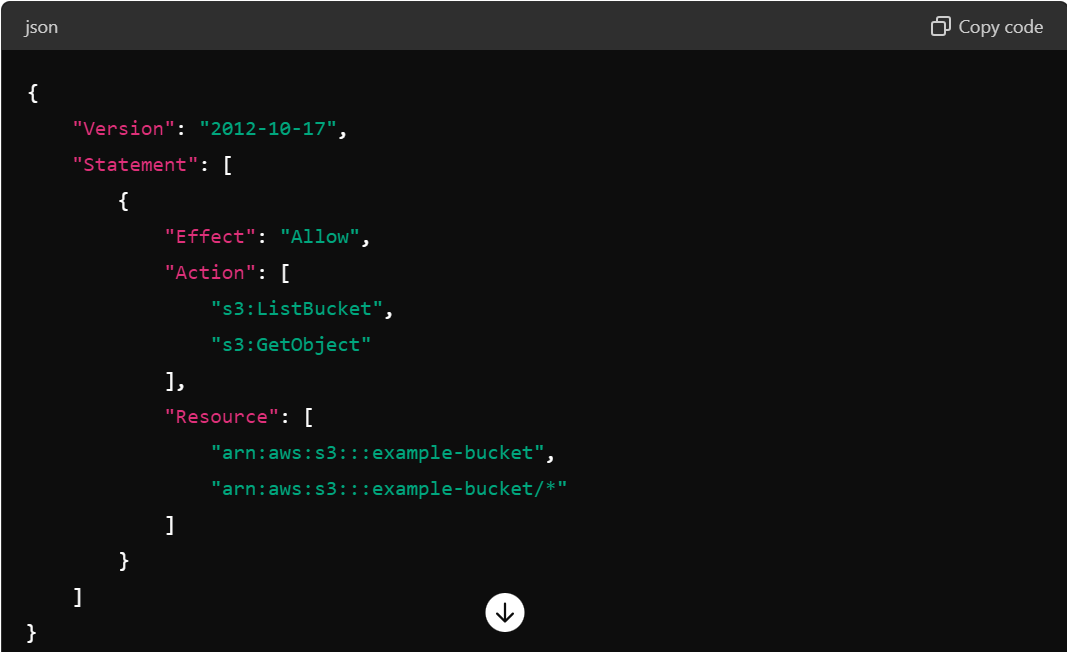
**Inline Policies**: The policy which we create and manage directly within a single user, group, or role.

**Policy Types**:

* **Identity-based Policies**: Attached to IAM users, groups, or roles and define what actions they are allowed to perform on which resources.
* **Resource-based Policies**: Attached to resources like S3 buckets, and define who (which IAM users, roles, or AWS accounts) can perform what actions on that resource. It means policy assign to particular resource like s3, EC2, RDS, EBS…

**Policy Structure**:

* **Statements**: Each policy is a collection of statements. Each statement grants or denies permission.
* **Effect**: Can be "Allow" or "Deny".
* **Action**: Specifies what actions are allowed or denied (e.g., s3:ListBucket).
* **Resource**: Specifies which resources the actions apply to (e.g., a specific S3 bucket).
* **Condition**: Optional conditions for when a policy is in effect.



5. **Multi-Factor Authentication (MFA)**: It is an extra layer of security for our AWS environment by requiring a second form of authentication.

Example: Microsoft Authentication

**Difference between authorization and authentication?**

**Authentication**: This process verifies the identity of a user attempting to access a system. Common methods include passwords, biometrics, and multi-factor authentication (MFA).

Example: User-id, password, Access key, MFA, Roles

**Authorization**: Once authenticated, authorization determines what an authenticated user is allowed to do. This involves granting or denying access to resources based on the user's permissions and roles.

Example: IAM policies, Resource-based policy, Permissions

**CRUD**: CRUD stands for Create, Read, Update, and Delete.

When dealing with IAM policies and permissions, these operations define what actions a user or role can perform on AWS resources.

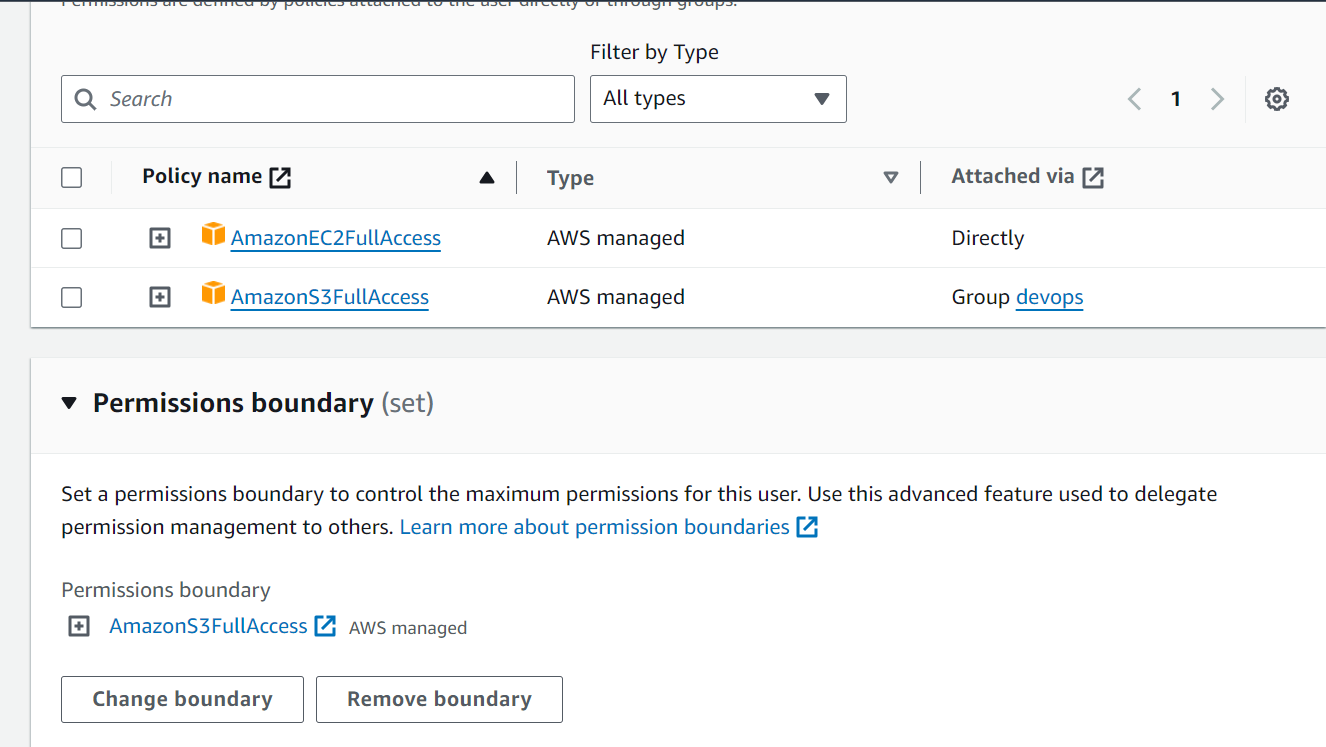
**What is permission boundary in AWS while creating an IAM user?**

>>> Permission boundary can limit/restrict the permissions granted by other policies. It means what ever the other access assigned to users it doesn’t works. The access which comes from permission boundary will only work to user.

Example:

Suppose we have created a user called Praveen and provided the S3 full access and EC2 full access. After some days we got a requirement to restrict the user only to access EC2. In that case what we have to do is set EC2 full access in permission boundary. Now user can access to only EC2 and the rest all policies will be restricted.

From the below screenshot user will be able to access to only S3 full access.



**What is access key and secret key?**

**Access Key**: This is like a username. It is a public identifier that is used to identify the user or application making the request.

**Secret Key**: This is like a password. It is a private identifier that is used to securely sign and verify the requests made using the access key.

Together, they ensure that only authorized users or applications can access and perform operations on the cloud services. Here's a simple analogy:

Usage:

By using aws configure in linux server, we can able to access aws services. While we install aws-configure in linux server it asks for access key and secret key.

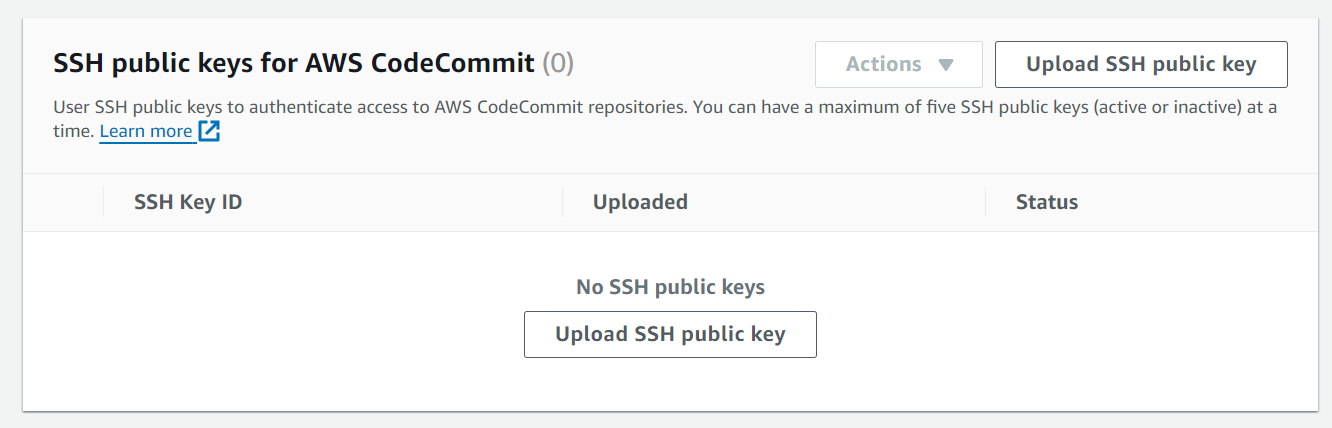
**Example:**

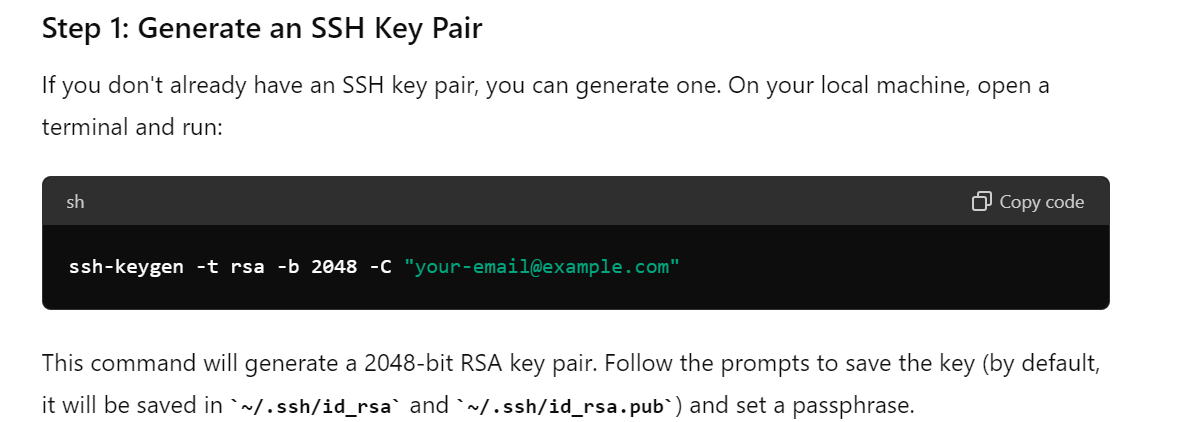
Access Key: Your email address.

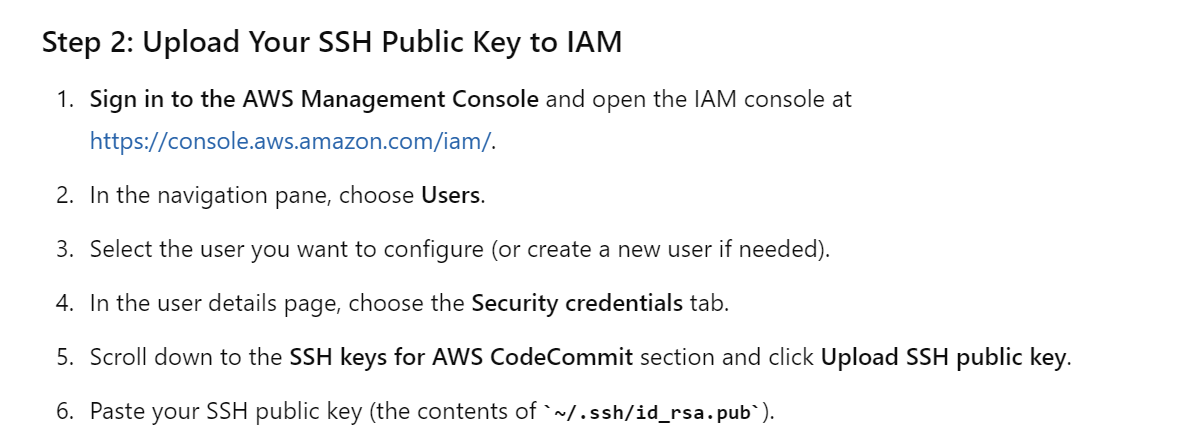
Secret Key: Your email password.

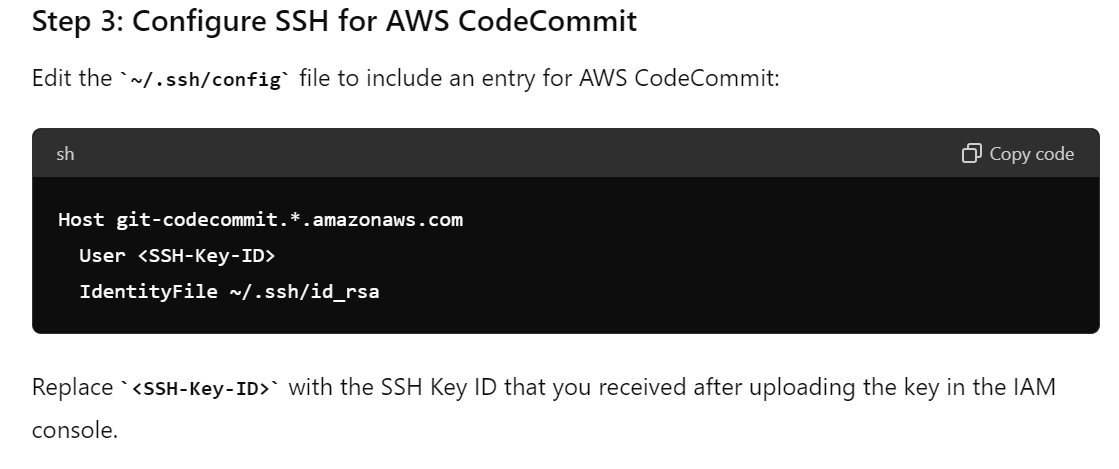
**What is SSH public keys for AWS CodeCommit in IAM?**

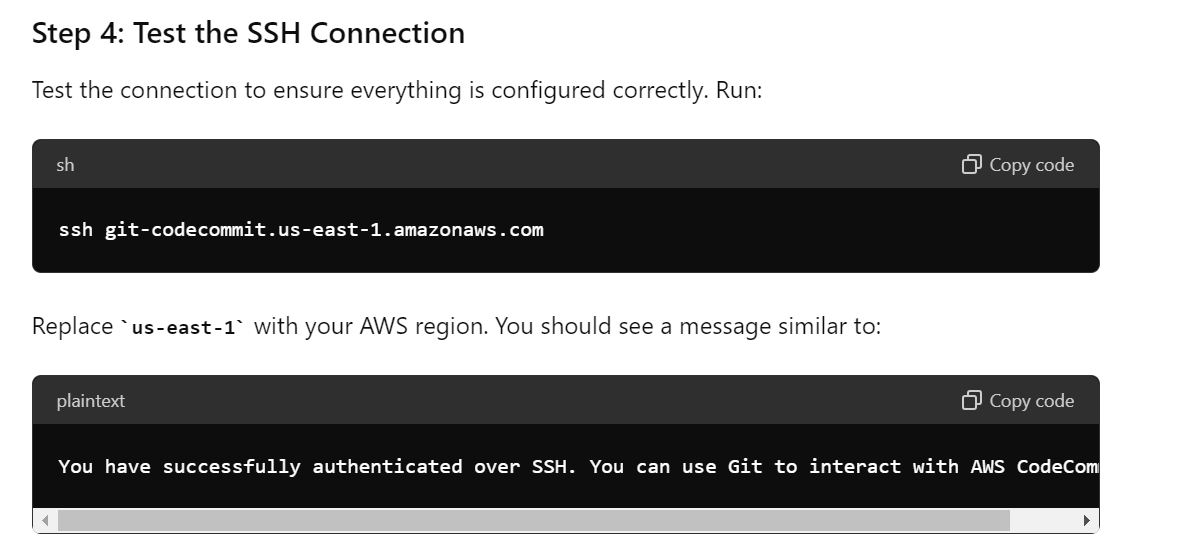
If we don't already have an SSH key pair, we can generate public key. On your local machine, open a terminal and run:

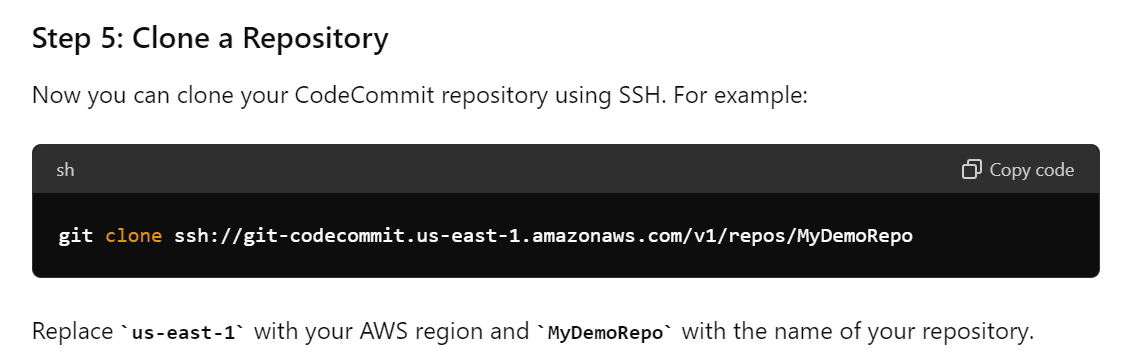












\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Q1) I am viewing an AWS Console but unable to launch the instance, I receive an IAM Error how can I rectify it?**

Answer: As AWS user I don’t have access to use it, I need to have permissions to use it further

**Q2) I don’t want my AWS Account id to be exposed to users how can I avoid it?**

**(Check this in real-time scenario)**

Answer: In IAM console there is option as sign in url where I can rename my own account name with AWS account

**Q3) The user has created the applications, which will be hosted on the EC2. The application makes calls to the Dynamo DB to fetch on certain data. The application using the DynamoDB SDK to connect with the EC2 instance. With respect to best practice for the security in this scenario?**

Answer: The user should be attach an IAM roles with the DynamoDB access to EC2 instance.

**How do the application use the AWS credentials to access S3 bucket securely?**

Answer: Create an IAM role for the EC2 that allows list access to objects in S3 buckets. Launch to instance with this role, and retrieve an role’s credentials from EC2 Instance make metadata.

(or)

We need to create IAM role for S3 bucket and go to EC2 instance and attach this role to EC2 instance.

**Q4) What are IAM Roles and Policies, what is the difference between IAM Roles and Policies.**

Answer: Roles are for AWS services, where we can assign permission of some AWS service to other Service.

Example – Giving S3 permission to EC2 to access S3 Bucket Contents.

Policies are for users and groups, Where we can assign permission directly to user’s and groups.

Example – Giving permission to user to access the S3 Buckets.

**Q5) How a AWS root user is different from in IAM User.**

Answer: Root User will have access to entire AWS environment and it will not have any policy attached to it. While IAM User will be able to do its task on the basis of policies attached to it.

**Q6) What do you mean by Principal of least privilege in term of IAM.**

Answer: Principle of Least Privilege, organizations can minimize the risks associated with unauthorized access, data breaches, and malicious activities. It helps to minimize the potential damage that can occur if a user account or system is compromised, as the attacker would have limited access to sensitive resources.

**Q227) What is the meaning of non-explicit deny for an IAM User.**

Answer: When an IAM user is created and it is not having any policy attached to it in that case he will not be able to access any of the AWS Service until a policy has been attached to it.

**Q229) What is the benefit of creating a group in IAM.**

Answer: Creation of Group makes the user management process much simpler and user with the same kind of permission can be added in a group and at last addition of a policy will be much simpler to the group in comparison to doing the same thing manually.

**Q233) What is the benefit of creating the AWS in Organization.**

Answer: It helps in managing the IAM Policies, creating the AWS Accounts programmatically, helps in managing the payment methods and consolidated billing.

**Q2. What are the core elements of an AWS IAM policy statement and what is each element’s purpose?**

Principal: Identifies the principals that the statement applies to, most importantly and commonly an AWS IAM principal. (It limits the policy by using arn)

Effect: whether to Allow or Deny access if the statement applies; Deny always wins when multiple statements apply.

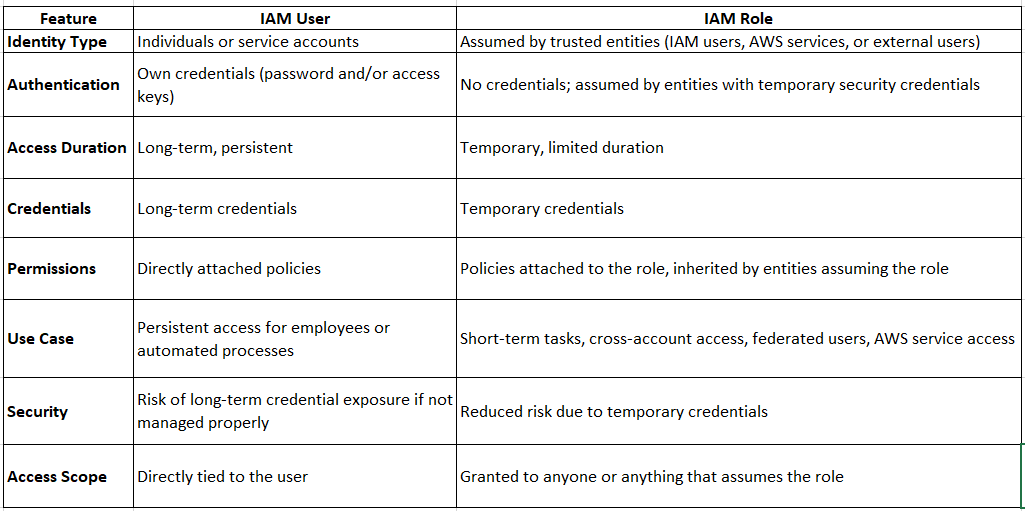
Action: one or more AWS API actions the statement will Allow or Deny the Principal to invoke as a string or list of strings. Supports wildcards, ? and \*.

Resource: one or more AWS resources the statement applies to, specified as ARNs. Supports wildcards, ? and \*.

**Q4. What happens if you have one IAM statement that allows a principal to perform an operation on a resource and another statement that denies that same operation on the same resource?**

The Deny effect always wins when multiple statements apply.

**What is the difference between IAM user and IAM Role?**



**1. In the IAM service, can we monitor the IAM user activity?**

Yes, you can monitor the activities of IAM users. If there is any violation, you can remove access for the IAM user

**2. What is Authorization in terms of AWS IAM service?**

It’s to provide Authorization for specific AWS resources – not all.

**3. What is federated user access management?**

A user who is allowed to access AWS resources from **third-party** vendors – such as Google, Facebook, Linked In, Corporate credentials, etc.

**4. What’s the other name of the IAM user?**

You can also be called an IAM entity. (Identity means entity)

**5. How to control Authorization in AWS IAM?**

You can control Authorization by creating policies.

**6. What are the 5 top security credentials in AWS IAM?**

* Key pair
* The E-mail address and Password
* User-id and Password
* Access Keys
* Multi-factor authentication

**10. What is CloudTrail in AWS?**

**>>>** By using cloudtrail we can monitor the IAM user

It’s a service that records the logs of each IAM entity so that you can use these logs for auditing and compliance purposes.

In these logs, you will get answers for what, where, when, who, which

* What was the request about?
* Where was the request made from and made to?
* When was the request made?
* Who made the request?
* Which resources were acted upon in response to the request?

**11. What are AWS IAM roles?**

Group – These people will have the same kind of Access

User–Specific IAM entity

12. **What are Temporary Security Credentials?**

These are short-lived security credentials. These you can create from the AWS STS service (AWS Security Token Service).

**13. What is the IAM Hierarchy of Privileges?**

* IAM user
* Root user
* User with temporary credentials

14. **What are the top AWS IAM Roles?**

In AWS IAM, there are two types of roles. The IAM user will have a permanent identity. The federated user (third party user) will not have an identity.

15. **What are the features of AWS IAM?**

There are many features of AWS IAM few of them would be:

* Identity federation
* Secure Access to AWS resources for applications that run on Amazon EC2
* Granular permissions
* Multi-factor authentication (MFA)
* Integrated with many AWS services
* Free to use

**16. In what ways can AWS IAM be used?**

You can work with AWS IAM in various ways, such as:

* AWS SDKs
* AWS Management Console
* IAM HTTPS API
* AWS Command Line Tools

**17. What are the terms of AWS IAM?**

The IAM terms include IAM Entities, IAM Resources, and IAM Identities.

**18. What is ‘Principal’ in IAM?**

A principal is an application or person who can request an operation or action on an AWS resource.

19. **What is ABAC for AWS?**

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called tags.

**20. What are the security features outside IAM?**

Some of the security features outside IAM are Amazon EC2, Workspaces, Amazon RDS and Amazon WorkDocs.

**21. Do I need to sign up for IAM?**

To use IAM, you do not need to sign up. You must first create an account to use IAM if you do not already have one. There is no charge for using IAM.

**22.Why would one use the feature of giving Access to users using an AWS account?**

Assume you want to add users to your AWS account and require them to use IAM attributes or have access to specific resources. In that case, you can use the feature of granting Access to AWS account users.

**23. How can one set up AWS IAM?**

AWS Identity and Access Management (IAM) enables you to securely manage access to Amazon Web Services (AWS) and account assets. Your account qualifications can also be kept private by IAM.

**24. Can one access all the features of AWS with only one account?**

One might obtain the characteristics that have been granted to them, and the Authorization can control the permissions. You can also enable access to resources across AWS accounts in some cases.

**25. What are some of the things you can do using IAM?**

Some of the services offered by AWS IAM are:

Manage passwords for IAM users

List the users in your AWS account and get information about their credentials

Manage permissions for IAM users

Add multi-factor authentication (MFA)

Tag IAM resources

View the actions, resources, and condition keys for all services

28. What is the IAM policy in AWS, and how does it work?

Answer: An IAM policy is a document that defines permissions, allowing or denying actions for users, groups, or roles. It consists of JSON statements with specified resources and actions. Policies can be attached to IAM users, groups, or roles, defining their level of access to AWS resources.

29. Explain the difference between an IAM policy and an IAM policy document.

Answer: An IAM policy is a named entity that defines permissions. An IAM policy document is the actual JSON or YAML document containing the policy’s permissions. You can have multiple policies attached to an IAM user, group, or role, and each policy has its own policy document.

30. What are managed policies and inline policies in AWS IAM?

Answer:

— Managed Policies: Managed policies are standalone policies that you can attach to multiple users, groups, or roles. They are created and managed independently and can be shared across AWS accounts.

— Inline Policies: Inline policies are policies that are embedded directly into a single user, group, or role. They are defined within the entity they are attached to and cannot be shared or reused outside of that entity.

31. What is the AWS IAM access key, and why is it used?

Answer: An AWS IAM access key is a pair of security credentials that consists of an access key ID and a secret access key. They are used to authenticate programmatic access to AWS services. DevOps engineers use access keys to interact with AWS services through the AWS CLI, SDKs, or automation scripts.

32. How can you ensure the principle of least privilege when setting up IAM policies?

Answer: To follow the principle of least privilege, IAM policies should grant only the permissions required for a user, group, or role to perform their tasks, and no more. Regularly review and update policies to remove unnecessary permissions. Utilize the AWS IAM policy simulator to validate the effect of policies.

33. What is MFA in AWS IAM, and how can it enhance security in a DevOps environment?

Answer: Multi-Factor Authentication (MFA) adds an extra layer of security by requiring users to present two or more separate authentication factors (typically something they know and something they have). In AWS IAM, MFA can be enabled for specific users, adding an additional level of protection when accessing critical AWS resources, especially for DevOps users with elevated privileges.

34. Explain the process of setting up cross-account access using IAM roles.

Answer: Cross-account access allows one AWS account to delegate access to another account using IAM roles. The process involves creating a role in the account granting access, defining a trust policy that specifies the trusted account, and granting permissions to the role. The trusted account can then assume the role to access resources in the granting account.

35. What is IAM access key rotation, and why is it important for security?

Answer: IAM access key rotation is the process of regularly updating or changing access keys to mitigate the risk of unauthorized access. It’s essential for security because it reduces the window of opportunity for attackers who might have access to compromised keys.

37. How can you securely manage AWS access keys, and what best practices should be followed?

Answer: Best practices for managing access keys include:

— Regularly rotating access keys.

— Limiting the use of long-term access keys in favour of temporary security credentials.

— Using MFA to enhance the security of IAM users with access keys.

— Ensuring that access keys are not hard-coded in code repositories.

38. Explain the use case for AWS IAM roles when interacting with AWS resources from EC2 instances.

Answer: IAM roles are used to grant permissions to AWS resources, such as EC2 instances. By attaching an IAM role to an EC2 instance, you can securely delegate permissions without the need for storing access keys on the instance. This is particularly useful for applications running on EC2 that need to interact with other AWS services.

(Or)

We can create an IAM role for Ec2 instance using S3 full access permission and attach the role to EC2 instance.

39. What is the IAM policy evaluation logic, and how does it determine whether an action is allowed or denied?

Answer: IAM policy evaluation follows a deny-by-default logic. If an action is explicitly allowed by any policy attached to a user, group, or role, it is permitted. However, if any policy attached to the entity explicitly denies the action or the action is not mentioned in any policy, it is denied.

40. Explain the “condition” element in IAM policies and provide an example use case.

Answer: Conditions in IAM policies allow you to specify additional criteria for when a policy should be applied. For example, you can create a condition to restrict access to an S3 bucket only from a specific IP address range, enhancing security by limiting access to trusted networks.

41. What is AWS Organizations, and how does it relate to IAM in a multi-account AWS environment?

Answer: AWS Organizations is a service that helps you manage multiple AWS accounts. In a multi-account setup, it allows you to centralize and simplify IAM management. You can create and manage policies and roles at the organization level, making it easier to control access across all accounts.

42. How can you monitor and audit AWS IAM activities for security and compliance purposes?

Answer: AWS provides AWS **CloudTrail** for monitoring and logging IAM activities. By enabling CloudTrail, you can track all IAM-related events, providing visibility into who performed actions, what actions were taken, and when they occurred. These logs can be used for security analysis and compliance reporting.

43. Explain the process of implementing a “least privilege” IAM policy for a DevOps team member.

Answer: To implement a least privilege policy for a DevOps team member, you should:

— Identify the specific tasks and resources they need access to.

— Create a policy that grants only the necessary permissions.

— Regularly review and update the policy to remove unnecessary permissions.

— Use IAM groups to manage and assign policies to simplify policy management.

44. What is the IAM policy “Effect” field, and what are the possible values?

Answer: The “Effect” field in an IAM policy can have two values:

— “Allow”: Grants permissions to perform specified actions.

— “Deny”: Explicitly denies permissions, even if they are allowed in other policies. Deny statements should be used sparingly.

45. How can you recover from a situation where you’ve locked yourself out of your AWS account due to overly restrictive IAM policies?

Answer: In such a situation, if you’re unable to sign in with sufficient permissions to fix your IAM policies, you can contact AWS support. They can assist in regaining access by verifying your identity and making necessary adjustments to your IAM policies.

**Question**: Describe a scenario where IAM authentication is more secure than using traditional access keys

**Answer**: Consider a scenario where an IAM user’s credentials (username and password) are leaked. If the user only uses regular access keys, an attacker could gain unauthorized access to resources. But, if the IAM setup uses authentication, especially with an extra layer like multi-factor authentication (MFA), even if the login details get leaked, the attacker would still need something extra (like a temporary code from a device) to actually get into the system, making it more secure.

**Question**: You notice that some IAM users are not using multi-factor authentication. How would you enforce MFA for all IAM users in your AWS account?

**Answer**: To make sure all IAM users use MFA, I can go to the IAM console, choose “Users,” and turn on MFA for each person one by one. But for a simpler way, I can create a MFA policy that says everyone must use MFA. Then, I attach this policy to all users or groups. This rule makes sure nobody can access things without using MFA, keeping everything secure.

**Question**: In a scenario where a user needs different permissions during office hours and non-office hours, how would you configure IAM policies to accommodate this requirement?

**Answer**: For this scenario,

Step 1: I would create two IAM policies – one for office hours and another for non-office hours.

Step 2: Each policy would specify the permissions required during the respective time frame.

Step 3: I would use IAM roles to assign these policies to the user dynamically based on the time of day.

Step 4: By using time-based rules in the policies, I can make sure the user has the right permissions depending on whether it’s work hours or not.

**Question**: How do you implement single sign-on (SSO) with AWS IAM?

**Answer**: Setting up Single Sign-On (SSO) with AWS IAM means using a third-party service that supports SAML (Security Assertion Markup Language) . In the AWS console, I’d set up this service, create a trust connection, and link IAM roles with information from the service. This allows users signed in through the service to easily get into AWS without needing extra AWS logins.

IAM Policies

**Question**: What is an IAM policy, and how does it contribute to access control in AWS?

**Answer**: An IAM policy in AWS is a group of rules, written in JSON format, that defines permissions. It decides what permissions are allowed or denied on what AWS resources.

IAM policies play a very important role in access control by helping you to manage and control access to AWS resources, ensuring that users, groups, and roles have the appropriate permissions for performing specific tasks within your AWS environment.

Question: What are the attributes of IAM policy?

**Answer**:

Version: It specifies the version of the IAM policy language being used.

Statement(s): Contains one or more statements, each defining a set of permissions.

Each statement has attributes like –

Effect (Allow/Deny)

Action (the specific AWS actions)

Resource (the AWS resources affected)

Question: Provide an example of an IAM policy that grants read-only access to all objects in a specific S3 bucket.

Answer:

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": "s3:GetObject",

"Resource": "arn:aws:s3:::specific-bucket/\*"

}

]

}

This policy allows users to perform the s3:GetObject action on any object (\*) within the specified S3 bucket (specific-bucket).

**Question**: You are tasked with allowing a group of users access to a specific DynamoDB table. How would you design the IAM policies to achieve this?

**Answer**:

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"dynamodb:Query",

"dynamodb:GetItem",

"dynamodb:Scan"

],

"Resource": "arn:aws:dynamodb:region:account-id:table/your-dynamodb-table-name"

}

]

}

This policy allows users to query, get items, and scan actions on the specified DynamoDB table (your-dynamodb-table-name).

IAM Roles

**Question**: What is an IAM role?

**Answer**: In AWS, an IAM role is like a set of permissions for doing things with AWS services. It’s not linked to a specific person or group but is used by trusted entities like IAM users or applications. It’s better to connect IAM roles directly to EC2 instances instead of giving credentials to the instance. This is the best way you can use IAM roles without sharing any secret keys with the instance.

**Question**: How are policies attached to IAM roles, and how does this affect the permissions of users assuming those roles?

Policies in AWS IAM are attached to IAM roles to define the permissions that the role assumes. Policies can be either attached directly to the role or included in a managed policy, which is then attached to the role. When a user assumes an IAM role, they inherit the permissions defined by the policies attached to that role. This allows for fine-grained access control, and the permissions of the user are limited to those granted by the role during the period it is assumed.

**Question**: How do you create and manage IAM roles?

**Answer**:

Step 1: Sign in to the AWS Management Console

Step 2: Select “Roles” from the left navigation pane.

Step 3: Click on the “Create role” button.

Step 4: Select Trusted Entities: Choose the trusted entity type, which could be AWS service, another AWS account, or SSO identity provider.

Step 5: Configure Permissions: Attach policies to the role, specifying the permissions it should have.

Step 6: Name the Role: Provide a name and description for the role.

Step 7: Review and Create: Review the role configuration and click “Create role.”

To manage IAM roles, you can:

Modify Policies

Modify Trust Relationships

Rotate Credentials

Advanced IAM Concepts

**Question**: How does IAM evaluate IAM policies to determine whether to grant access to AWS resources?

**Answer**: IAM policies use a process called policy evaluation. When a user or role makes a request, IAM checks all the policies attached to that user or role.

It checks –

Effect (Allow/Deny)

Action (operation)

Resource (target)

The rules are checked in a specific order, and the first clear decision (Allow or Deny) is what’s followed. The final result is a mix of permissions from all the rules they have.

**Question**: How do you monitor and audit access to AWS resources using IAM roles for service accounts?

**Answer**:

Enable AWS **CloudTrail** for tracking role usage and API calls.

Set up AWS Config rules to ensure IAM roles follow security best practices.

Monitor IAM role activity using CloudWatch metrics

Utilize IAM Access Advisor to identify unused or rarely used roles.

Regularly review and align IAM role policies with security requirements.

Automate audits with AWS Config Rules or third-party tools for real-time security alerts.

Conclusion

In conclusion, this AWS IAM Interview Questions is very helpful for both freshers and working professionals who are practicing for interviews or who are training for jobs.

These questions and answers will help you understand the main ideas, best practices, and scenarios-based questions related to IAM, whether you are preparing for an interview at Amazon or just want to learn more about IAM this blog is something for you.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Q4. What are the best practices you will follow while creating IAM users?**

**Answer**. We should always create individual IAM users for each person needing access to AWS services. Even if there are many employees who require the same access, we should create individual IAM users for all of them. This increases the security posture by providing every user of IAM with a unique set of credentials.

**Q5. What are the best practices you would follow while creating any IAM Policy?**

**Answer**. When granting permissions, we should follow the least privileged principle. We should avoid giving users or roles more permissions than they need to accomplish their tasks by following this principle. For example, if an employee needs only access to a specific EC2 instance, specify the instance in the IAM policy. Rather than granting an employee access to every instance in your AWS account.

**Q8. Please explain the IAM Policy Structure.**

**Answer**. We can create IAM policies from the AWS web console and by the visual editor using the JSON-based policy editor. If you take a look at the JSON policy document it basically consists of below elements:

Effect — Decides whether the resource is allowed or denied (Allow/Deny)

Action — A set of service-specific parameters

Resource — Resource names

Condition (Optional) — Grant conditions

**Q9. Define AWS IAM roles.**

**Answer**. An IAM role is a temporary way to access permissions through your identity.

**Q10. What is a Root user?**

**Answer**. The Root User is the Owner Account (administrator) that is created when the AWS Account is created. By default, it has access to all AWS services and resources. It is not possible for IAM Policies to explicitly deny this user access to AWS services or resources.

**Q11. How do you revoke access rights?**

**Answer**. If you need to revoke access rights from an existing user, it’s simple. Simply click on Manage Permissions on his or her profile page and select Revoke Access. You’ll be presented with a list of all services to which they are granted access; check each service that is correct and then click Revoke Access in the bottom right corner.

**Q12. What is MFA in AWS IAM?**

**Answer**. Multi-factor authentication (MFA) adds an extra layer of security for users accessing AWS resources. In addition to a username and password, an MFA-enabled user must provide a one-time code generated by an authenticator app or sent via SMS or voice call before gaining access. An MFA device can be enabled on your computer, phone, or tablet.

**Q13. Is it possible to monitor the activity of IAM users?**

**Answer**. Yes, IAM users’ activities can be monitored. In case of a violation, you can remove the IAM user’s access.

**Q14. What are IAM users’ access keys?**

**Answer**. Each IAM user receives an access key along with a secret key. Users can use their access keys to authenticate themselves to Amazon Web Services when they launch an instance, run a command, or call an API. If you lose your access key, please make sure that you terminate all instances and delete any resources linked to them before creating a new one. If you lose your secret key, we recommend deleting all related resources in order to minimize potential harm.

**Q15. What is Access control to AWS resources?**

**Answer**. The first step in securing your resources is using access control lists (ACLs) to allow or deny access. An AWS account has an owner, so you need an access key and secret key when using ACLs with any service. Make sure you keep these keys safe! The first step in securing your resources is using access control lists (ACLs) to allow or deny access. An AWS account has an owner, so you need an access key and secret key when using ACLs with any service.

**Q17. Explain best practices to manage access to AWS resources.**

**Answer**.

Do not use root accounts – Since root accounts have access to all the AWS resources(Full access) and services, it is not a good idea to share or use them.

Use Groups – Create groups, grant access to them, and add users to them – so that all users within the group have the same access.

Enable Multi-factor Authentication (MFA) – MFA should be enabled for privileged users such as admins. MFA adds an additional layer of security.

Grant least privileges – Only grant permissions that are necessary for the user or group.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**12. What is IAM federation?**

**Answer**: IAM federation allows you to integrate your existing identity system with AWS, enabling users to access AWS resources using their existing credentials. This can be achieved through federation services like AWS Single Sign-On (SSO) or third-party identity providers.

**13. What is the IAM policy evaluation logic?**

**Answer**: IAM policy evaluation follows the "deny by default" principle. If there are no policies explicitly allowing an action, it is denied. Policies can be attached to users, groups, roles, or resources. The most specific policy (with the least privilege) is applied.

**14. How do you create a custom IAM policy?**

**Answer**: To create a custom IAM policy, you can do so through the AWS Management Console, AWS CLI, or AWS SDKs. You write the policy in JSON format, specifying the actions, resources, and conditions. Once created, you can attach it to users, groups, or roles.

**15. What is IAM condition element in a policy?**

**Answer**: Conditions in IAM policies allow you to control when a policy is in effect. They are expressed as key-value pairs, and they can be used to limit access based on various factors such as time, source IP, and more.

**16. How do you rotate access keys for an IAM user?**

**Answer**: You can rotate access keys for an IAM user by creating a new access key, updating applications or services to use the new key, and then deleting the old access key. This ensures a seamless transition without interrupting access.

**17. What is IAM policy versioning?**

**Answer**: IAM policy versioning allows you to have multiple versions of a policy. When you update a policy, AWS creates a new version while keeping the old versions intact. This enables you to maintain backward compatibility and roll back changes if needed.

**18. How can you monitor IAM events and activities?**

**Answer**: You can monitor IAM events and activities by enabling AWS **CloudTrail**, which records all API calls made on your account. CloudTrail logs can be analyzed to track IAM activities and events.

**19. What is AWS Organizations and how does it relate to IAM?**

**Answer**: AWS Organizations is a service that allows you to centrally manage and govern multiple AWS accounts. It helps you consolidate billing, apply policies across accounts, and simplify management. IAM is used within each individual account, while AWS Organizations provides management at the organizational level.

**20. How do you troubleshoot IAM permission issues?**

**Answer**: Troubleshooting IAM permission issues involves checking policies, roles, and group memberships to ensure that the user has the necessary permissions. CloudTrail logs can be reviewed to identify any denied actions and diagnose the issue.