AWS Devops Interview Questions: Fundamentals of IAM

1. What is IAM in AWS?
    - Explain users, groups, roles, and policies.

What is IAM in AWS?

- IAM stands for **Identity and Access Management**.
- It helps you **securely control access** to AWS services and resources.
- You can **create and manage users, groups, roles**, and **permissions**.

👤 Users

- Represent **individual people or applications**.
- Have **long-term credentials** like username/password or access keys.
- Can be assigned **permissions directly** or through groups.

👥 Groups

- A collection of IAM users.
- Used to **assign the same permissions** to multiple users.
- Example: A "Developers" group with access to EC2 and S3.

🎭 Roles

- Do not have credentials by default.
- Used to **delegate access to AWS resources**.
- Can be assumed by **users, applications, EC2, Lambda, or other services**.

- Good for **temporary access**.

Here's a simple explanation of each **Trusted Entity Type** in bullet points:

- **AWS service**
  ➤ Trust services like EC2 or Lambda to act on your behalf.
- **AWS account**
  ➤ Trust another AWS account (yours or third-party) to access your account.
- **Web identity**
  ➤ Trust external users (e.g., from Google, Facebook, Amazon login) via web identity to access your account.
- **SAML 2.0 federation**
  ➤ Trust users from your company's directory (like Active Directory) using SAML-based login.
- **Custom trust policy**
  ➤ Write your own trust rules to control who can access your account.

📜 Policies

- JSON documents that define **what actions are allowed or denied**.
- Can be attached to users, groups, or roles.
- Control access to **specific services, actions, and resources**.
- Types: **AWS-managed, customer-managed, inline**.

What is the difference between IAM user and IAM role?

Feature	IAM User	IAM Role
Identity Type	Specific person/app	Temporary assumed identity
Credentials	Long-term	Temporary
Use Case	Human access, system account	EC2, Lambda, cross-account, federated
Security Best Practice	Limited use, rotate keys	Preferred for automation

IAM Role Itself Is Not Temporary

- The **IAM Role** (its definition) is **permanent**, just like a user.

- You **create a role**, attach policies, and it stays until you delete it.

⚠️ Temporary = Credentials, Not the Role

- When a role is **assumed** (by an EC2, Lambda, or a user), **temporary credentials** are issued by AWS Security Token Service (STS).
- These credentials typically last **15 minutes to a few hours**, depending on how the role is used.
- After expiry, the credentials **must be reissued** by re-assuming the role.

□ Example (S3 Access)

If you give **an EC2 instance a role** that allows access to an S3 bucket:

- The **role is permanent**.
- The EC2 instance **automatically gets temporary credentials** via the Instance Metadata Service.
- These credentials **rotate automatically** behind the scenes every few hours.

So even though the **role persists**, the **access it provides is via short-lived credentials**, which improves security.

✅ Why This Matters

- **No long-term static keys** are stored on machines or in code.
- If credentials are compromised, they **expire quickly**, limiting damage.
- IAM roles + temporary credentials = **secure, automatic, short-term access**.

- What are IAM policies?
  - JSON documents to define permissions; mention types: managed, inline, customer-managed, AWS-managed.

What are IAM groups? Can a user belong to multiple groups?

  - Yes, a user can be in multiple groups; groups help manage permissions collectively.

- Explain the structure of an IAM policy.
  - JSON with fields: `Version`, `Statement`, `Effect`, `Action`, `Resource`, and optional `Condition`.

• What are IAM policies? ◦ • Explain the structure of an IAM policy

What are IAM Policies?

- IAM Policies are **JSON documents** that define **permissions**.
- They control **who can do what** on which **AWS resources**.
- Policies are attached to **users, groups, or roles**.

🧩 **Types of IAM Policies:**

- **AWS-managed**: Predefined by AWS (e.g., `AmazonS3ReadOnlyAccess`).
- **Customer-managed**: Created and managed by you.
- **Inline**: Directly embedded into a single user, group, or role (used for one-off, specific cases).

Structure of an IAM Policy

A policy is a JSON document with the following key fields:

```
json
CopyEdit
{
  "Version": "2012-10-17",           // Policy language version
  "Statement": [
    {
      "Effect": "Allow",             // Can be "Allow" or "Deny"
      "Action": "s3:GetObject",      // The operation (e.g., S3 read)
      "Resource": "arn:aws:s3:::my-bucket/*"  // The resource ARN
    }
  ]
}
```

Field Breakdown:

- **Version**: Always `"2012-10-17"` (latest version format).
- **Statement**: A list of permission blocks.
  - **Effect**: `Allow` or `Deny`.
  - **Action**: Specific AWS service actions (like `s3:GetObject`).
  - **Resource**: ARN of the resource the action applies to.
  - **Condition** *(optional)*: Adds extra logic (e.g., IP address, time, MFA).

1. What does IAM stand for in AWS?

A) Internet Access Manager
B) Internal Account Management
C) Identity and Access Management
D) Infrastructure Access Mechanism

✅ **Answer:** C) Identity and Access Management

---

2. Which of the following statements is true about IAM Users?

A) IAM users represent groups of AWS accounts.
B) IAM users have temporary credentials.
C) IAM users are used to manage EC2 only.
D) IAM users can have long-term credentials like access keys.

✅ **Answer:** D) IAM users can have long-term credentials like access keys.

---

3. What is the purpose of IAM Groups in AWS?

A) To manage S3 buckets
B) To assign the same permissions to multiple users
C) To create temporary credentials
D) To run Lambda functions

✅ **Answer:** B) To assign the same permissions to multiple users

---

4. Which statement is true about IAM Roles?

A) IAM Roles have long-term passwords.
B) IAM Roles are only for human users.
C) IAM Roles are used to delegate temporary access.
D) IAM Roles are used only within IAM Groups.

✅ **Answer:** C) IAM Roles are used to delegate temporary access.

---

5. What are IAM Policies in AWS?

A) Rules to stop billing in AWS
B) JSON documents that define permissions

C) Templates to launch EC2 instances
D) Monitoring dashboards

✅ **Answer:** B) JSON documents that define permissions