**AWS IAM (Identity and Access Management) – Complete Notes**

**1. What is IAM?**

* **IAM (Identity and Access Management)** is a service that helps you securely manage access to AWS services and resources.
* It allows you to create users, groups, and roles and define permissions to control access.

**2. IAM User vs IAM Role**

* **IAM User**: A permanent identity with access credentials like username, password, and access keys.
* **IAM Role**: A temporary identity used by trusted entities (e.g., EC2, Lambda) to access AWS resources. It provides temporary credentials.

**3. What is an IAM Policy?**

* An IAM policy is a document that defines permissions to perform actions on resources.
* **JSON Format**: Policies are written in JSON format.
* **Key components**: Action, Resource, Effect (Allow/Deny).

**4. Inline Policy vs Managed Policy**

* **Inline Policy**: Attached directly to a user, group, or role. Not reusable across multiple entities.
* **Managed Policy**: Standalone policies that can be reused and attached to multiple entities. AWS provides **AWS Managed Policies** and users can create **Customer Managed Policies**.

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

* **MFA (Multi-Factor Authentication)** adds an extra layer of security by requiring a second form of authentication (e.g., a code from a mobile app like Google Authenticator) along with the password.

**6. AWS STS (Security Token Service)**

* **STS** allows you to generate temporary security credentials for IAM roles or federated users.
* Used for granting temporary access to AWS resources for users or services.

**7. Difference between Identity-Based and Resource-Based Policies**

* **Identity-Based Policies**: Attached to IAM users, groups, or roles to define permissions for what actions they can perform on resources.
* **Resource-Based Policies**: Attached to resources (e.g., S3 buckets, Lambda functions) to define who can access the resource and what actions they can perform.

**8. Role-Based Access Control (RBAC)**

* RBAC is a method of controlling access based on the role a user or entity performs. Permissions are granted based on the job function, e.g., Developer, Tester, Admin.

**9. What is an IAM Policy Simulator?**

* It is a tool that helps test and troubleshoot IAM policies by simulating the permissions of a user or role. It shows if a particular action is allowed or denied.

**10. Allow vs Deny in IAM Policies**

* **Allow** grants permission to perform an action.
* **Deny** explicitly blocks access, even if there is an **Allow** in another policy.
* **Deny** takes precedence over **Allow**.

**11. What is AWS IAM Federation?**

* Federation allows users from external systems (e.g., Active Directory, Google) to access AWS resources by using temporary credentials. It integrates with **AWS STS** for temporary access.

**12. What is an IAM Group?**

* A collection of IAM users. Groups make it easier to manage permissions for multiple users at once. Policies can be applied to the group, and all users in the group inherit those permissions.

**13. Best Practices for Using IAM**

* **Enable MFA** on the root account and IAM users.
* **Follow the Principle of Least Privilege** – only give users the permissions they absolutely need.
* Use **IAM Roles** for applications and services instead of access keys.
* Regularly **review IAM permissions** and policies.

**14. What is an IAM Trust Policy?**

* A trust policy defines which entities (users, groups, or AWS services) can assume a particular IAM role.

**Conclusion & Key Takeaways:**

* **IAM** is a critical service for managing access and permissions securely in AWS.
* You should always adhere to the **Principle of Least Privilege** and **use roles for temporary access**.
* **MFA** is a simple but highly effective way to improve security.
* **Testing policies** using the IAM **Policy Simulator** is essential to avoid permission-related issues.