SESSION 3

IAM
(Identity And Access Management)

IAM is a free service.

Administrator adds the users, the groups, authorises the users for the resources. It is up to an Administrator to allow access and to regulate it. In AWS, the IAM service does this for you. Root user is an only Administrator at the launch of your AWS account.

IAM Users and Groups

Root Account: This account is created by default by the user when the sign up is done. It should not be shared with anyone.

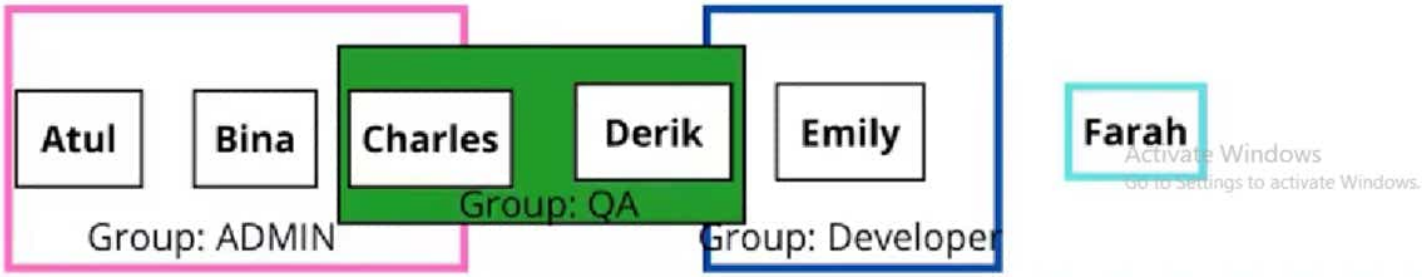
Users: are people within your organisation and can be grouped together. These are physical people.

An IAM user has a name and password that they use to log in to the AWS management console. You should not use the root account. You should create a new IAM user with the required permissions to access the AWS Management console.

Groups: only contains users, not other groups.

users don't have to belong to a group

Users can belong to multiple groups.



IAM Permissions AND Policies

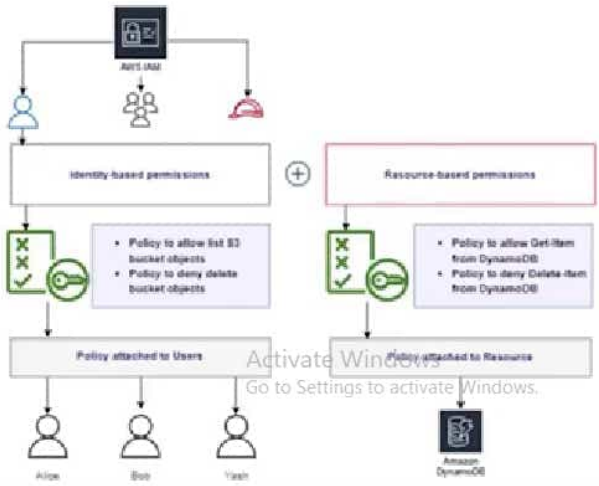
Policy: A policy is a JSON document with information about permissions. IAM policies are associated with the AWS users, groups, and roles.

JSON: JSON is a text format for storing and transporting data. JSON is "self describing" and easy to understand. Eg: '{"name":"John", "age":30, "car":null}'

By default IAM entities (users, groups, and roles) start with no permissions. You need to follow the "Least privilege principle" as a security best practices. grant only the required permissions to complete a task

Identity-based policies: Identity-based policies are attached to AWS identities like user, group, or role.

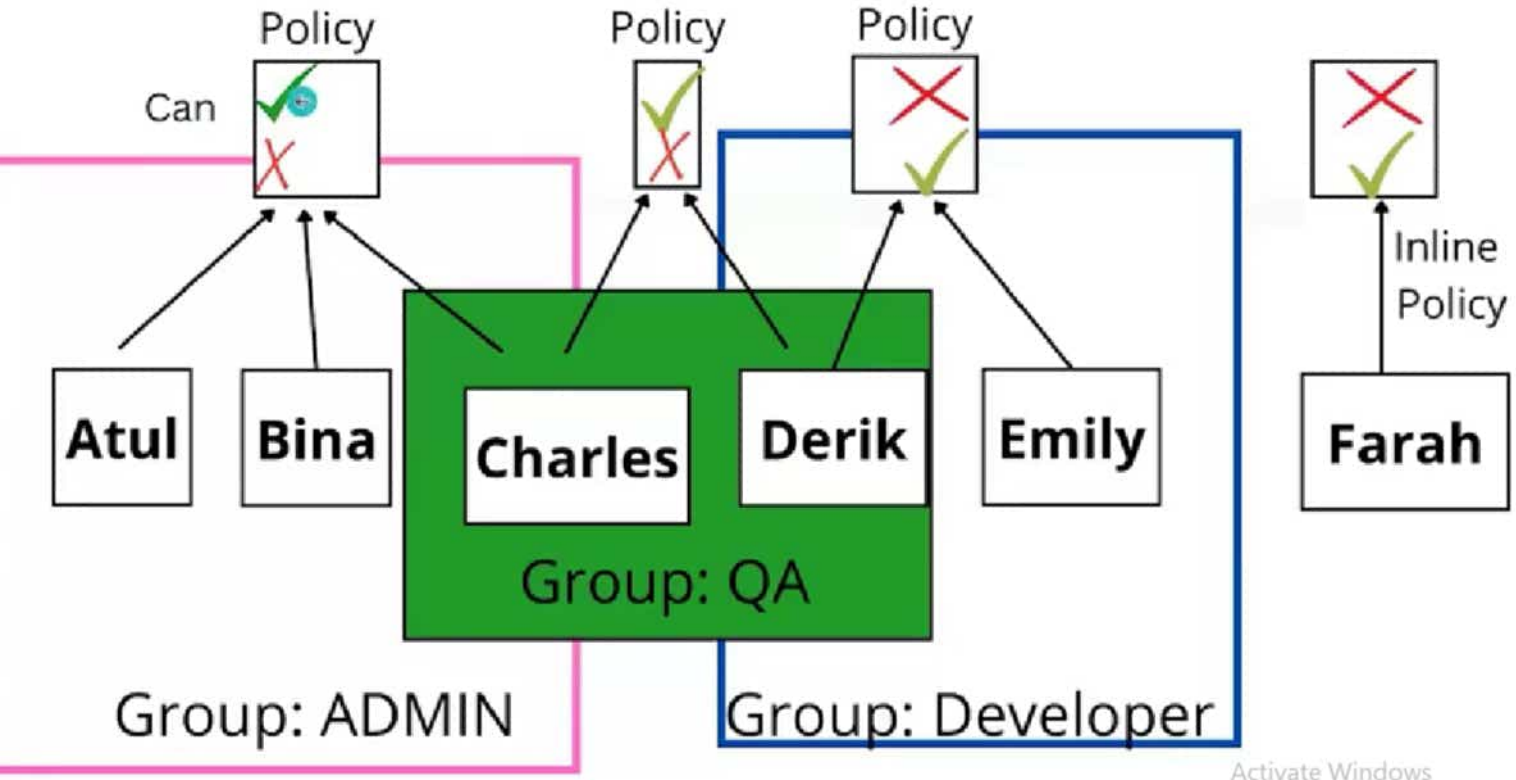
Resource-based policies: Resource-based policies are attached to the AWS resources



IAM Policy Consists of

- **Version:** Policy language version
- **ID:** an identifier of the policy(optional)
- **Statement:** one or more individual statement
    - **Sid:** an identifier for the statement(optional)
    - **Effect:** Whether the statement allows or deny access(Allow, Deny)
    - **Principal:** account/user/role to which this policy allows/denies. Who can access it
    - Action: List of actions this policy allows/ denies
    - **Resource:** List of resources to which this policy allows/denies
    - **Condition:** for when this policy is in effect(optional)

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "iam:AttachUserPolicy",
            "Resource": "arn:aws:iam::*:user/*"
        }
    ]
}
```



IAM Roles

Some AWS services will need to perform actions on your behalf. To do so we will assign permissions to AWS services with IAM Roles.
An IAM entity that defines set of permissions for making requests to AWS services, and will be used by an AWS services.
eg. EC2 instance role, Lambda Function Roles

- **IAM Password Policy: Strong password=High Security**
- **IAM MFA(MultiFactor Authentication)= Password you know+Security device you own**
- **Access and Secrete Key**
- **IAM Security Tools:** IAM Credential Repots: list of all account users and their status.
  IAM Access Advisor: Shows permissions assigned to a user.

Canva Sans - 34 + Effects Animate Position

IAM Questions

1. Name 2 IAM Security tools.
2. Which of the following is INCORRECT about IAM Users
   - IAM users can belong to multiple Groups
   - IAM Users don't have to belong to any group
   - IAM Policy can be attached directly to IAM User
   - IAM Users access AWS services using root user credential
3. What is IAM best practice?
4. Which principle would you apply regarding IAM permissions?
5. How can you increase your root account security?
6. IAM User Groups can contain IAM Users and other User Groups? True/False
7. A statement in an IAM policy does not consists of:
   Effect Principal Version Action Resource

1.Credential Report, Access Advisor 2. d 3. Don't use root user account 4. Least Privilege principle 5. MFA 6. False 7. Version