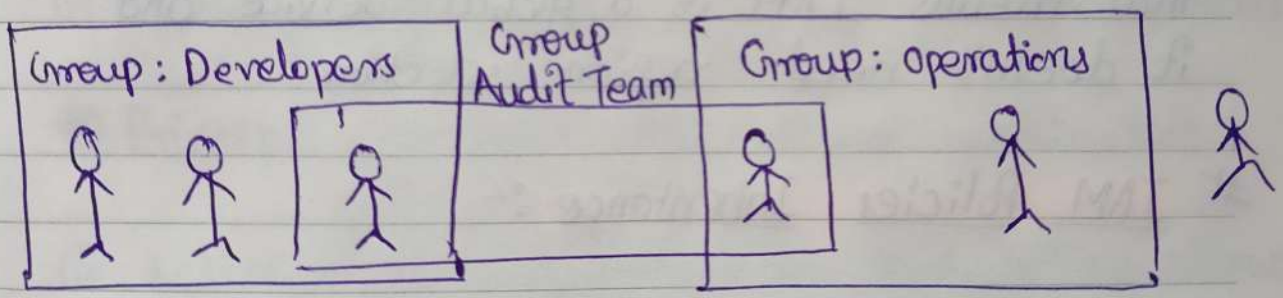
IAM : Users & Groups

→ IAM : Identity & Access Management, Global Service

→ Root account created by default, shouldn't be used or shared.
→ Users are people within your organization, & can be grouped.
→ Groups only contain users, not other groups
→ Users don't have to belong to a group, & user can belong to multiple groups.



IAM : Permissions

→ Users or Groups can be assigned JSON documents called policies
→ These policies define the permissions of the users
→ In AWS you apply the least priviledge principle: don't give more permissions than a user needs.

* When you go to AWS console ~~you~~ and select IAM you will see the ~~re~~gion is Global that means IAM is a global service and it doesn't need region selection.

IAM Policies Inheritance :-

* users in same group follows the same group policy
→ we can attach policy to single user as well if that user is not part of any group.

IAM Policies Structure

* Consists of :-
  1. Version :- policy lang version, always include "2012-10-17"
  2. Id : an identifier for the policy (optional)
  3. Statement : one or more individual statements (required)

* Statement Consists of :
  1. Sid :- identifier for the statement (optional)
  2. Effect :- whgre the statement allow or denies access (Allow, Deny)
  3. Principal :- account / user / role to which this policy applied to
  4. Action :- List of actions of this policy allows or denies
  5. Resource :- List of resources to which the action applied to
  6. Condition :- Conditions for when this policy is in effect (optional)

How can users access AWS?

→ Options :-

① AWS management Console (Protected by Pass + MFA)
② AWS Cmd line interface (CLI): Protected by access keys
③ AWS Software Developer Kit (SDK) - for code: protected by access keys

* Access keys are generated through the AWS Console
* Users manage their own access keys.
* Access keys are secret, just like pass. Don't share them.
* Access key ID ≅ username
* ~~Acc~~ Secrete access key ≅ password

What's the AWS CLI?

→ A tool that enables you to interact with AWS services using commands in your cmd-line shell
→ Direct access to the public APIs of AWS services
→ You can develop scripts to manage your resources
→ Its open-source https://github.com/aws/aws-cli

Whats the AWS SDK?

→ AWS Software Development kit (AWS SDK)
→ Language-specific APIs (set of libraries)
→ Enables you to access & manage AWS service programmatically
→ Embedded within your application
→ Supports
  * SDKs (JS, Python, PHP, .Net, Ruby, Java, Go, Node.js
  * Mobile SDKs (Android, ios...)
  * IOT devices SDKs (Embedded, C, Audino,..)

ex:- AWS CLI is built on AWS SDK for Python

Shared Responsibility Model for IAM

AWS	Me / You / User
* Infrastructure (Global network security)	* Users, Groups, Roles, Policies management & monitoring
* Configuration & Vulnerability analysis	* Enable MFA on all acc
* Compliance Validation	* Rotate all your keys often
	* Use IAM tools to apply appropriate permissions
	* Analyse access patterns & review permissions.

~~Imp~~ Imp

IAM Section - Summary

① Users : mapped to physical user, has a password for aws console
② Groups : contains users only
③ Policies : JSON document that outlines permissions for users or groups.
④ Roles : for EC2 instances or AWS services (Lambda)
⑤ Security : MFA + Password Policy
⑥ AWS CLI : manage your AWS services using the command-line
⑦ AWS SDK : manage your AWS services using a programming language.
⑧ Access keys : access AWS using the CLI or SDK.
⑨ Audit : IAM Credential Reports & IAM Access Advisor