AWS FUNDAMENTALS

A Region is a physical location in the world that consists of two or more Availability Zones (AZs).

An AZ is one or more discrete data centers – each with redundant power, networking, and connectivity – housed in separate facilities.

Edge locations are end points for AWS that are used for caching content. Typically, this consists of CloudFront, Amazon's CDN.

Shared Responsibility

This defines who is responsible for what if something happens. In general, ask yourself this,

Can you do this yourself in the AWS Management Console?

- If answer is yes, you are likely responsible.
- If answer is no, AWS is likely responsible.
- Encryption is shared responsibility (since both you & AWS have ability to encrypt)

Key Services to know for the Exam

- Compute – EC2, Lambda, Elastic Beanstalk
- Storage – S3, EBS, EFS, FSx, Storage Gateway
- Databases – RDS, DynamoDB, Redshift
- Networking – VPCs, Direct Connect, Route 53, API Gateway, AWS Global Accelerator

Well-Architected Framework

- https://aws.amazon.com/whitepapers to read whitepapers.
- Read 'AWS Well-Architected Framework' whitepaper a day/night before taking the exam by checking 'Well-Architected Framework' filter in the website.

IDENTITY ACCESS MANAGEMENT

It allows us to

- Create users & grant permissions to those users.
- Create groups & roles.
- Control access to AWS resources.

The **root account** is the email address you used to sign up for AWS. The root account has full administrative access to AWS. For this reason, it is important to secure this account.

4 steps to secure AWS Root Account

- Enable MFA on the root account.

- Create an admin group for your administrators and assign the appropriate permissions to this group.
- Create user accounts for your administrators.
- Add users to the admin group.

Control Permissions using IAM

We assign permissions using policy documents, which are made up of JSON.

Example for admin policy

```
{
    "Version":"2012-10-17"
    "Statement":[
        {
            "Effect":"Allow",
            "Action":"*",
            "Resource":"*"
        }
        ]
}
```

Building blocks of IAM:

- Users – A physical person
- Groups – Functions, such as admins, developers etc. contains users.
- Roles – Internal usage within AWS

Best practices

- It's best practice for users to inherit permissions from group (instead of directly providing permissions to users, create group -> create policy and assign to group -> add users to that group).
- Always work on the principle that one user equals one physical person. Never share user accounts across multiple people.
- The principle of least privilege- only assign a user the minimum amount of privileges they need to do their job.
- By default, when we create a user every time, that user has no permissions, has no privileges.

Points to Remember

- IAM is universal: It does not apply to specific regions at this time.
- The root account: The account created when you first setup your AWS account and which has complete access. Secure it ASAP and do not use it to login day to day.
- New users: No permissions when first created.

- Access key ID and secret access keys are not same as usernames & passwords. Access Key Id and secret access keys are used in API & consoles.
- You only get to view access key id and secret access key once. If you lose them, you have to regenerate them. So, save them in a secure location.
- Always setup password rotations. You can create & customize your own password rotation policies.
- IAM Federation: you can combine your existing user account with AWS. For example, when you logon to your PC (usually using Microsoft active directory), you can use the same credentials to login to AWS if you setup federation.
- Identity Federation: Uses the SAML standard, which is Active Directory.
- When we 'Deny' in Effect in any policy, then it is known as explicit deny and an explicit deny will always override any allow when that policy is applied to any user.