AWS:

CLOUD CONSOLE, CLI & SDK

How can users access AWS ?

- To access AWS, you have three options:
  - AWS Management Console (protected by password + MFA)
  - AWS Command Line Interface (CLI): protected by access keys
  - 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
- <u>Access Keys are secret, just like a password. Don’t share them</u>
- Access Key ID ~= username
- Secret Access Key ~= password

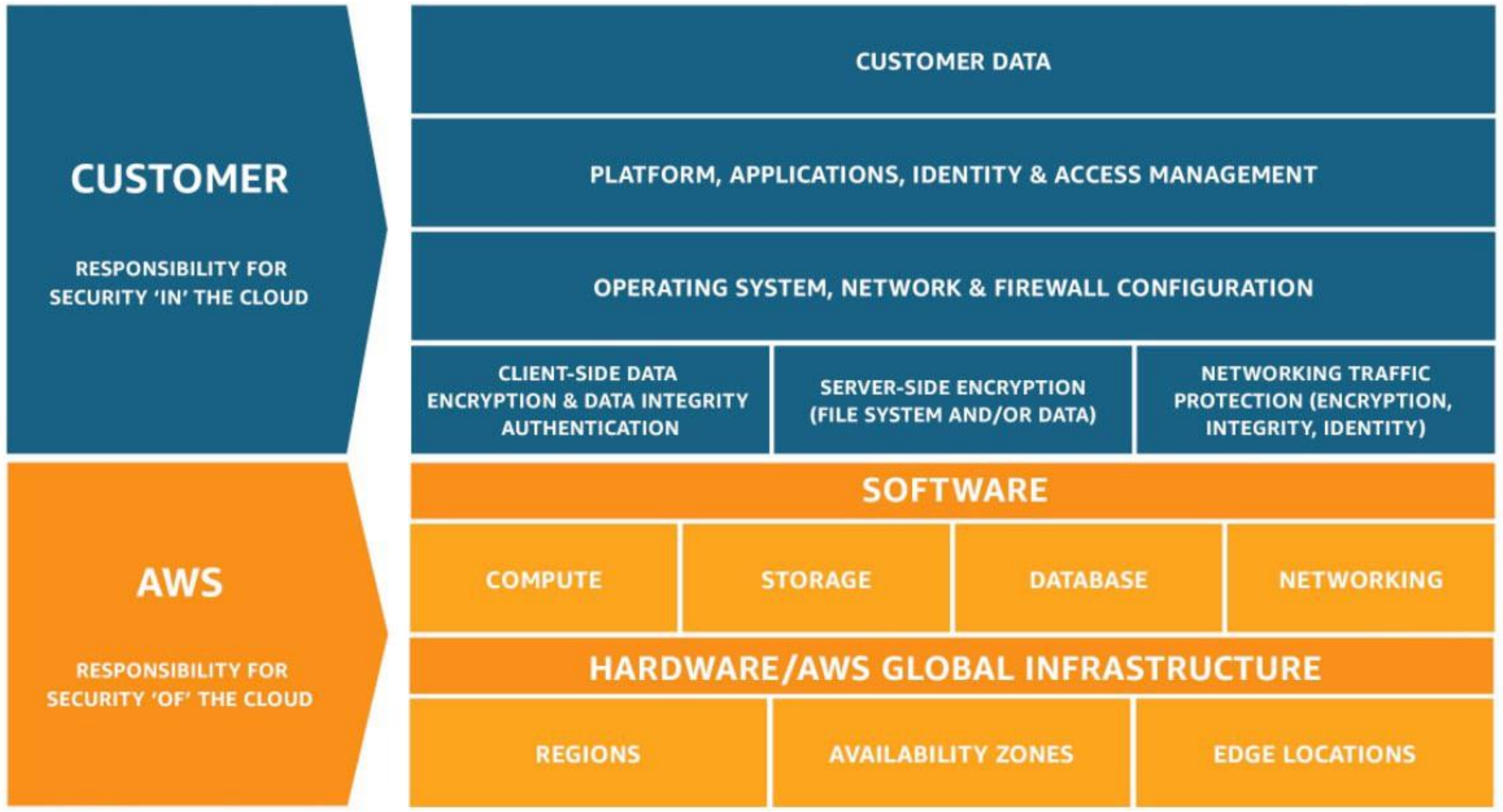
Shared Responsibility Model diagram

CUSTOMER = RESPONSIBILITY FOR

THE SECURITY **IN** THE CLOUD

AWS = RESPONSIBILITY FOR

THE SECURITY **OF** THE CLOUD


CUSTOMER
RESPONSIBILITY FOR SECURITY 'IN' THE CLOUD
CUSTOMER DATA
PLATFORM, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT
OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION
CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION
SERVER-SIDE ENCRYPTION (FILE SYSTEM AND/OR DATA)
NETWORKING TRAFFIC PROTECTION (ENCRYPTION, INTEGRITY, IDENTITY)
AWS
RESPONSIBILITY FOR SECURITY 'OF' THE CLOUD
SOFTWARE
COMPUTE
STORAGE
DATABASE
NETWORKING
HARDWARE/AWS GLOBAL INFRASTRUCTURE
REGIONS
AVAILABILITY ZONES
EDGE LOCATIONS

https://aws.amazon.com/compliance/shared-responsibility-model/

AWS: Console

- **Advantages:**
  - **User-Friendly Interface:** The AWS Console provides a graphical user interface (GUI) that is intuitive and easy to navigate, making it accessible to users with limited technical expertise.
  - **Visual Representation:** It offers visual representations of AWS resources, making it easier to understand and manage complex configurations.
  - **Quick Start:** For beginners, it's a quick and easy way to start working with AWS services without the need for installation or scripting.
  - **Built-in Documentation:** The Console often provides helpful tooltips and links to AWS documentation, aiding users in understanding services and features.
- **Disadvantages:**
  - **Limited Automation:** It's not ideal for automating tasks or managing resources at scale, as it requires manual interactions.
  - **Slower Workflow:** For experienced users, the graphical interface can be slower compared to CLI or SDKs when performing repetitive tasks.
  - **Lack of Scripting:** Advanced automation and scripting capabilities are limited within the Console.

Tour of the AWS Console

- **AWS has Global Services:**
  - Identity and Access Management (IAM)
  - Route 53 (DNS service)
  - CloudFront (Content Delivery Network)
  - WAF (Web Application Firewall
- **Most AWS services are Region-scoped:**
  - Amazon EC2 (Infrastructure as a Service)
  - Elastic Beanstalk (Platform as a Service)
  - Lambda (Function as a Service)
  - Rekognition (Software as a Service)
- Region Table: https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services

AWS: Command Line Interface (CLI_)

- **Advantages:**
  - **Scripting and Automation:** The AWS CLI allows for scripting and automation of AWS tasks, which is essential for managing resources at scale.
  - **Efficiency:** It's often faster for experienced users who prefer command-line interfaces and can perform tasks with fewer keystrokes.
  - **Integration:** CLI commands can easily be integrated into scripts, DevOps pipelines, and scheduled tasks.
  - **Wide Range of Features:** The CLI provides access to the full range of AWS services and their features.
  - **Cross-Platform:** It's available on various operating systems, making it versatile for different environments.
- **Disadvantages:**
  - **Learning Curve:** It can be challenging for beginners or those unfamiliar with command-line interfaces.
  - **No Visual Feedback:** There's no visual representation of resources, which may make it harder to visualize complex configurations.
  - **Potential for Syntax Errors:** Typing errors can lead to unintended actions, potentially affecting resources.

AWS: Software Development Kit (SDK)

- **Advantages:**
  - **Programmatic Control:** SDKs allow developers to interact with AWS services programmatically, making it suitable for custom applications.
  - **Extensive Language Support:** AWS provides SDKs for various programming languages, enhancing flexibility for developers.
  - **Full AWS Service Coverage:** SDKs offer comprehensive access to AWS services and their features, giving developers fine-grained control.
  - **Integration:** They can be integrated into existing applications and workflows seamlessly.
  - 5. **Error Handling:** SDKs often include error-handling mechanisms to manage issues gracefully.
- **Disadvantages:**
  - **Development Effort:** Developing and maintaining custom applications using SDKs may require significant development effort and expertise.
  - **Complexity:** SDK usage can be complex for non-developers and require knowledge of programming languages.
  - **Version Management:** Developers must stay up to date with AWS SDK version changes and updates.

AWS: Hands On Session

- **AWS Console:**
  - Take a tour of the console
  - Create an account or Sign into an existing account.
  - Launch a Linux Ec2 Instance with Apache (HTTPD) server Installed on it.
- **AWS CLI:**
  - Download the AWS CLI. Depending on your operating system, it will require a different method.
  - Windows Prerequisites:
    - You must be running Microsoft Windows XP or later.
  - **Installation Linux x86 (64-bit):**
    - Link https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html
    - Create an IAM user and give the user an administrative access.
    - Create Secret Access Key
    - Launch a Linux Ec2 Instance with Apache (HTTPD) server Installed on it

AWS: Hands On Session

- **AWS SDK:**
  - SDKs Link: https://aws.amazon.com/developer/tools/
    - Boto3 Installation Link: https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html
    - Create an IAM user and give the user an administrative access.
  - Launch a Linux Ec2 Instance with Apache (HTTPD) server Installed on It using AWS SDK (boto3)
    - Start/ stop / terminate our instance using AWS SDK (boto3).

Thanks
Merci
Gracias