# AWS Identity & Access

Cloud Computing

Brenden west

# Contents

*Learning Outcomes*

- AWS Shared Responsibility model
- AWS Identity & Access Management
- AWS account security

Reading

- AWS Cloud Foundations - module 4
- AWS Cloud Developing - modules 3, 4

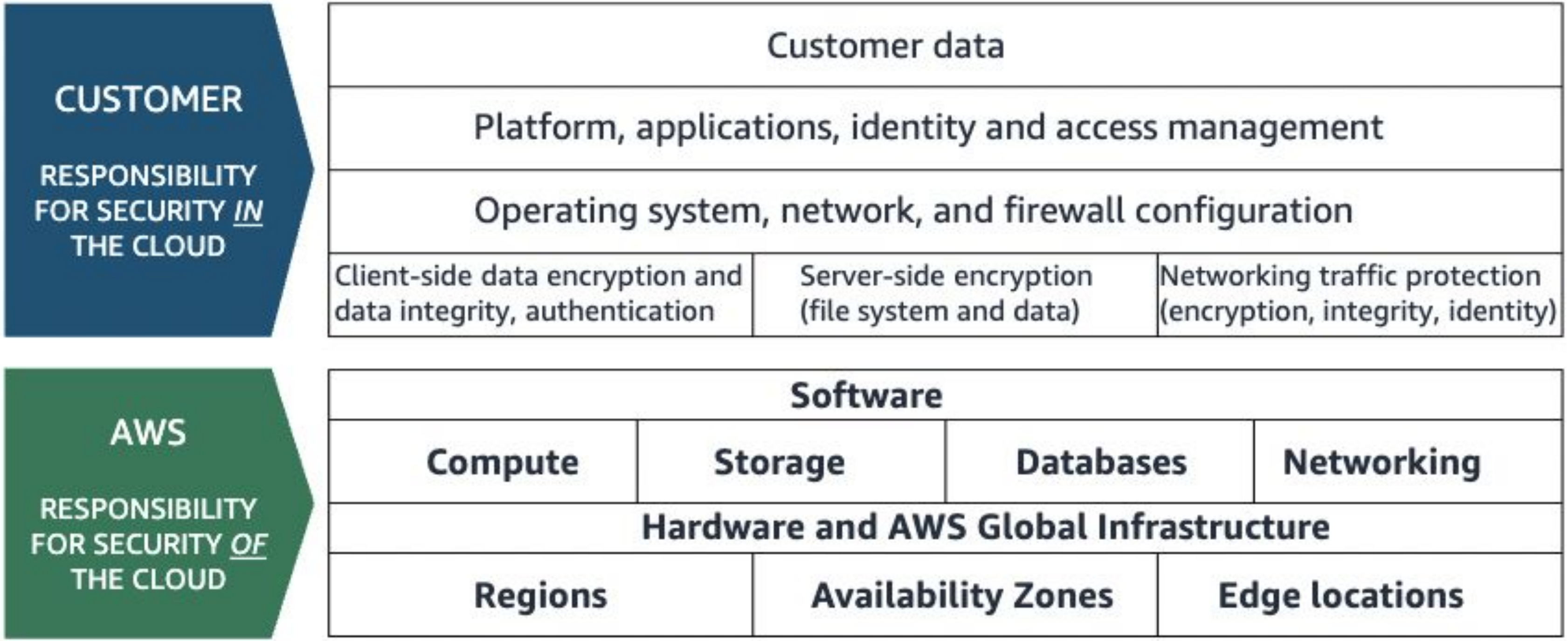
# Terminology

- **IAM -** Identity & Access Management
- **Authentication** - mechanism to verify the identity of a user
- **MFA** - multi-factor authentication uses two or more mechanisms to authenticate a user
- **Authorization** - mechanism to verify a user has permission to access a service or resource
-

# AWS Shared Responsibility Model

- AWS & the customer share responsibility for security & compliance
- AWS responsible for security **of** the cloud (physical infrastructure)
- Customer responsible for security **in** the cloud
- Customers responsible for what is implemented using AWS products & services. E.g:
  - Managing their data
  - Configuring appropriate security using IAM & other security services
  - Guest operating systems on virtual machines
  - Firewalls & network configurations

# AWS Shared Responsibility, cont.



CUSTOMER

RESPONSIBILITY FOR SECURITY *IN* THE CLOUD

Customer data

Platform, applications, identity and access management

Operating system, network, and firewall configuration

| Client-side data encryption and data integrity, authentication | Server-side encryption (file system and data) | Networking traffic protection (encryption, integrity, identity) |
|---|---|---|

AWS

RESPONSIBILITY FOR SECURITY *OF* THE CLOUD

**Software**

| **Compute** | **Storage** | **Databases** | **Networking** |
|---|---|---|---|

**Hardware and AWS Global Infrastructure**

| **Regions** | **Availability Zones** | **Edge locations** |
|---|---|---|

# AWS Identity & Access Management (IAM)

Allows AWS customer to grant unique security credentials to users, roles, and groups.

- Securely controls who can access customer’s AWS resources, what resources they can use, and in what ways
- Integrates with other AWS services
- Supports granular permissions
- Supports federated identity management (via corporate identity providers)
- Supports multi-factor authentication (MFA)

# IAM Overview

- **IAM user**  - a person or application with permanent credentials to access the services & resources in an AWS account
- **IAM group** - a collection of IAM users with same permissions. Group members inherit permissions attached to the group.
- **IAM role** - an AWS identity with attached permission policies. Does not have long-term credentials
- **IAM policy** - a document that lists explicit permissions. Can be attached to an IAM user, IAM group, or IAM role

**Best practice** - attach IAM policies to IAM groups and then assign IAM users to these groups.

# Authenticating with IAM

- Any interaction with AWS services, whether through management console, AWS CLI, or AWS SDK, requires authentication by providing credentials
- Management console authentication depends on **user name** and **password**
- CLI, SDKs, and APIs depend on **AWS access keys** (access key and secret key)
- Users or services that assume an IAM role are provided with temporary security credentials to use in accessing AWS resources

# AWS Credentials File

The AWS CLI client depends on credentials stored in a local text file to interact with AWS accounts.

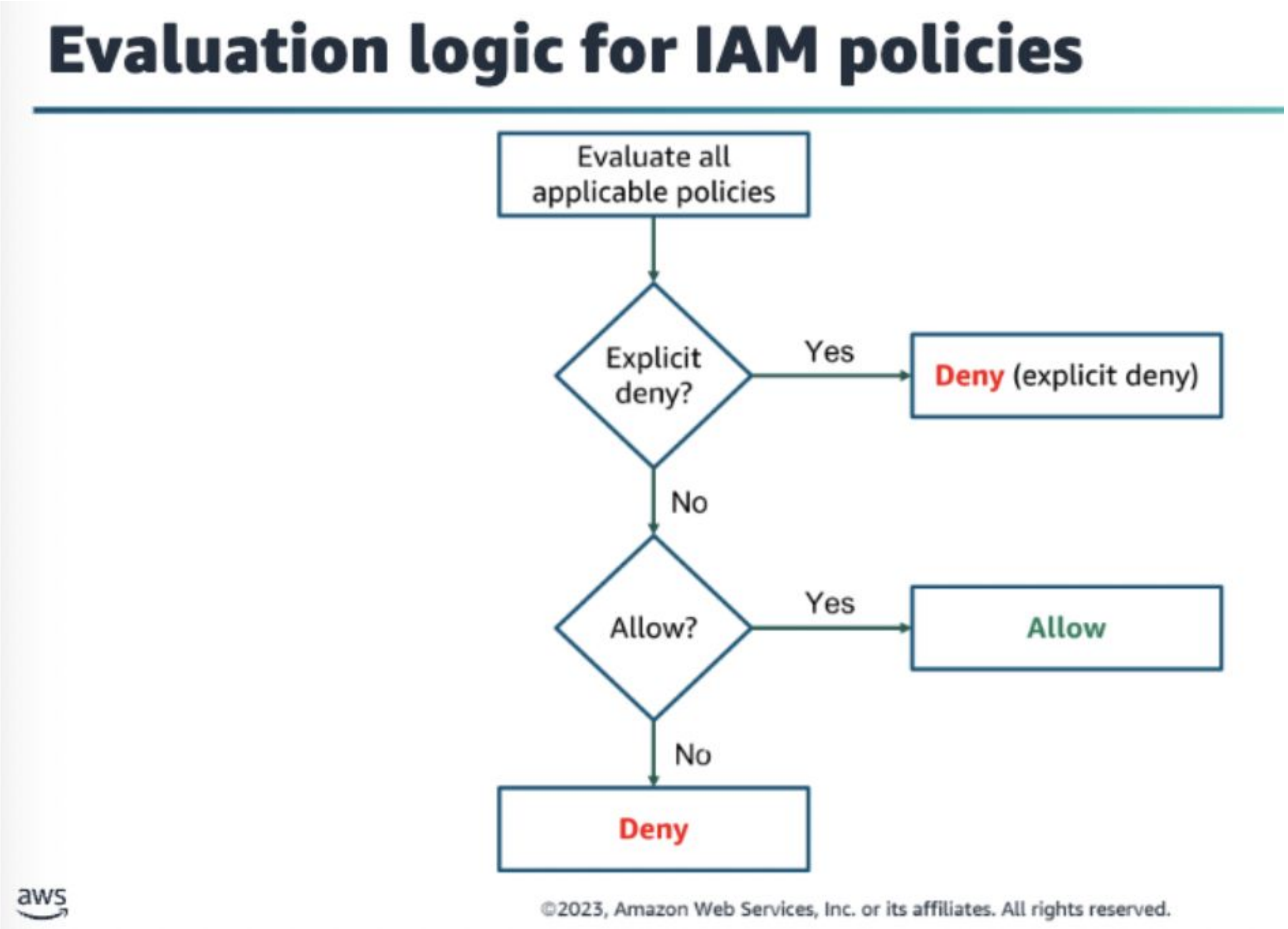
- File location is **~/.aws/credentials** (Unix, Linux, MacOS) or **c:\Users\<USERNAME>\.aws\credentials** (Windows)
- credentials file can be used by multiple projects
- credentials file can contain keys (**profiles**) for multiple AWS accounts or environments
- Credentials should not be stored in code or publicly accessible locations

# Authorizing with IAM

- By default, an authenticated user, group, or role has no access permissions
- Permissions to access AWS resources are controlled through IAM policies
- IAM policy is a JSON document that defines effect, action, resources, and optional conditions under which an entity can invoke API operations in an AWS account
- Any actions or resources not explicitly allowed are denied
- Actions may include wildcards (asterisks) to cover a set of related actions

# Principle of Least Privilege

- Grant only permissions needed to perform a task
- Start with minimum set of permissions and grant additional permissions as needed
- Use account root user to create one or more IAM users
- Use IAM users for ongoing account access and management tasks



# Evaluation logic for IAM policies

Evaluate all applicable policies

Explicit deny?

Yes

Deny (explicit deny)

No

Allow?

Yes

Allow

No

Deny

# IAM Policy Types

**Identity-based policy**

- Attached to an IAM user, group, or role
- Specifies what an identity can do

**Resource-based policy**

- Attached to a resource
- Specifies what a user or group is permitted to do with the resource

# IAM Policies

**Managed policy**

- Standalone, identity-based for attaching to multiple users, groups, and roles
- Provide reusability, central charge management, versioning, rollback, and ability to delegate permissions management

**Inline policy**

- Embedded in an entity
- If used for multiple entities, each has its own copy