Identity and Access Management

- Overview
- Principals
- Authentication
- Authorization
- Multi-Factor Authentication
- Key Rotation
- Multiple Permissions
- AWS Compliance
- Shared Responsibility

IAM Overview

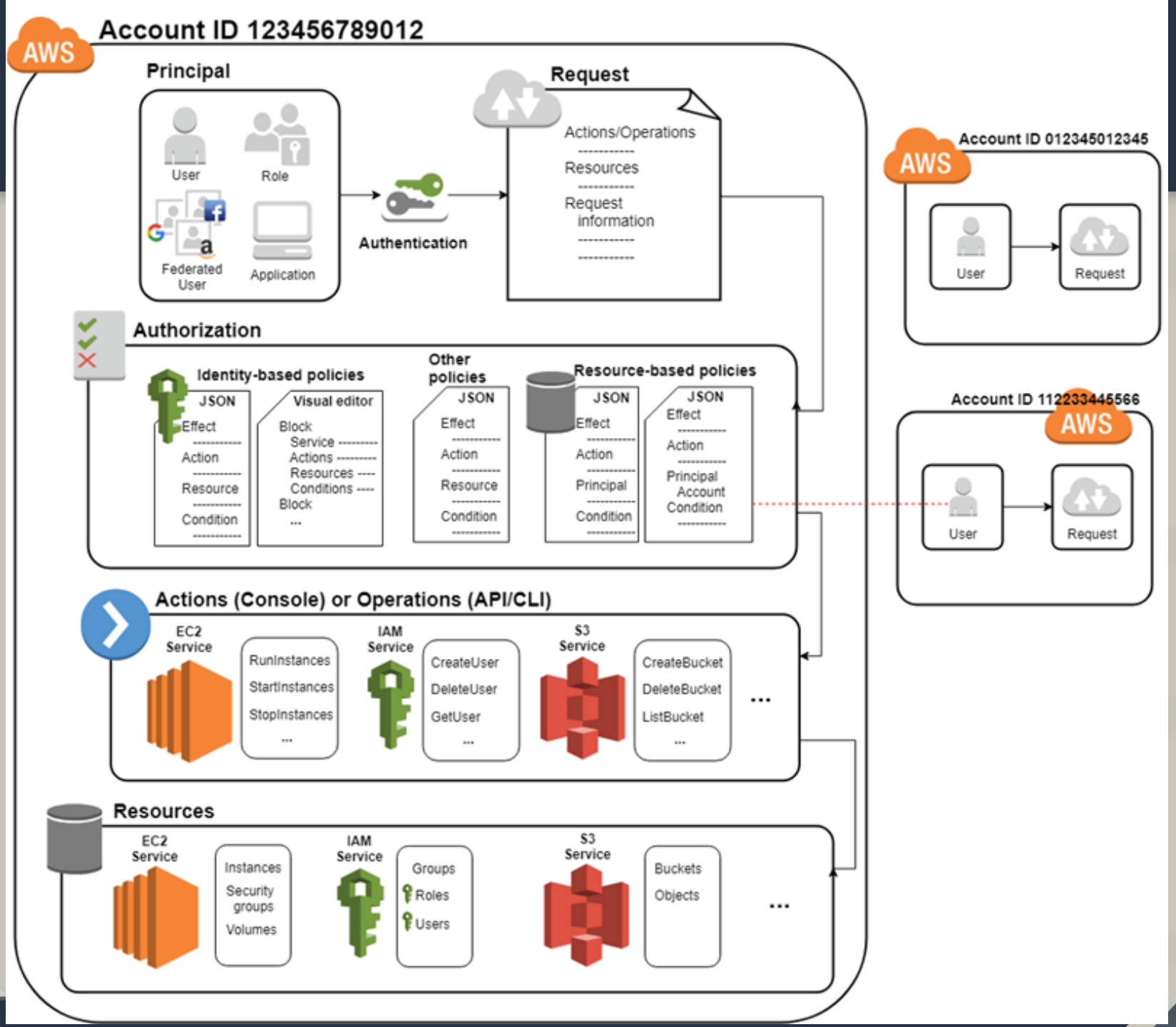
EPISODE 7.01

Identity and Access Management

- Manage access to AWS
  - Doesn't manage OS, services, or applications
- Supports users, groups, and roles
- Free
- AWS services implemented by the users incur charges

IAM Concepts

- Resources
- Principals
  - Users
  - Groups
  - Roles
- Policies


Account ID 123456789012
Principal
User
Role
Federated User
Application
Authentication
Request
Actions/Operations
Resources
Request information
Authorization
Identity-based policies
JSON
Effect
Action
Resource
Condition
Visual editor
Block
Service
Actions
Resources
Conditions
Block
Other policies
JSON
Effect
Action
Resource
Condition
Resource-based policies
JSON
Effect
Action
Principal
Condition
JSON
Effect
Action
Principal
Account
Condition
Actions (Console) or Operations (API/CLI)
EC2 Service
RunInstances
StartInstances
StopInstances
...
IAM Service
CreateUser
DeleteUser
GetUser
...
S3 Service
CreateBucket
DeleteBucket
ListBucket
...
...
Resources
EC2 Service
Instances
Security groups
Volumes
IAM Service
Groups
Roles
Users
S3 Service
Buckets
Objects
...
Account ID 012345012345
User
Request
Account ID 112233445566
User
Request

Principals

EPISODE 7.02

Principals

- Also called identities
- Entity that can perform an action
  - Users
  - Groups
  - Roles

Users

- IAM users are entities created in AWS
- Person or service with permissions:
  - AWS Management Console
  - AWS API/CLI

Users

- User credentials
  - Consists of a name and password and up to two access keys
    - Access keys are used with the API or CLI
- Users can be members of groups

Groups

- A collection of IAM users
- Permissions should be managed at the group level
- Users can be added and removed
- Groups are not used to log in

Roles

- An identity granted permissions
- Roles aren’t permanently assigned
- Assumable by any entity with a need for it

Roles

- Compatible with federated users
  - Users from other identity provider systems
  - Mapped to the role
  - Allows for SSO (single sign-on)

Users vs. Roles

- Create user accounts when:
  - You're the only person working with the account
  - Multiple people need permanent access
  - One or more users require CLI access

Users vs. Roles

- Create roles when:
  - Applications need access to an AWS service
  - Mobile phone apps make requests of AWS
  - Existing company users need federated access

Root User

EPISODE 7.03

AWS Root User

- Email address used to create the AWS subscription
- Unlimited capabilities
- Not recommended for everyday access
- Create an IAM admin user and safely store the root user account

Root Access Tasks

- Modifying the root user
- Changing the AWS support plan
- Closing an AWS account

Root Access Tasks

- Creating a CloudFront key pair
- Enabling Multi-Factor Authentication (MFA) on an S3 bucket
- Restore permissions for other IAM users

Authentication

EPISODE 7.04

Authentication

- Validation of credentials
- Credentials provide identity
- Single-factor
- Multi-factor

Authentication

- Authentication of persons
- Authentication of processes

Authentication in AWS

- Required to manage AWS
- S3 allows anonymous access

Authentication in AWS

- User name and password
  - Console
- Access key and secret key
  - API
  - CLI

Authorization Policies

EPISODE 7.05

Policies

- Rules that determine allowed actions or access
- Used throughout AWS
- Uses JSON
  - Created by GUI
  - Coded directly
- Vary by object

Authorization

- Validation of actions
- Provided by AWS policies

Authorization

- Identity-based policies
  - Used with users, groups, or roles
- Resource-based policies
  - Used for cross-account access (accounts from different AWS subscriptions)

Policy Processing

- By default, all requests are denied
- Explicit allow overrides the default
- Permission boundaries can override explicit allows
- Explicit denies override explicit allows

Actions or Operations

- Request is authenticated
  - Action or operation is processed
- Request is authorized
  - Linked to a service

Actions or Operations

- Process against a resource
- Includes CRUD:
  - Create (launch)
  - Read (view)
  - Update (edit)
  - Delete (terminate)

DEMO

- Actions, Resources and Condition Keys
- https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_actions-resources-contextkeys.html

Multi-Factor Authentication

EPISODE 7.06

Multi-Factor Authentication (MFA)

AWS MFA

- Best practice
- Couples user name and password with another factor
  - Something you know
  - Something you have
  - Something you are
  - Something you receive
- Can be enabled for the root account and users

DEMO

- MFA Form Factors
- https://aws.amazon.com/iam/details/mfa/

Key Rotation

EPISODE 7.07

Key Rotation

- Best practices suggest rotating keys
  - Access key ID
  - Secret access key
- Key rotation only applies to user accounts

Key Rotation Process

1. Create a second access key in addition to the one in use
2. Update all your applications to use the new access key and validate that the applications are working
3. Change the state of the previous access key to inactive
4. Validate that your applications are still working as expected
5. Delete the inactive access key

Key Listing

```
aws iam list-access-keys --user-name Alice
```

```
{
        "AccessKeyMetadata": [
         {
                "UserName": "Alice",
                "Status": "Active",
                "CreateDate": "2013-04-03T18:49:57Z",
                "AccessKeyId": "AKIAI44QH8DHBEXAMPLE“
        }
        ]
}
```

Key Creation

```
aws iam create-access-key --user-name Alice
```

```
{
        "AccessKey": {
                "UserName": "Alice",
                "Status": "Active",
                "CreateDate": "2013-09-06T17:11:57Z",
                "SecretAccessKey":"wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",
                "AccessKeyId": "AKIAIOSFODNN7EXAMPLE“
        }
        ]
}
```

Multiple Permissions

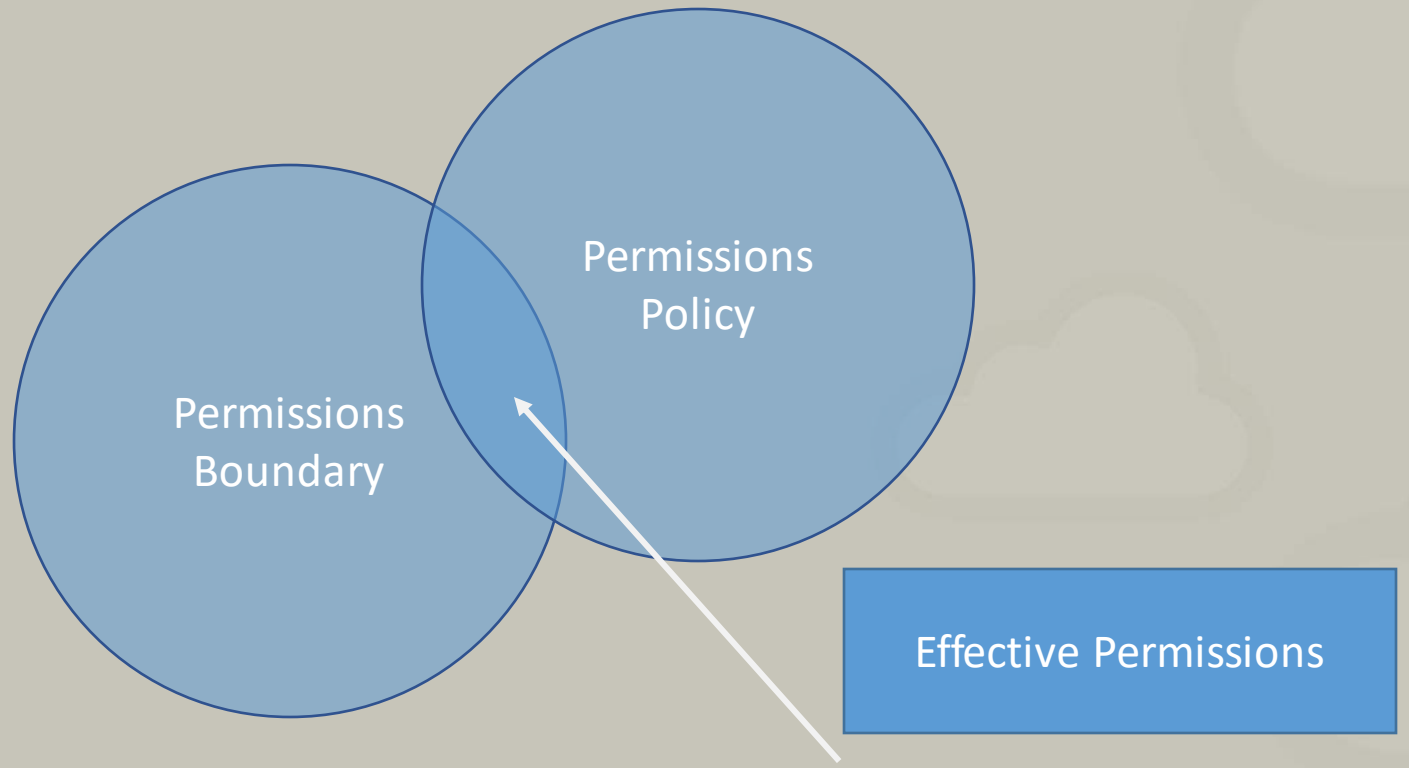
EPISODE 7.08

Multiple Permissions

- Users
- Groups
- Boundaries

Permission Boundaries

- Constrain permissions a user can receive
  - Limit a used to specific services



Example Boundary Policy

```
{
        "Version": "2012-10-17",
        "Statement": [
                {
                "Effect": "Allow",
                "Action": [
                        "s3:*",
                        "cloudwatch:*",
                        "ec2:*“
                ],
                "Resource": "*“
                }
        ]
}
```

Example Boundary Policy

```
{
        "Version": "2012-10-17",
        "Statement": [
                {
                "Effect": "Allow",
                "Action": [
                        "s3:*",
                        "cloudwatch:*",
                        "ec2:*“
                ],
                "Resource": "*“
                }
        ]
}
```

AWS Compliance Program

EPISODE 7.09

DEMO

- AWS Compliance Program
- aws.amazon.com/compliance

Shared Responsibility Model

EPISODE 7.10

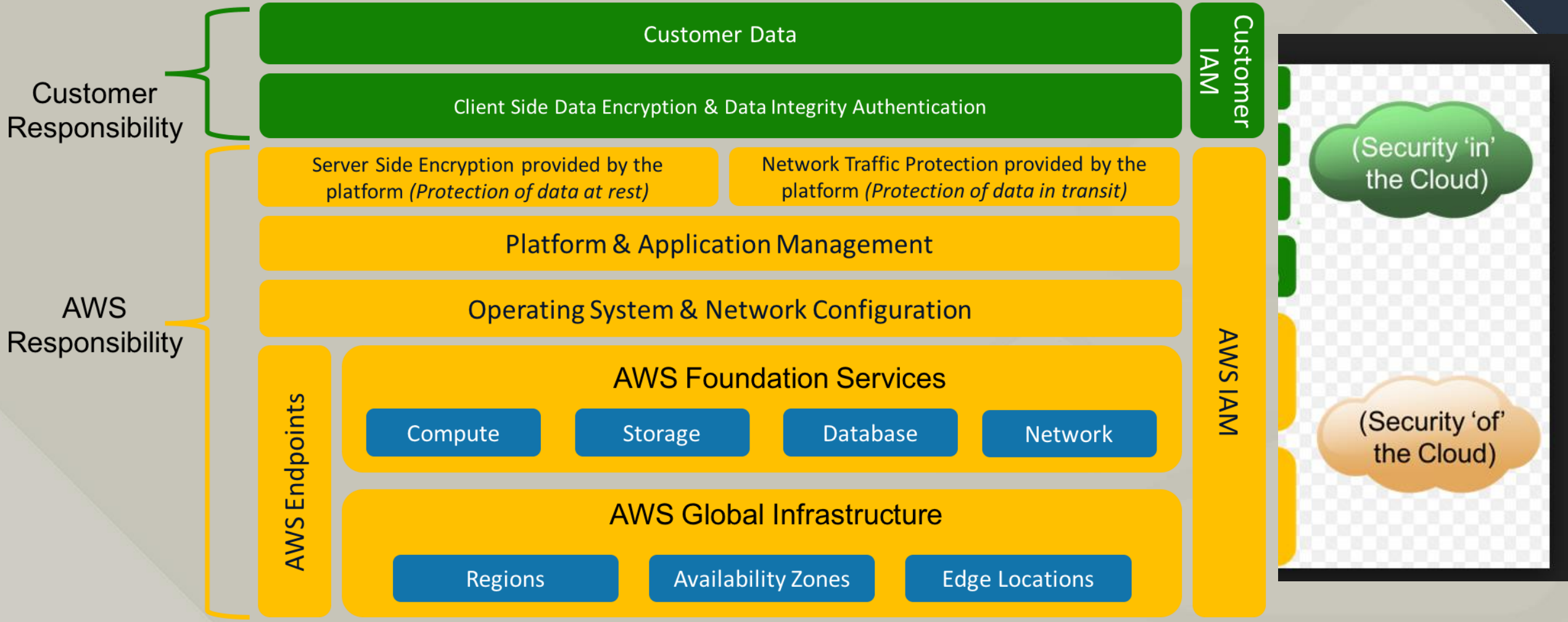
Shared Responsibility

- AWS provides security of the cloud
  - Physical
  - Network
  - Hypervisor
  - Managed services (DynamoDB, Redshift, etc.)

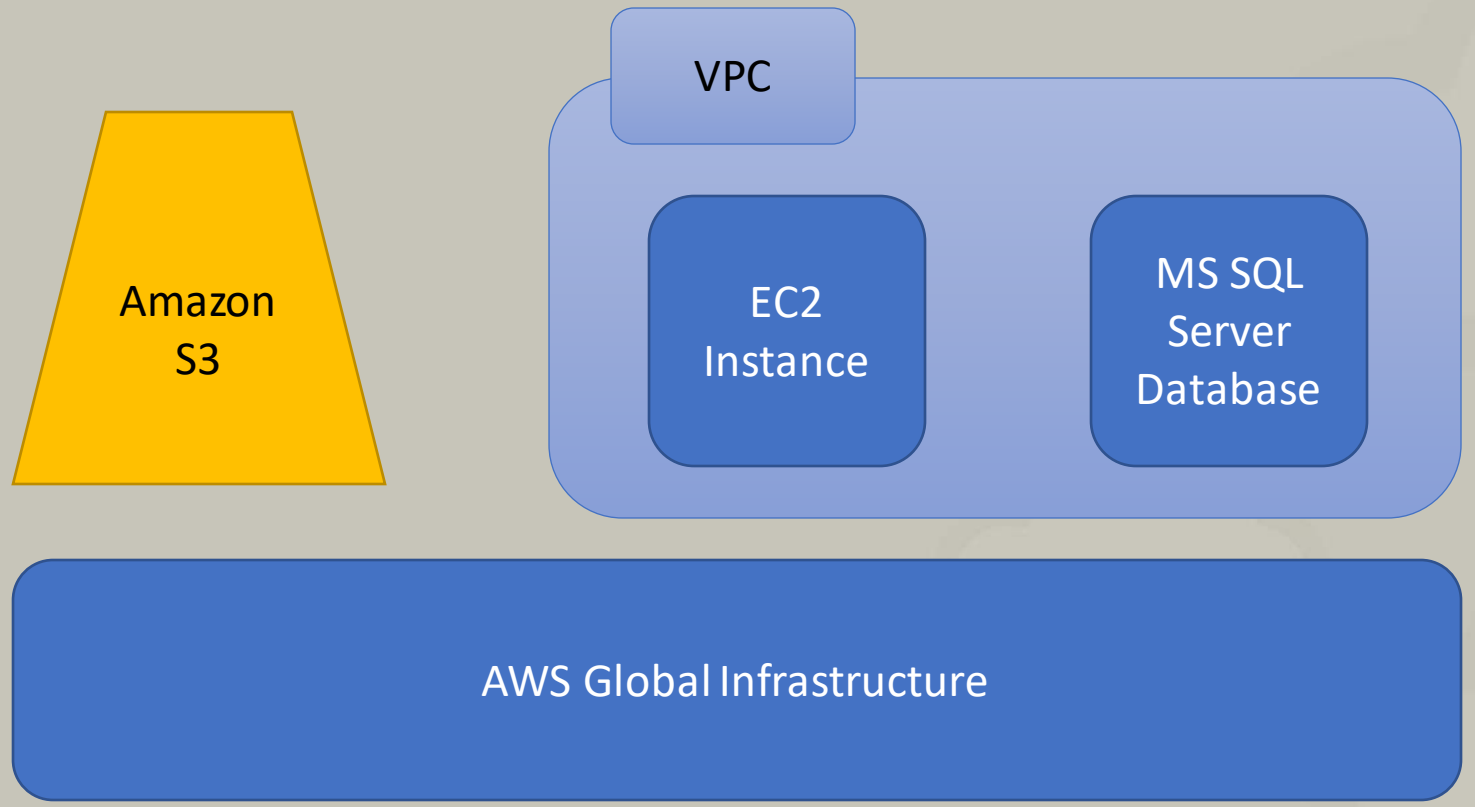
Shared Responsibility

- You provide security in the cloud
  - Guest OS
  - Application
  - User Data

Shared Responsibility Model



Shared Responsibility Example



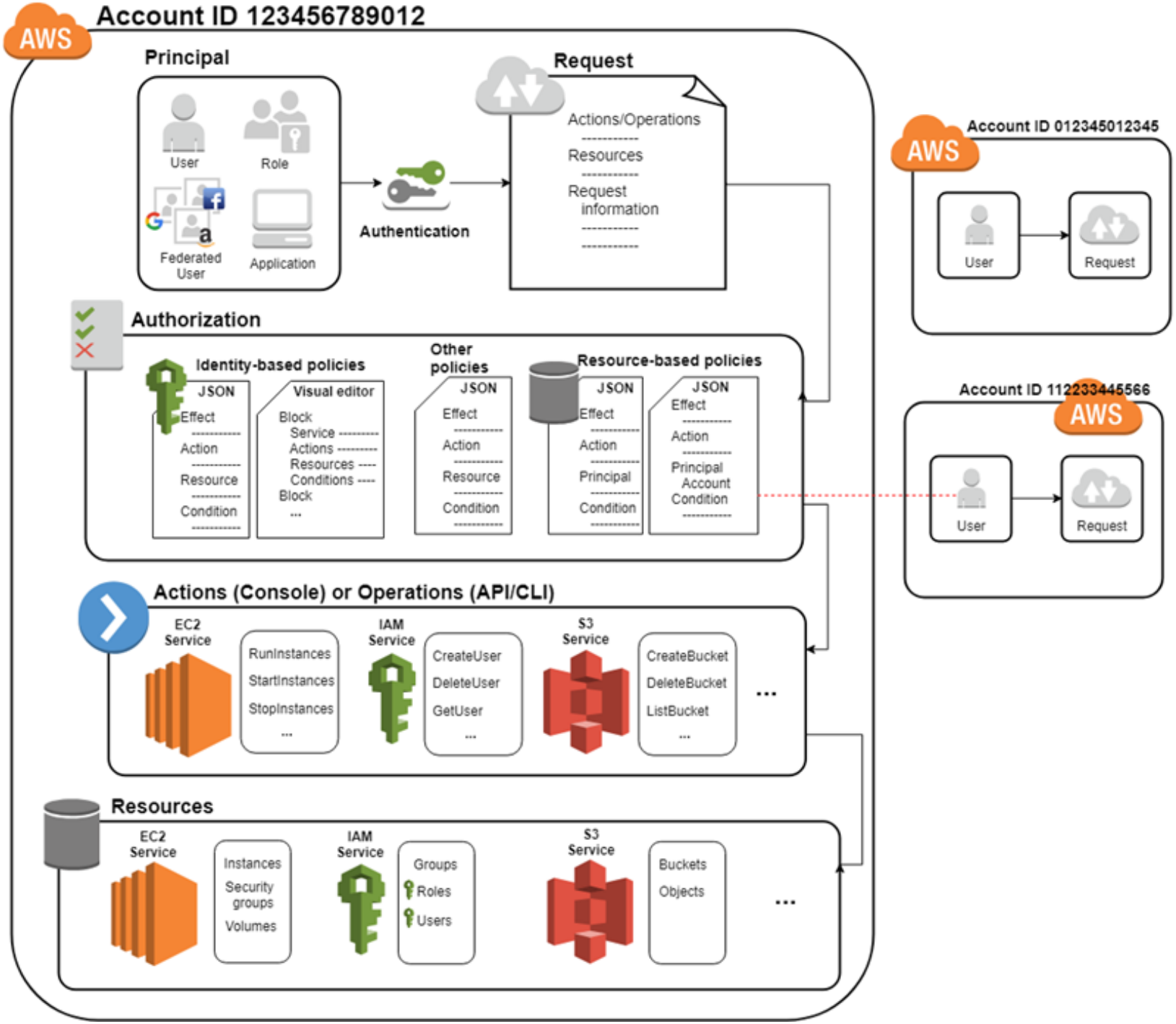
- EPISODE 7.01
- IAM Overview

Identity and Access Management

- Manage access to AWS
  - Doesn't manage OS, services, or applications
- Supports users, groups, and roles
- Free
- AWS services implemented by the users incur charges

IAM Concepts

- Resources
- Principals
  - Users
  - Groups
  - Roles
- Policies


Account ID 123456789012
AWS
Principal
User
Role
Federated User
Application
Authentication
Request
Actions/Operations
Resources
Request information
Authorization
Identity-based policies
JSON
Effect
Action
Resource
Condition
Visual editor
Block
Service
Actions
Resources
Conditions
Block
Other policies
JSON
Effect
Action
Resource
Condition
Resource-based policies
JSON
Effect
Action
Principal
Condition
JSON
Effect
Action
Principal
Account
Condition
Actions (Console) or Operations (API/CLI)
EC2 Service
RunInstances
StartInstances
StopInstances
...
IAM Service
CreateUser
DeleteUser
GetUser
...
S3 Service
CreateBucket
DeleteBucket
ListBucket
...
...
Resources
EC2 Service
Instances
Security groups
Volumes
IAM Service
Groups
Roles
Users
S3 Service
Buckets
Objects
...
Account ID 012345012345
AWS
User
Request
Account ID 112233445566
AWS
User
Request

- EPISODE 7.02
- Principals

Principals

- Also called identities
- Entity that can perform an action
  - Users
  - Groups
  - Roles

Users

- IAM users are entities created in AWS
- Person or service with permissions:
  - AWS Management Console
  - AWS API/CLI

Users

- User credentials
  - Consists of a name and password and up to two access keys
    - Access keys are used with the API or CLI
- Users can be members of groups

Groups

- A collection of IAM users
- Permissions should be managed at the group level
- Users can be added and removed
- Groups are not used to log in

Roles

- An identity granted permissions
- Roles aren’t permanently assigned
- Assumable by any entity with a need for it

Roles

- Compatible with federated users
  - Users from other identity provider systems
  - Mapped to the role
  - Allows for SSO (single sign-on)

Users vs. Roles

- Create user accounts when:
  - You're the only person working with the account
  - Multiple people need permanent access
  - One or more users require CLI access

Users vs. Roles

- Create roles when:
  - Applications need access to an AWS service
  - Mobile phone apps make requests of AWS
  - Existing company users need federated access

AWS Account Root

- Email address used to create the AWS subscription
- Unlimited capabilities
- Not recommended for everyday access
- Create an IAM admin user and safely store the account root

Root Access Tasks

- Modifying the root user
- Changing the AWS support plan
- Closing an AWS account

Root Access Tasks

- Creating a CloudFront key pair
- Enabling Multi-Factor Authentication (MFA) on an S3 bucket
- Restore permissions for other IAM users

- EPISODE 7.03
- Authentication

Authentication Defined

- Validation of credentials
- Credentials provide identity
- Single-factor
- Multi-factor

Authentication Defined

- Authentication of persons
- Authentication of processes

Authentication in AWS

- Required to manage AWS
- S3 allows anonymous access

Authentication in AWS

- User name and password
  - Console
- Access key and secret key
  - API
  - CLI

- EPISODE 7.04
- Authorization Policies

Policies

- Rules that determine allowed actions or access
- Used throughout AWS
- Uses JSON
  - Created by GUI
  - Coded directly
- Vary by object

Authorization

- Validation of actions
- Provided by AWS policies

Authorization

- Identity-based policies
  - Used with users, groups, or roles
- Resource-based policies
  - Used for cross-account access (accounts from different AWS subscriptions)

Policy Processing

- By default, all requests are denied
- Explicit allow overrides the default
- Permission boundaries can override explicit allows
- Explicit denies override explicit allows

Actions or Operations

- Request is authenticated
  - Action or operation is processed
- Request is authorized
  - Linked to a service

Actions or Operations

- Process against a resource
- Includes CRUD:
  - Create (launch)
  - Read (view)
  - Update (edit)
  - Delete (terminate)

DEMO

- Actions, Resources and Condition Keys
- https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_actions-resources-contextkeys.html

- EPISODE 7.05
- Multi-Factor Authentication

Multi-Factor Authentication (MFA)

AWS MFA

- Best practice
- Couples user name and password with another factor
  - Something you know
  - Something you have
  - Something you are
  - Something you receive
- Can be enabled for the root account and users

DEMO

- MFA Form Factors
- https://aws.amazon.com/iam/details/mfa/

- EPISODE 7.06
- Key Rotation

Key Rotation

- Best practices suggest rotating keys
  - Access key ID
  - Secret access key
- Key rotation only applies to user accounts

Key Rotation Process

- Create a second access key in addition to the one in use
- Update all your applications to use the new access key and validate that the applications are working
- Change the state of the previous access key to inactive
- Validate that your applications are still working as expected
- Delete the inactive access key

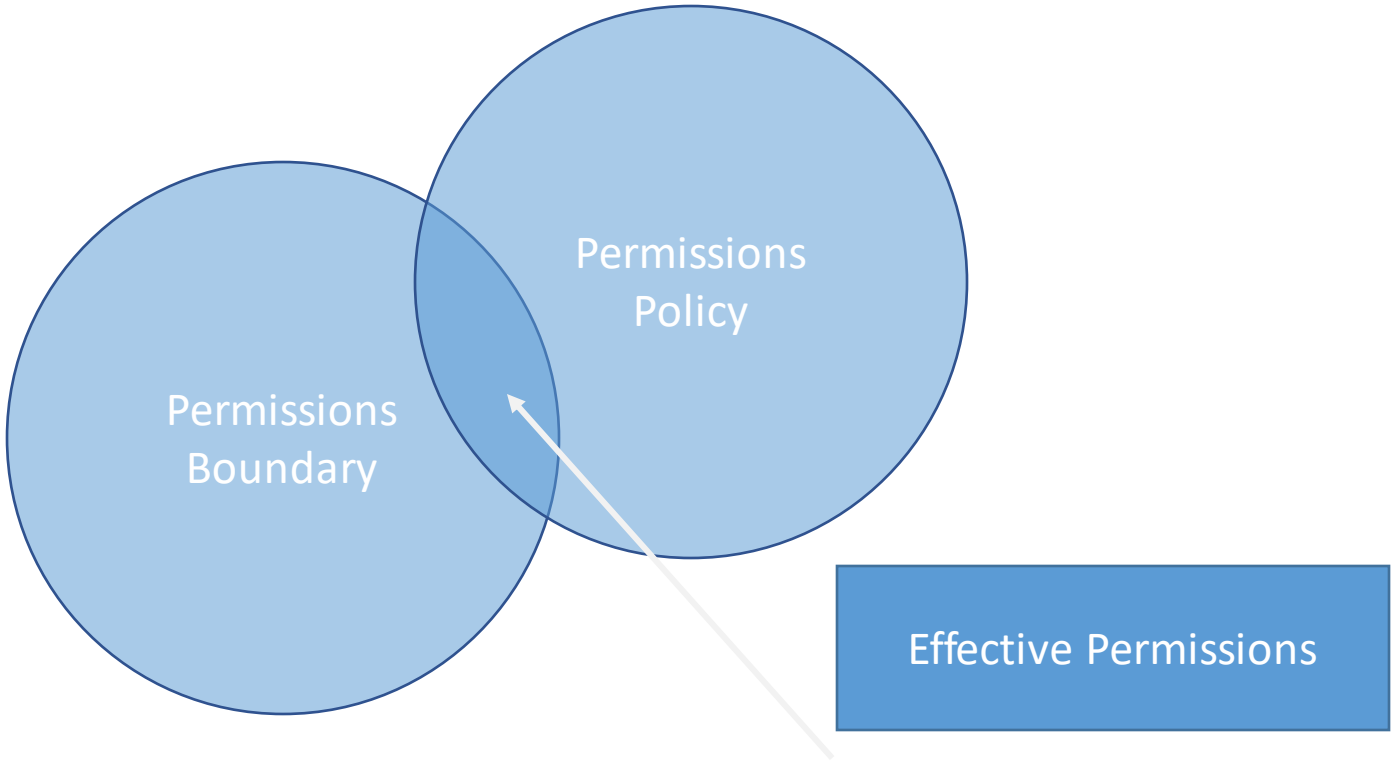
- EPISODE 7.07
- Multiple Permissions

Multiple Permissions

- Users
- Groups
- Boundaries

Permission Boundaries

- Constrain permissions a user can receive
  - Limit a used to specific services



- EPISODE 7.08
- AWS Compliance Program

DEMO

- AWS Compliance Program
- aws.amazon.com/compliance

- EPISODE 7.09
- Shared Responsibility Model

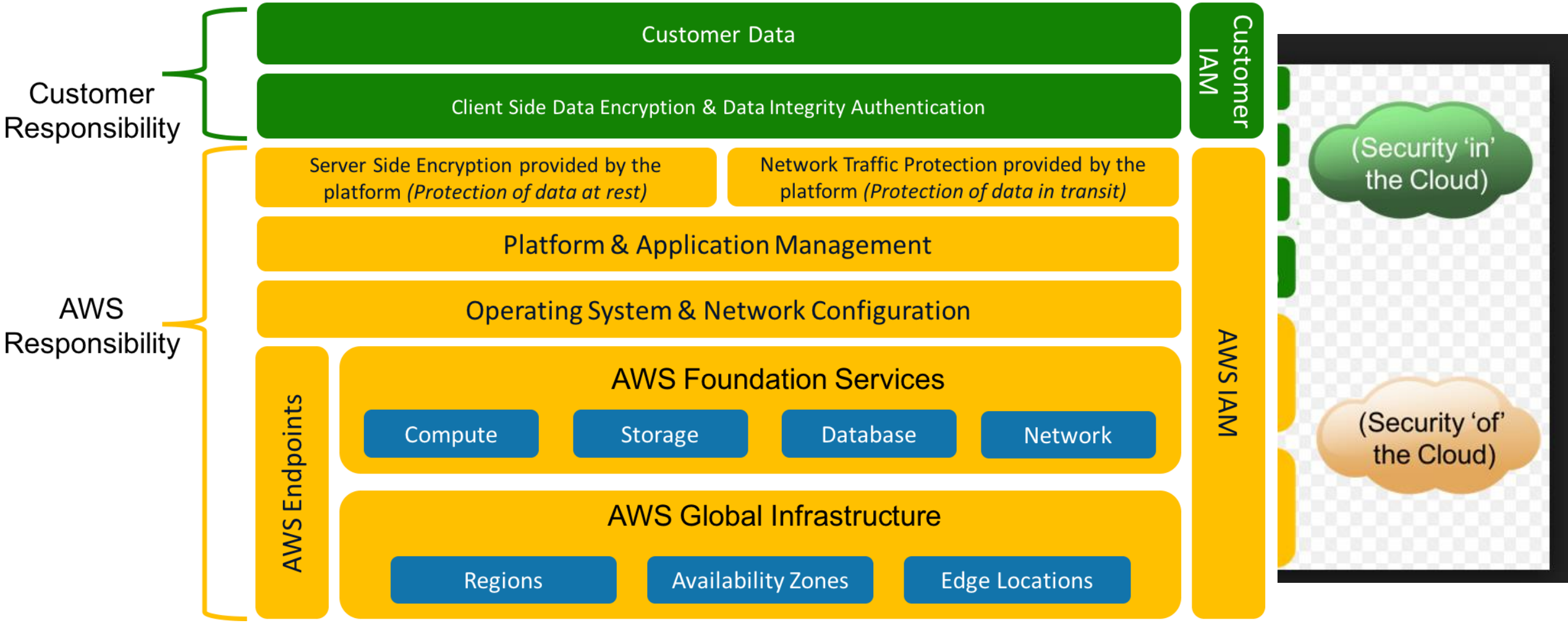
Shared Responsibility

- AWS provides security of the cloud
  - Physical
  - Network
  - Hypervisor
  - Managed services (DynamoDB, Redshift, etc.)

Shared Responsibility

- You provide security in the cloud
  - Guest OS
  - Application
  - User Data

Shared Responsibility Model



Shared Responsibility Example