ENGENHARIA DE SOFTWARE

41492-ES

Nuno Sá Couto

(nuno.sacouto@ua.pt)

Department of Electronics, Telecommunications and Informatics (DETI)

UNIVERSITY OF AVEIRO (UA), PORTUGAL

2022

universidade de aveiro

Cloud Computing fundamentals

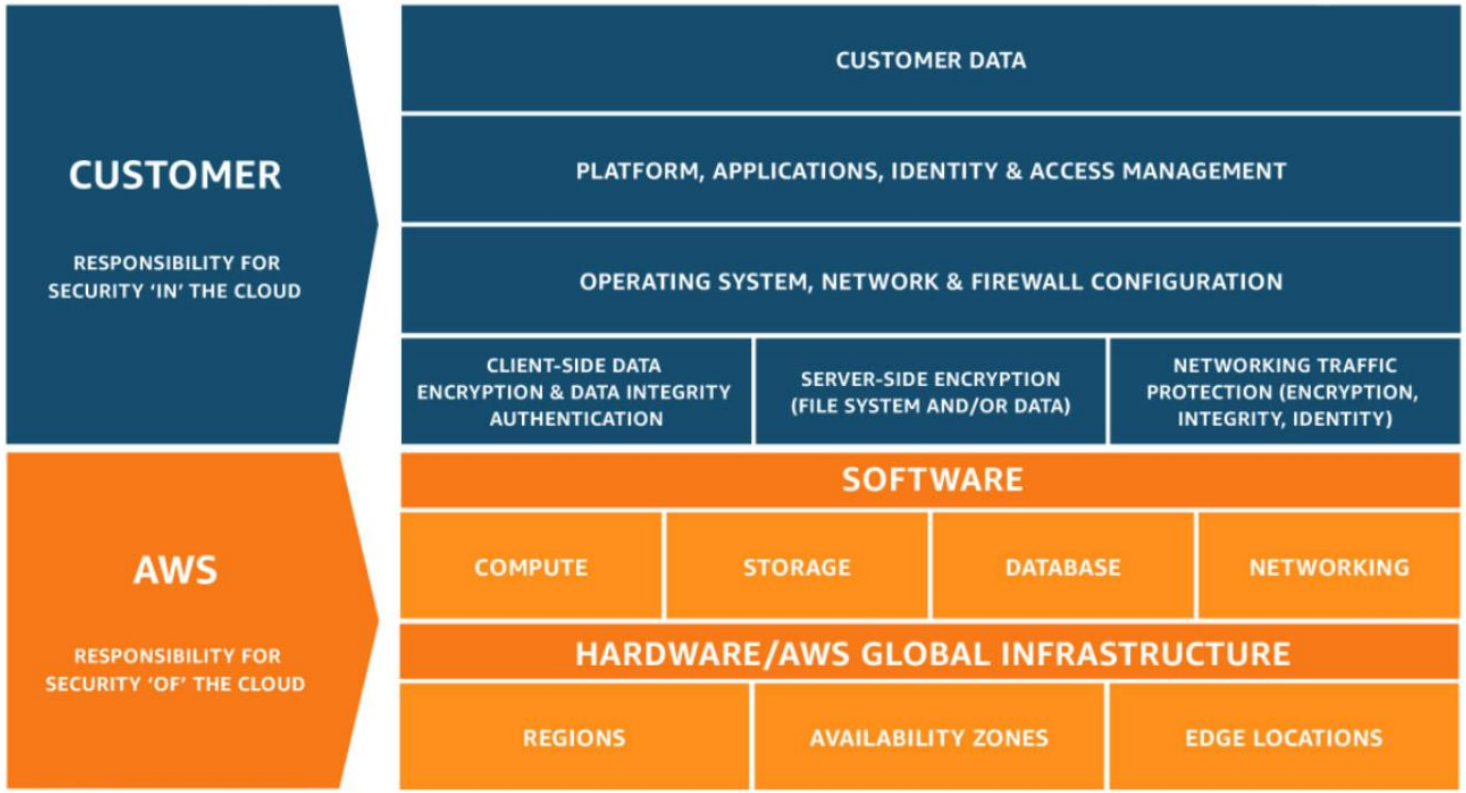
Agenda

- AWS Educate
  - AWS Academy Cloud Foundations [27857]
    - Module 1: Cloud Concepts Overview
    - Module 2: Cloud Economics and Billing
    - Module 3: AWS Global Infrastructure Overview
    - **Module 4: AWS Cloud Security**
    - Module 5: Networking and Content Delivery
    - Module 6: Compute
    - Module 7: Storage
    - Module 8: Databases
    - Module 9: Cloud Architecture
    - Module 10: Automatic Scaling and Monitoring

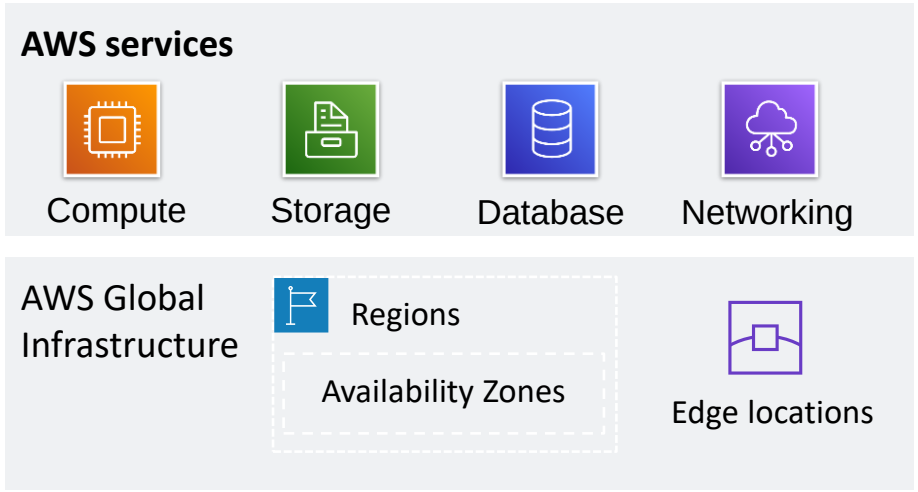
Module 4: AWS Cloud Security

Section 1: AWS shared responsibility model

AWS shared responsibility model



AWS responsibility: Security *of* the cloud

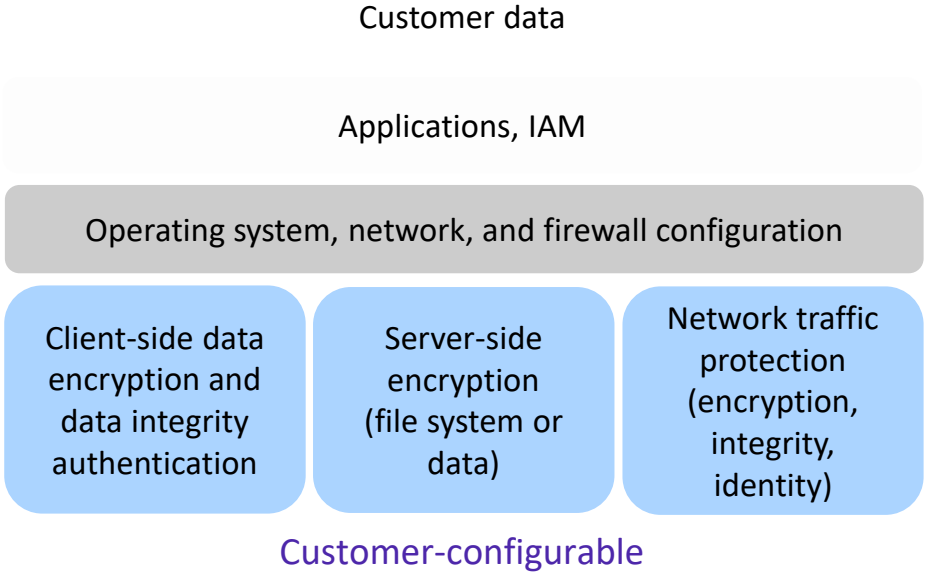


AWS responsibilities:

- Physical security of data centers
  - Controlled, need-based access
- Hardware and software infrastructure
  - Storage decommissioning, host operating system (OS) access logging, and auditing
- Network infrastructure
  - Intrusion detection

- Virtualization infrastructure
  - Instance isolation

Customer responsibility: Security *in* the cloud



Customer responsibilities:

- Amazon Elastic Compute Cloud (Amazon EC2) instance operating system
  - Including patching, maintenance
- Applications
  - Passwords, role-based access, etc.
- Security group configuration
- OS or host-based firewalls
  - Including intrusion detection or prevention systems
- Network configurations
- Account management
  - Login and permission settings for each user

Service characteristics and security responsibility

Example services managed by the customer

Amazon EC2

Amazon Elastic Block Store (Amazon EBS)

Amazon Virtual Private Cloud (Amazon VPC)

Example services managed by AWS

AWS Lambda

Amazon Relational Database Service (Amazon RDS)

AWS Elastic Beanstalk

Infrastructure as a service (IaaS)

- Customer has more flexibility over configuring networking and storage settings
- Customer is responsible for managing more aspects of the security
- Customer configures the access controls

Platform as a service (PaaS)

- Customer does not need to manage the underlying infrastructure
- AWS handles the operating system, database patching, firewall configuration, and disaster recovery
- Customer can focus on managing code or data

Service characteristics and security responsibility (continued)

SaaS examples

AWS Trusted Advisor

AWS Shield

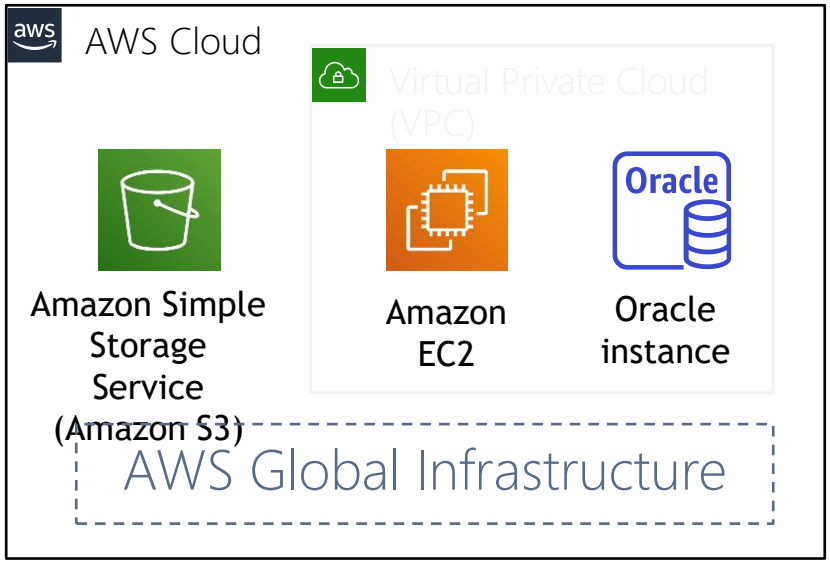
Amazon Chime

Software as a service (SaaS)

- Software is centrally hosted
- Licensed on a subscription model or pay-as-you-go basis.
- Services are typically accessed via web browser, mobile app, or application programming interface (API)
- Customers do not need to manage the infrastructure that supports the service

Activity: Scenario 1 of 2

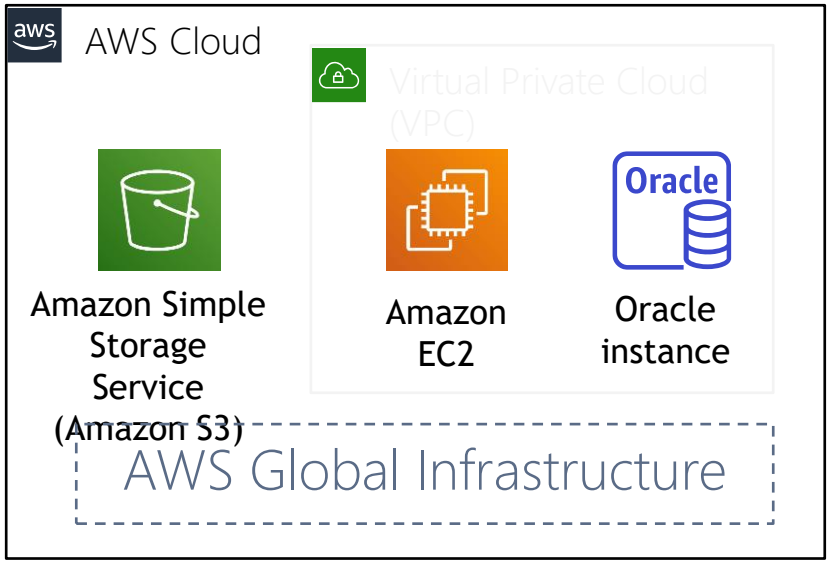
Consider this deployment. Who is responsible - AWS or the customer?



1. Upgrades and patches to the operating system on the EC2 instance?
   - ANSWER:
2. Physical security of the data center?
   - ANSWER:
3. Virtualization infrastructure?
   - ANSWER:
4. EC2 security group settings?
   - ANSWER:
5. Configuration of applications that run on the EC2 instance?
   - ANSWER:
6. Oracle upgrades or patches If the Oracle instance runs as an Amazon RDS instance?
   - ANSWER:
7. Oracle upgrades or patches If Oracle runs on an EC2 instance?
   - ANSWER:
8. S3 bucket access configuration?
   - ANSWER:

Activity: Scenario 1 of 2

Consider this deployment. Who is responsible - AWS or the customer?



1. Upgrades and patches to the operating system on the EC2 instance?
   - **ANSWER:** The customer
2. Physical security of the data center?
   - **ANSWER:** AWS
3. Virtualization infrastructure?
   - **ANSWER:** AWS
4. EC2 security group settings?
   - **ANSWER:** The customer
5. Configuration of applications that run on the EC2 instance?
   - **ANSWER:** The customer
6. Oracle upgrades or patches If the Oracle instance runs as an Amazon RDS instance?
   - **ANSWER:** AWS
7. Oracle upgrades or patches If Oracle runs on an EC2 instance?
   - **ANSWER:** The customer
8. S3 bucket access configuration?
   - **ANSWER:** The customer

Section 1 key takeaways

- AWS and the customer share security responsibilities:
  - AWS is responsible for security of the cloud
  - Customer is responsible for security in the cloud
- **AWS is responsible for protecting the infrastructure**—including hardware, software, networking, and facilities—that run AWS Cloud services
- For services that are categorized as infrastructure as a service (IaaS), the **customer is responsible for performing necessary security configuration and management tasks**
  - For example, guest OS updates and security patches, firewall, security group configurations

Module 4: AWS Cloud Security

Section 2: AWS Identity and Access Management (IAM)

AWS Identity and Access Management (IAM)

- Use **IAM** to manage access to **AWS resources** –
  - A resource is an entity in an AWS account that you can work with
  - Example resources; An Amazon EC2 instance or an Amazon S3 bucket
- *Example* – Control who can terminate Amazon EC2 instances
- Define fine-grained access rights –
  - **Who** can access the resource
  - **Which** resources can be accessed and what can the user do to the resource
  - **How** resources can be accessed
- IAM is a no-cost AWS account feature

AWS Identity and Access Management (IAM)

IAM: Essential components

IAM user

A **person** *or* **application** that can authenticate with an AWS account.

IAM group

A **collection of IAM users** that are granted identical authorization.

IAM policy

The document that defines **which resources can be accessed** and the **level of access** to each resource.

IAM role

Useful mechanism to grant a set of permissions for making AWS service requests.

IAM: Authorization

- Assign permissions by creating an IAM policy.
- Permissions determine **which resources and operations** are allowed:
  - All permissions are implicitly denied by default.
  - If something is explicitly denied, it is never allowed.

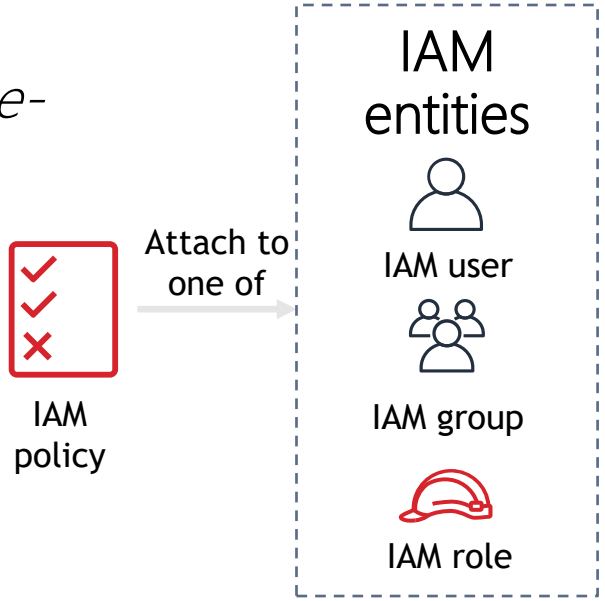
IAM permissions

Best practice: Follow the **principle of least privilege**.

Note: The scope of IAM service configurations is **global**. Settings apply across all AWS Regions.

IAM policies

- An IAM policy is a document that defines permissions
  - Enables fine-grained access control
- Two types of policies – *identity-based* and *resource-based*
- **Identity-based** policies –
  - Attach a policy to any IAM entity
    - An IAM user, an IAM group, or an IAM role
  - Policies specify:
    - Actions that ***may*** be performed by the entity
    - Actions that ***may not*** be performed by the entity
  - A single *policy* can be attached to multiple *entities*
  - A single *entity* can have multiple *policies* attached to it
- **Resource-based** policies
  - Attached to a resource (such as an S3 bucket)



IAM policy example

```
{
  "Version": "2012-10-17",
  "Statement":[{
    "Effect":"Allow",
    "Action":["DynamoDB:*","s3:*"],
    "Resource":[
       "arn:aws:dynamodb:region:account-number-without-hyphens:table/table-name",
      "arn:aws:s3:::bucket-name",
      "arn:aws:s3:::bucket-name/*"]
    },
    {
    "Effect":"Deny",
    "Action":["dynamodb:*","s3:*"],
    "NotResource":["arn:aws:dynamodb:region:account-number-without-hyphens:table/table-name",
      "arn:aws:s3:::bucket-name",
      "arn:aws:s3:::bucket-name/*"]
    }
  ]
}
```

Explicit allow gives users access to a specific DynamoDB table and…

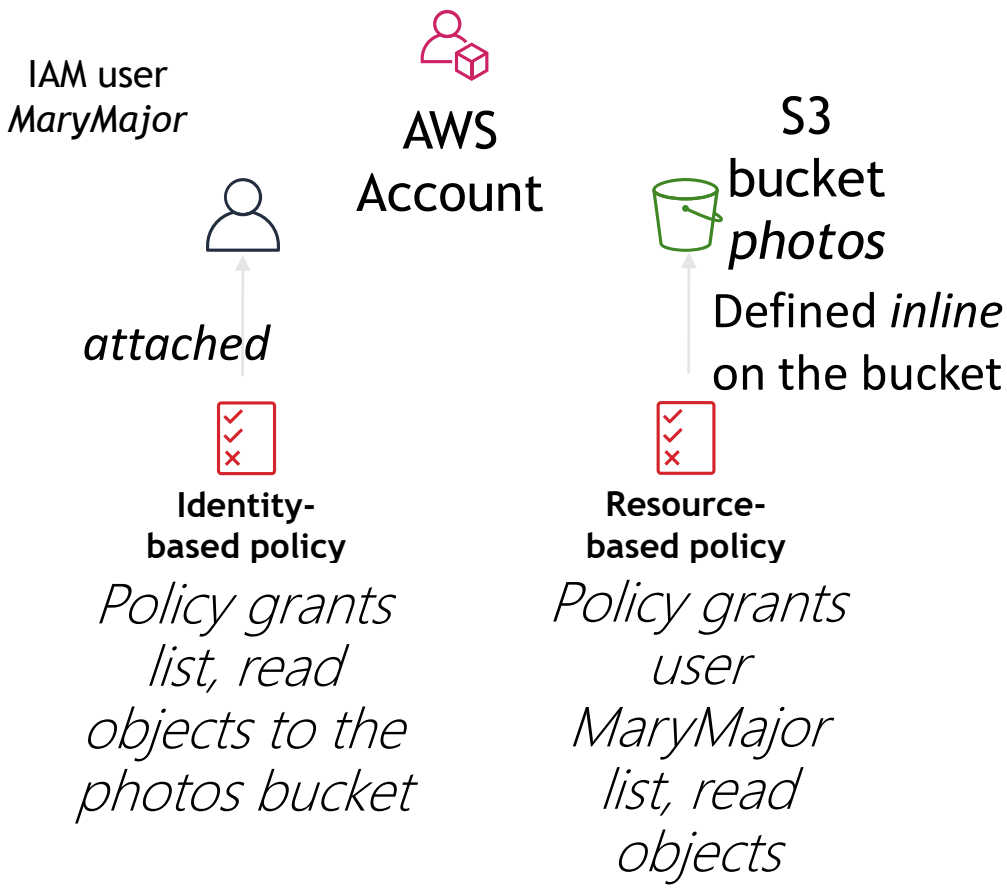
…Amazon S3 buckets.

Explicit deny ensures that the users cannot use any other AWS actions or resources other than that table and those buckets.

An explicit deny statement **takes precedence** over an allow statement.

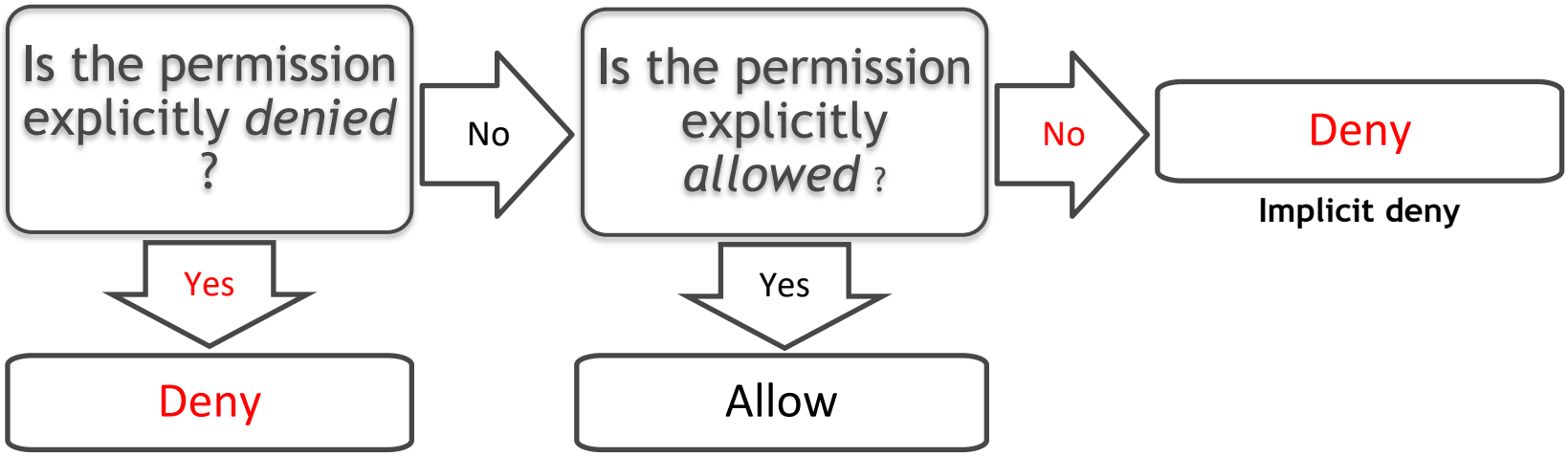
Resource-based policies

- *Identity-based policies* are attached to a user, group, or role
- **Resource-based policies** are attached to a **resource** (*not* to a user, group or role)
- Characteristics of resource-based policies –
  - Specifies who has access to the resource and what actions they can perform on it
  - The policies are *inline* only, not managed
- Resource-based policies are supported only by some AWS services



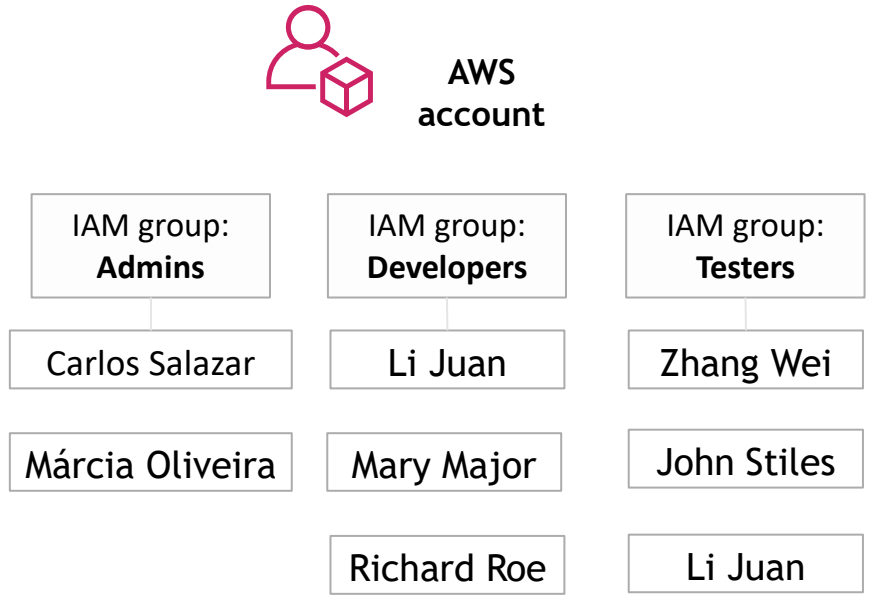
IAM permissions

How IAM determines permissions:



IAM groups

- An **IAM group** is a collection of IAM users
- A group is used to grant the same permissions to multiple users
  - Permissions granted by attaching IAM *policy* or policies to the group
- A user can belong to multiple groups
- There is no default group
- Groups cannot be nested



IAM roles

- An **IAM role** is an IAM identity with specific permissions
- Similar to an IAM user
  - Attach permissions policies to it
- Different from an IAM user
  - Not uniquely associated with one person
  - Intended to be *assumable* by a **person**, **application**, or **service**
- Role provides *temporary* security credentials
- Examples of how IAM roles are used to **delegate** access –
  - Used by an IAM user in the same AWS account as the role
  - Used by an AWS service—such as Amazon EC2—in the same account as the role
  - Used by an IAM user in a different AWS account than the role

IAM role

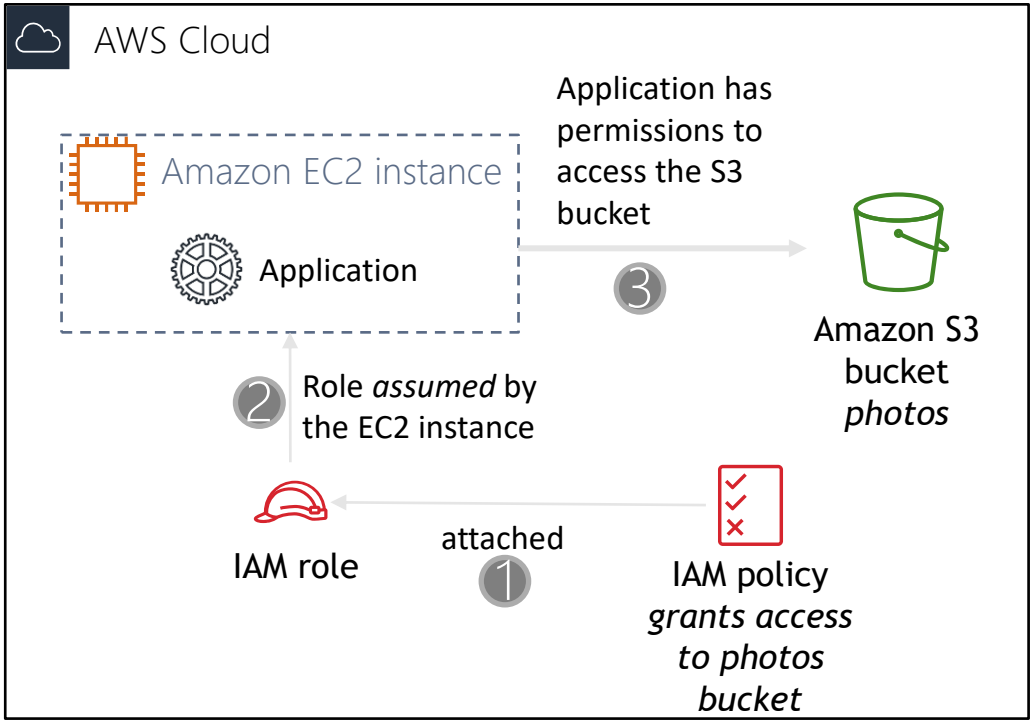
Example use of an IAM role

Scenario:

- An application that runs on an EC2 instance needs access to an S3 bucket

Solution:

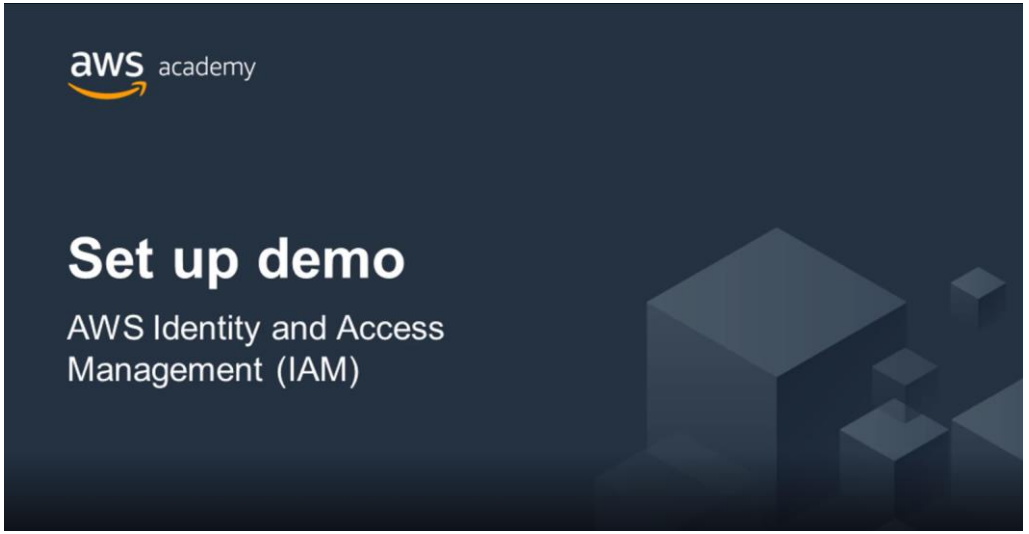
- Define an IAM policy that grants access to the S3 bucket.
- Attach the policy to a role
- Allow the EC2 instance to assume the role



Section 2 key takeaways

- **IAM policies** are constructed with JavaScript Object Notation (JSON) and define permissions.
  - IAM policies can be attached to any **IAM entity**.
  - Entities are IAM users, IAM groups, and IAM roles.
- An **IAM user** provides a way for a person, application, or service to authenticate to AWS.
- An **IAM group** is a simple way to attach the same policies to multiple users.
- An **IAM role** can have permissions policies attached to it, and can be used to delegate temporary access to users or applications.

Recorded demo: IAM



Module 4: AWS Cloud Security

Section 3: Securing a new AWS account

Securing a new AWS account: Account root user

Step 1: Stop using the account root user as soon as possible.

- The account root user has unrestricted access to all your resources.

- To stop using the account root user:
  1. While you are logged in as the account root user, create IAM users for all the group. Save the access keys if needed.
  2. Create an IAM group, give it full administrator permissions, and add the IAM user to the group.
  3. Enable a password policy for users.
  4. Sign in with your new IAM user credentials.
  5. Store your account root user credentials in a secure place.
  6. Create also a test IAM user

IAM Policies hands-On

1. Go to User Groups
2. Validate that the test user created in the previous lab is present
3. With Root account remove user from assigned User Group
4. In a separate page logged as the IAM user validate if the user lost its permissions (menu users)
    a. Do not have permission iamListUsers
5. With the root account navigate to Users > Select IAM User > Menu Permissions
    a. Add permissions
        i. Attach policies directly
            1) IAMReadOnlyAccess
6. With the IAM account refresh the page with the error
    a. Now it should be possible to list all users
7. Validate that now with the IAM user, it is not possible to create a new user group
    a. Should get exception, we have only read only
8. As root, re-add IAM user to admin group (esAdmin)
9. Create a second group named esDevelopers
10. Add IAM user to the second group also
11. Add a random policy to the group
12. Validate that IAM user now has the following permissions
    a. 1 - Attached directly
    b. 2 - Attached from group
        i. 1 per each group
13. As root go to the IAM policies Menu
    a. Select AdministratorAccess
        i. Select JSON detail
        ii. Select policy Detail > Allowed Services: All
    b. Select IAMReadOnlyAccess
        i. Select policy Detail > Allowed Services: 1
        ii. Select JSON detail and check which service it has access
14. As root create a new policy
    a. Write a simple IAM policy using the visual editor
        i. Service IAM
        ii. Actions: ListUser, GetUser
        iii. Resources: All
    b. Validate JSON
15. Delete the just created Policy
16. Delete the esDevelopers group
17. Remove the inline policy from the IAM user
18. As IAM user validate that it has full access again as a Admin
19. Housekeeping: remove test user

Securing a new AWS account: MFA

Step 2: Enable multi-factor authentication (MFA).

- Require MFA for your account root user and for all IAM users.
- You can also use MFA to control access to AWS service APIs.

- Options for retrieving the MFA token –
  - Virtual MFA-compliant applications:
    - Google Authenticator.
    - Authy Authenticator (Windows phone app)
      - Allows multi device (best for a group setting)
  - U2F security key devices:
    - For example, YubiKey.
  - Hardware MFA options:
    - Key fob or display card offered by Gemalto.

MFA token

Setup CLI version 2

Search "aws CLI install Windows"

- Go to AWS CLI Page
- Scroll to the OS you use
  - Download and install the AWS CLI MSI (if you use Windows)
- Install
- Test Installation (aws –version)

With your IAM user log on the AWS Web Console

- Go to users > Security Credentials > Create new access key
- On the CLI: aws configure
  - Enter access key id / secret access key / region (eu-west-1)
- Test / list all users:
  - aws iam list-users

Section 3 key takeaways

Best practices to secure an AWS account:

- **Secure** logins with multi-factor authentication (MFA).
- **Delete** account root user access keys.
- **Create** individual IAM users and grant permissions according to the principle of least privilege.
- **Use groups** to assign permissions to IAM users.
- **Configure** a strong password policy.
- **Delegate** using roles instead of sharing credentials.
- **Monitor** account activity by using AWS CloudTrail.

Module 4: AWS Cloud Security

Section 4: Securing accounts

AWS Organizations

- **AWS Organizations** enables you to consolidate multiple AWS accounts so that you centrally manage them.

AWS Organizations

- Security features of AWS Organizations:
  - **Group AWS accounts into organizational units** (OUs) and attach different access policies to each OU.
  - **Integration and support for IAM**
    - Permissions to a user are the intersection of what is allowed by AWS Organizations and what is granted by IAM in that account.
  - **Use service control policies** to establish control over the AWS services and API actions that each AWS account can access

AWS Organizations: Service control policies

- **Service control policies (SCPs)** offer centralized control over accounts.
  - Limit permissions that are available in an account that is part of an organization.
- Ensures that accounts comply with access control guidelines.
- SCPs are *similar* to IAM permissions policies –
  - They use similar syntax.
  - However, an SCP never grants permissions.
  - Instead, SCPs specify the maximum permissions for an organization.

Module 4: AWS Cloud Security

Section 5: Securing data on AWS

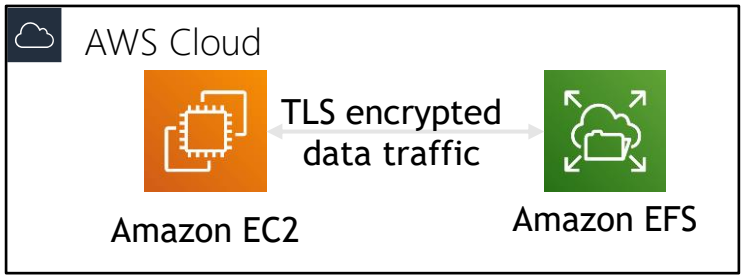
Encryption of data *at rest*

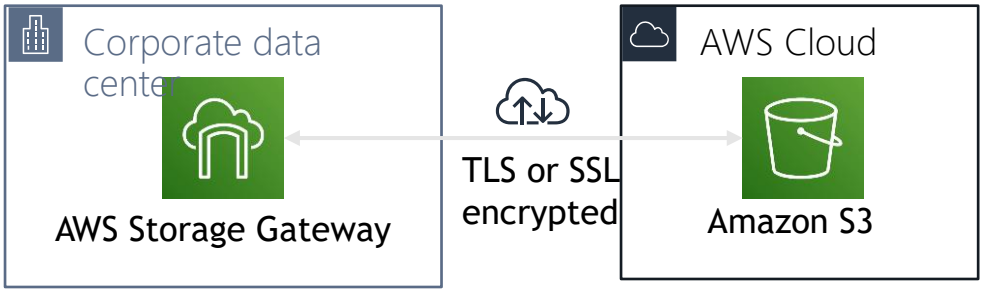
- **Encryption** encodes data with a secret key, which makes it unreadable
  - Only those who have the secret key can decode the data
  - AWS KMS can manage your secret keys

- AWS supports encryption of data at rest
  - Data at rest = Data stored physically (on disk or on tape)
  - You can encrypt data stored in any service that is supported by AWS KMS, including:
    - Amazon S3
    - Amazon EBS
    - Amazon Elastic File System (Amazon EFS)
    - Amazon RDS managed databases

Encryption of data *in transit*

- Encryption of **data in transit** (data moving across a network)
  - **Transport Layer Security (TLS)**—formerly SSL—is an open standard protocol
  - **AWS Certificate Manager** provides a way to manage, deploy, and renew TLS or SSL certificates
- Secure HTTP (HTTPS) creates a secure tunnel
  - Uses TLS or SSL for the bidirectional exchange of data
- **AWS services support data in transit encryption.**
  - Two examples:





Module 4: AWS Cloud Security

Section 6: Working to ensure compliance

AWS compliance programs

- Customers are subject to many different security and compliance regulations and requirements.
- AWS engages with certifying bodies and independent auditors to provide customers with detailed information about the policies, processes, and controls that are established and operated by AWS.

- Compliance programs can be broadly categorized –
  - Certifications and attestations
    - Assessed by a third-party, independent auditor
    - Examples: ISO 27001, 27017, 27018, and ISO/IEC 9001

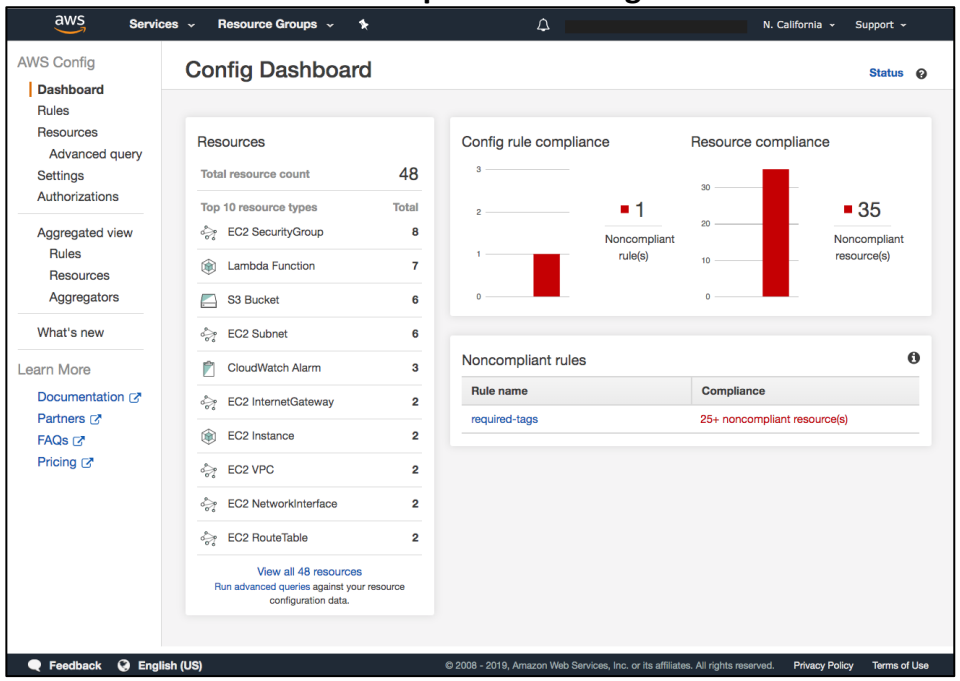
  - Laws, regulations, and privacy
    - AWS provides security features and legal agreements to support compliance
    - Examples: EU General Data Protection Regulation (GDPR), HIPAA

  - Alignments and frameworks
    - Industry- or function-specific security or compliance requirements
    - Examples: Center for Internet Security (CIS), EU-US Privacy Shield certified

AWS Config

AWS Config

Example AWS Config Dashboard view



- Assess, audit, and evaluate the configurations of AWS resources.
- Use for continuous monitoring of configurations.
- Automatically evaluate *recorded* configurations versus *desired* configurations.
- Review configuration changes.
- View detailed configuration histories.
- Simplify compliance auditing and security analysis.

AWS Artifact


AWS Artifact

- Is a resource for compliance-related information
- Provide access to security and compliance reports, and select online agreements
- Can access example downloads:
  - AWS ISO certifications
  - Payment Card Industry (PCI) and Service Organization Control (SOC) reports
- Access AWS Artifact directly from the AWS Management Console
  - Under **Security, Identify & Compliance**, click **Artifact**.

Section 6 key takeaways

- **AWS security compliance programs** provide information about the policies, processes, and controls that are established and operated by AWS.
- **AWS Config** is used to assess, audit, and evaluate the configurations of AWS resources.
- **AWS Artifact** provides access to security and compliance reports.

Additional resources

- AWS Cloud Security home page
- AWS Security Resources
- AWS Security Blog
- Security Bulletins
- Vulnerability and Penetration testing
- AWS Well-Architected Framework – Security pillar
- AWS documentation - IAM Best Practices