Module 4: AWS Cloud Security

Module overview

Topics

- AWS shared responsibility model
- AWS Identity and Access Management (IAM)
- Securing a new AWS account
- Securing accounts
- Securing data on AWS
- Working to ensure compliance

Activities

- AWS shared responsibility model activity

Knowledge check

Module objectives

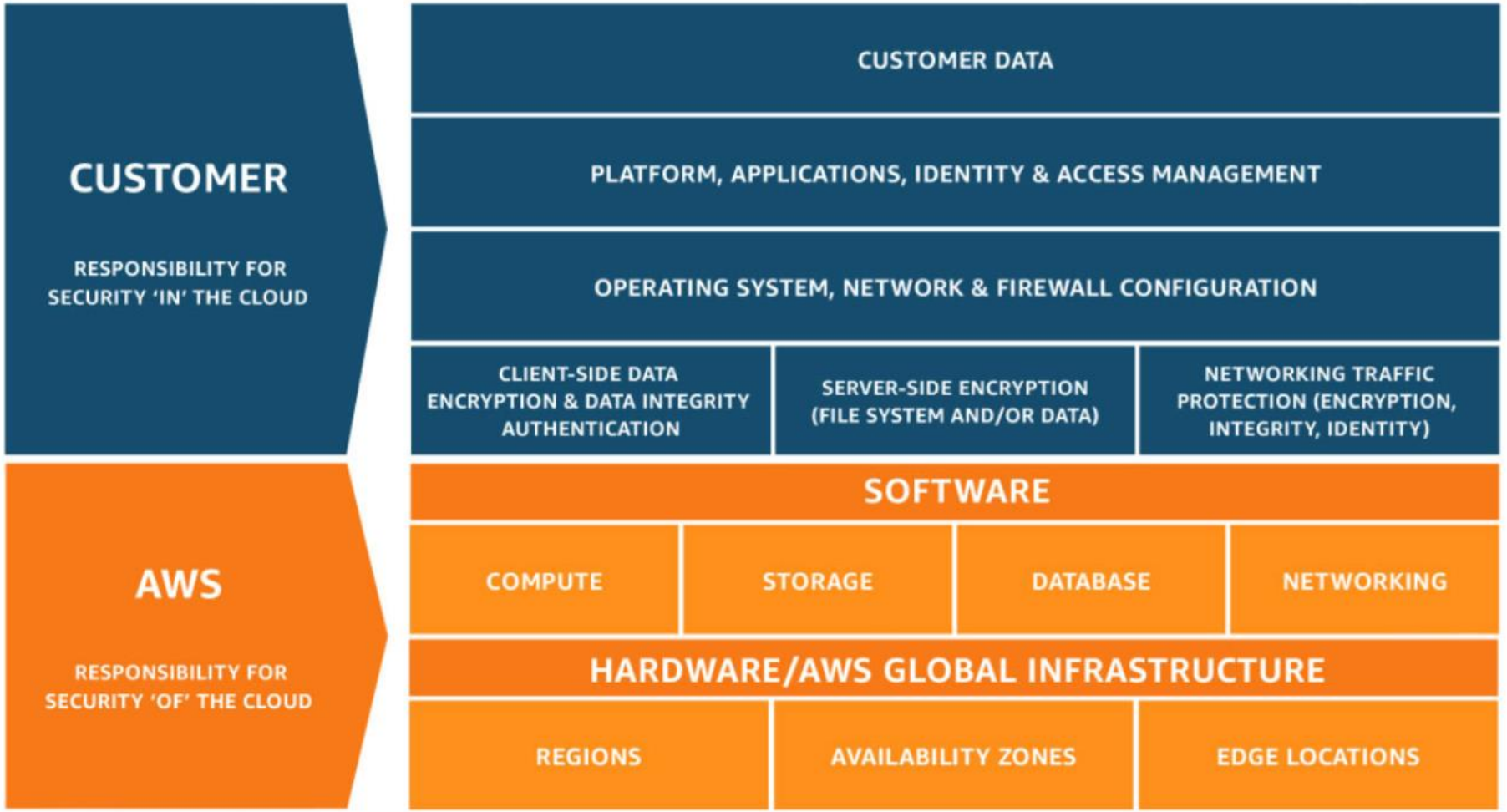
After completing this module, you should be able to:

- Recognize the shared responsibility model
- Identify the responsibility of the customer and AWS
- Recognize IAM users, groups, and roles
- Describe different types of security credentials in IAM
- Identify the steps to securing a new AWS account
- Explore IAM users and groups
- Recognize how to secure AWS data
- Recognize AWS compliance programs

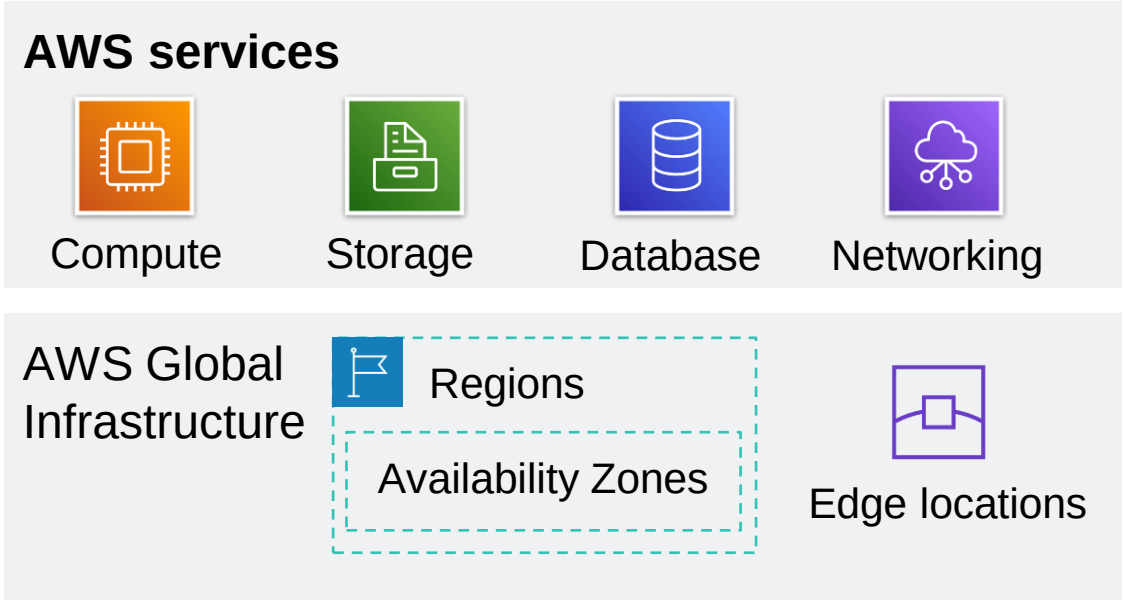
Section 1: AWS shared responsibility model

Module 4: AWS Cloud Security

AWS shared responsibility model



AWS responsibility: Security *of* the cloud

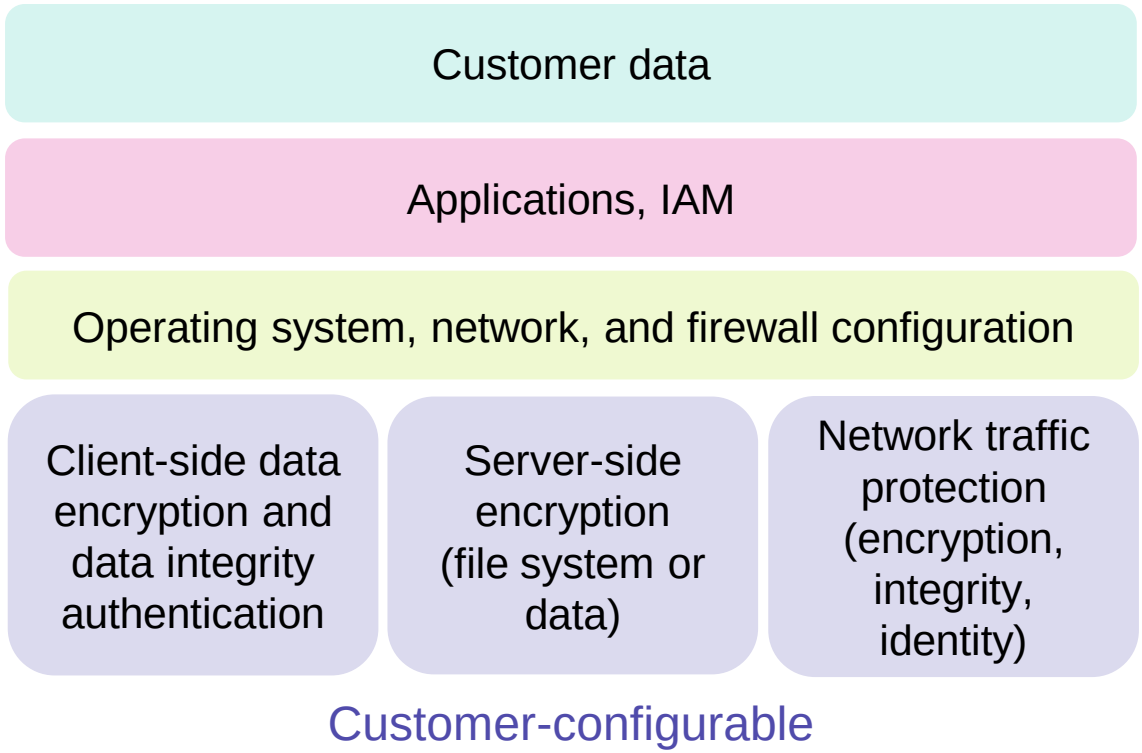


AWS responsibilities:

- 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

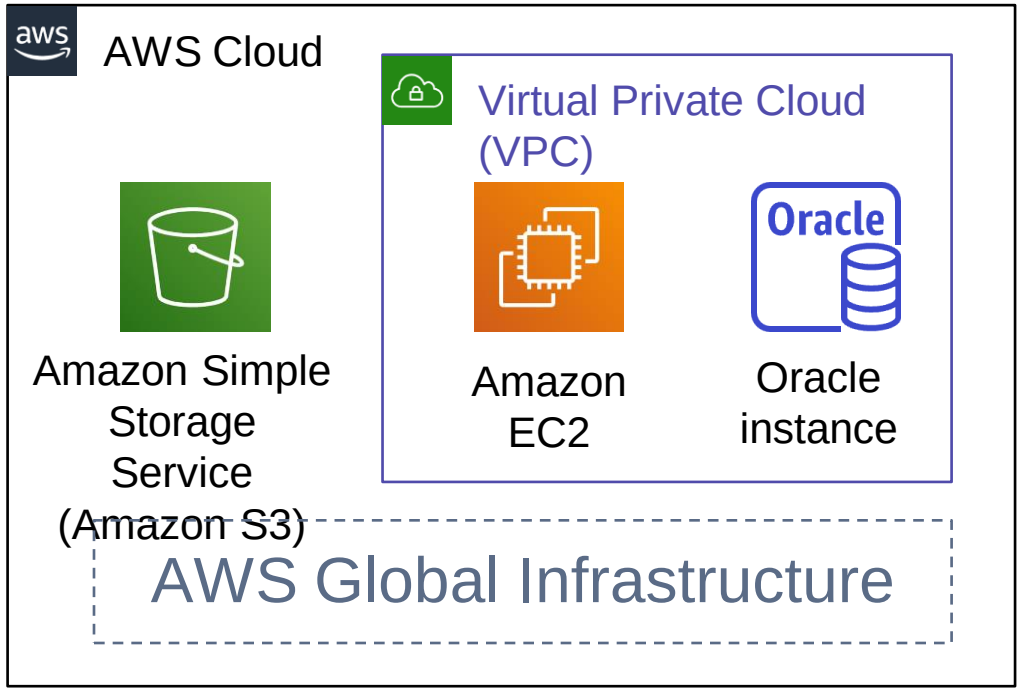


Customer responsibilities:

- 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

Activity: Scenario 1 of 2

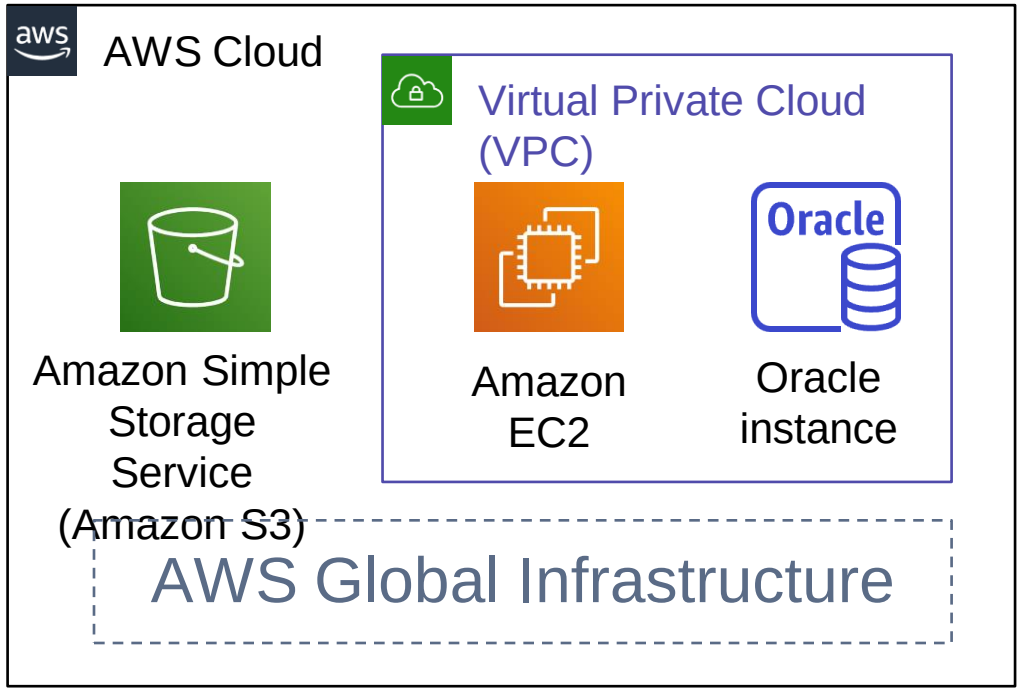
Consider this deployment. Who is responsible – AWS or the customer?



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

Activity: Scenario 1 of 2 Answers

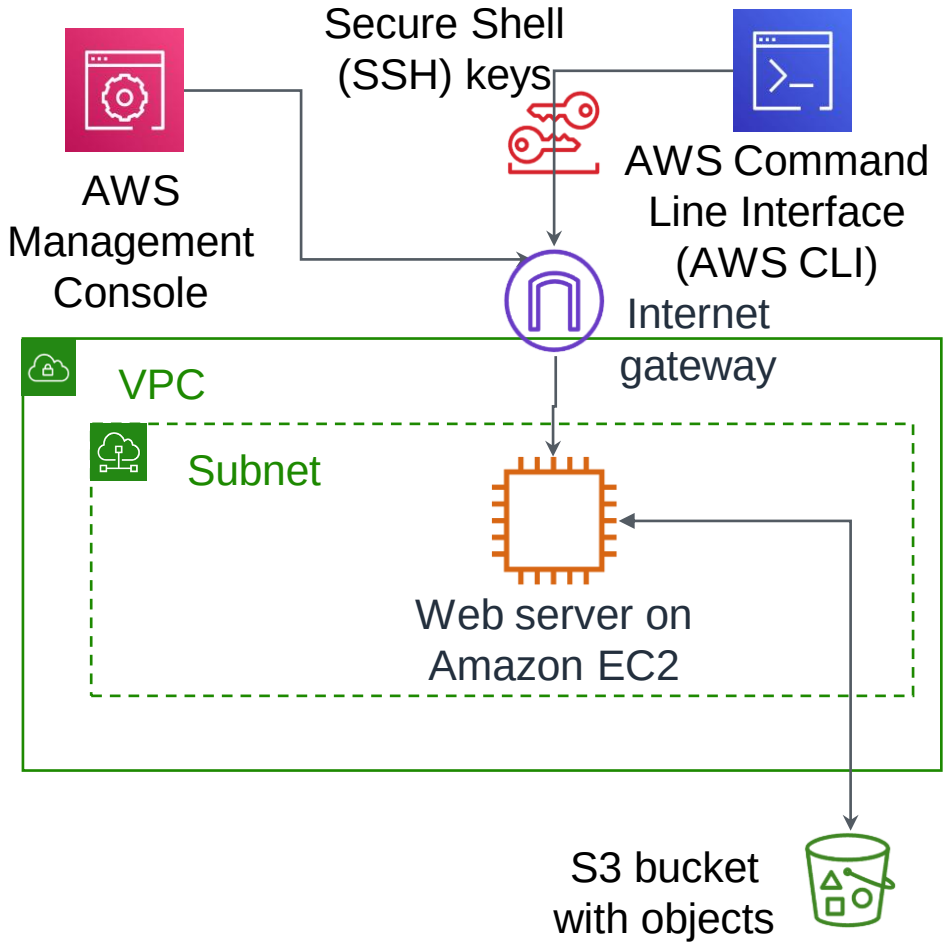
Consider this deployment. Who is responsible – AWS or the customer?



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

Activity: Scenario 2 of 2

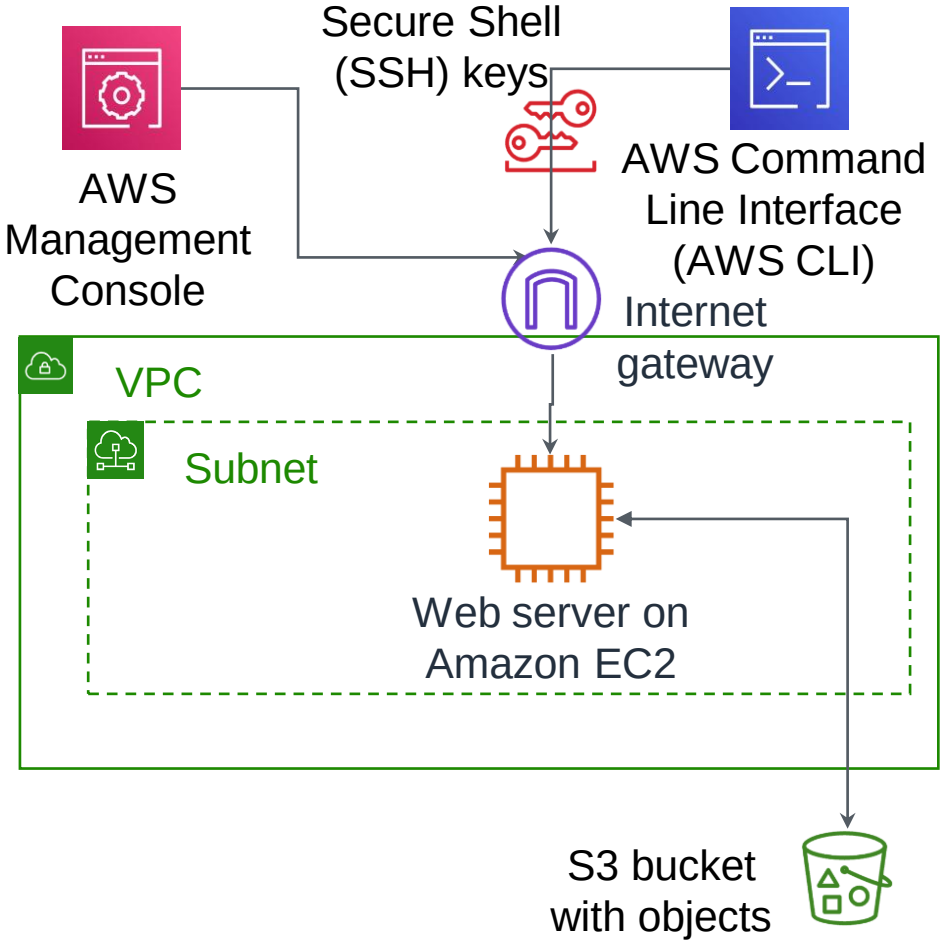
Consider this deployment. Who is responsible – AWS or the customer?



1. Ensuring that the AWS Management Console is not hacked?
2. Configuring the subnet?
3. Configuring the VPC?
4. Protecting against network outages in AWS Regions?
5. Securing the SSH keys
6. Ensuring network isolation between AWS customers' data?
7. Ensuring low-latency network connection between the web server and the S3 bucket?
8. Enforcing multi-factor authentication for all user logins?

Activity: Scenario 2 of 2 Answers

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



1. Ensuring that the AWS Management Console is not hacked?
   - **ANSWER:** AWS
2. Configuring the subnet?
   - **ANSWER:** The customer
3. Configuring the VPC?
   - **ANSWER:** The customer
4. Protecting against network outages in AWS Regions?
   - **ANSWER:** AWS
5. Securing the SSH keys
   - **ANSWER:** The customer
6. Ensuring network isolation between AWS customers' data?
   - **ANSWER:** AWS
7. Ensuring low-latency network connection between the web server and the S3 bucket?
   - **ANSWER:** AWS
8. Enforcing multi-factor authentication for all user logins?
   - **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

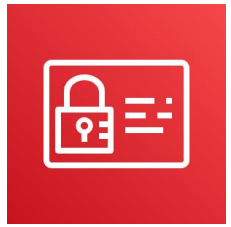
Section 2: AWS Identity and Access Management (IAM)

Module 4: AWS Cloud Security

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



AWS Identity and Access Management (IAM)

- 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

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.

Authenticate as an IAM user to gain access

When you define an **IAM user**, you select what ***types of access*** the user is permitted to use.

Programmatic access

- Authenticate using:
  - Access key ID
  - Secret access key
- Provides AWS CLI and AWS SDK access

AWS CLI

AWS Tools and SDKs

AWS Management Console access

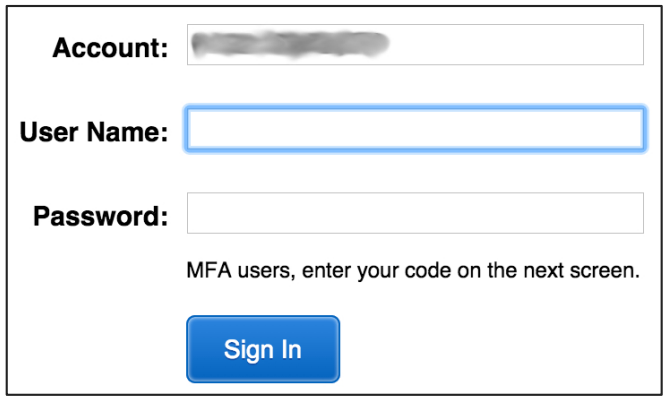
- Authenticate using:
  - 12-digit Account ID *or* alias
  - IAM user name
  - IAM password
- If enabled, **multi-factor authentication (MFA)** prompts for an authentication code.

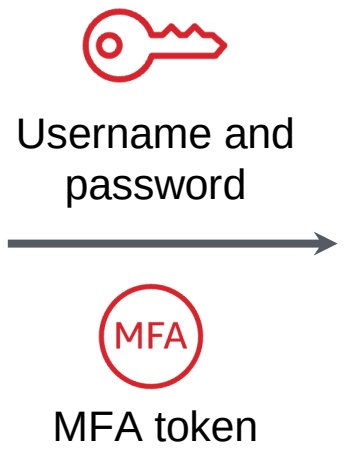
AWS Management Console

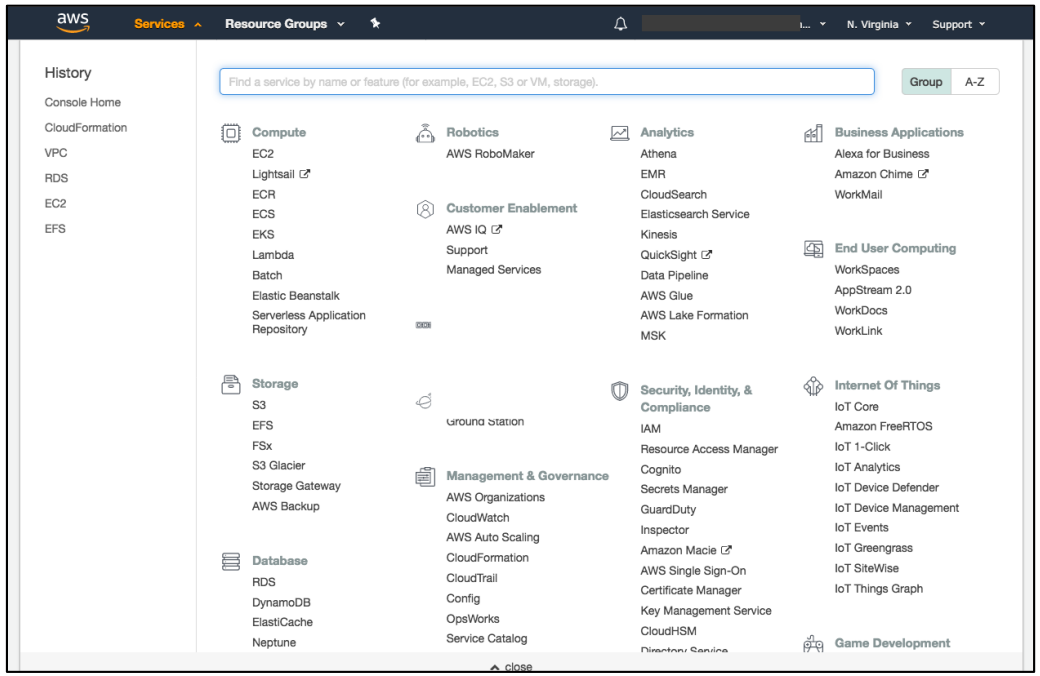
IAM MFA

- MFA provides increased security.

- In addition to **username** and **password**, MFA requires a unique authentication code to access AWS services.



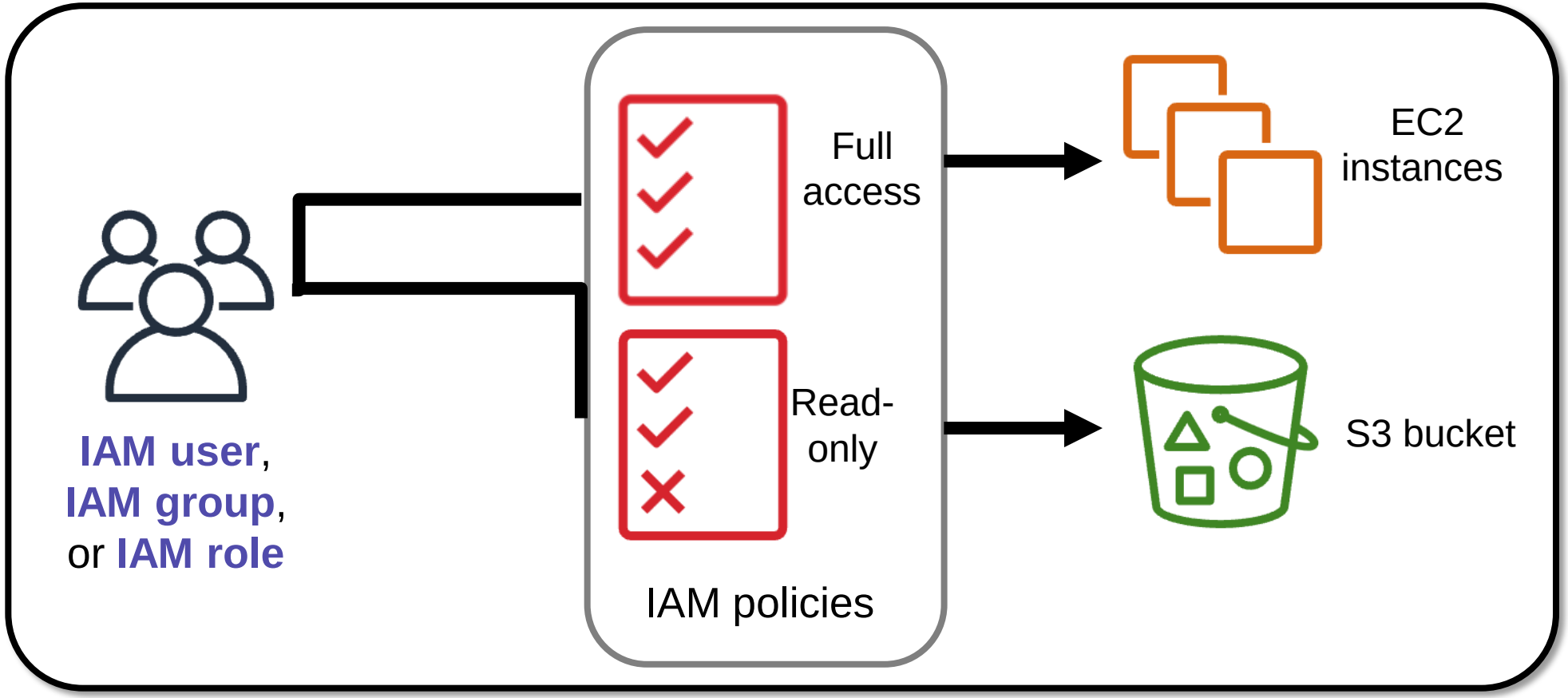




AWS Management Console

Authorization: What actions are permitted

After the user or application is connected to the AWS account, what are they allowed to do?



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.

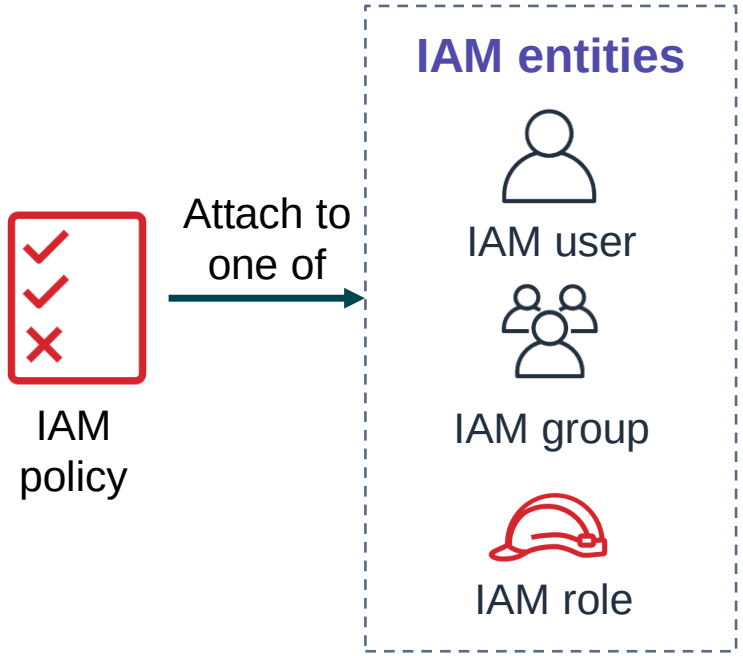
Best practice**:** Follow the **principle of least privilege**.

IAM permissions

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…

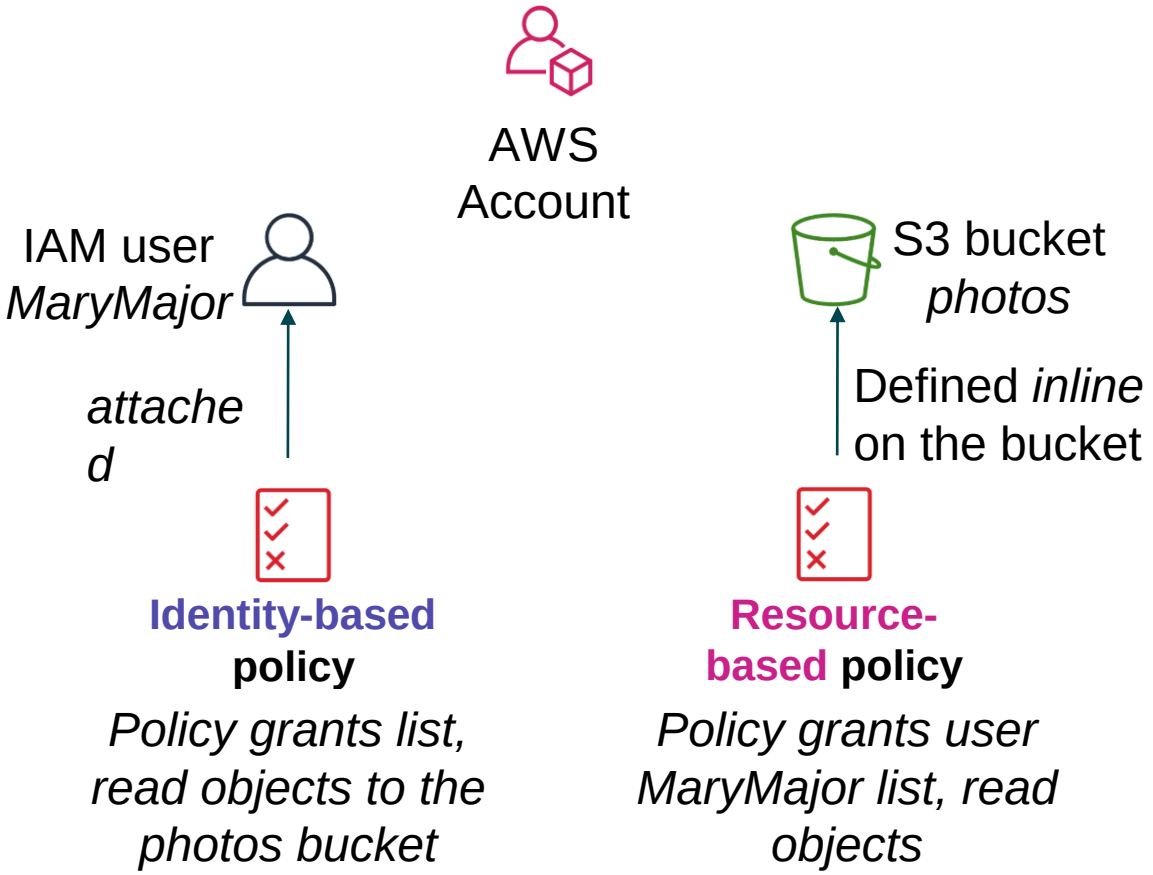
…Amazon S3 buckets.

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.

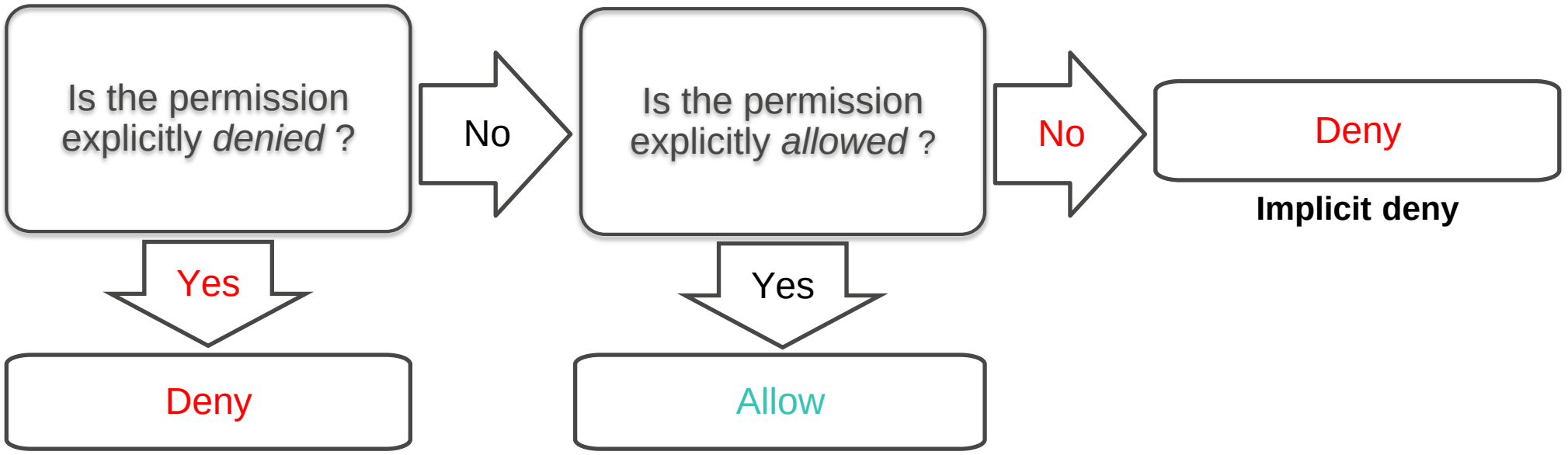
Resource-based policies

- *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



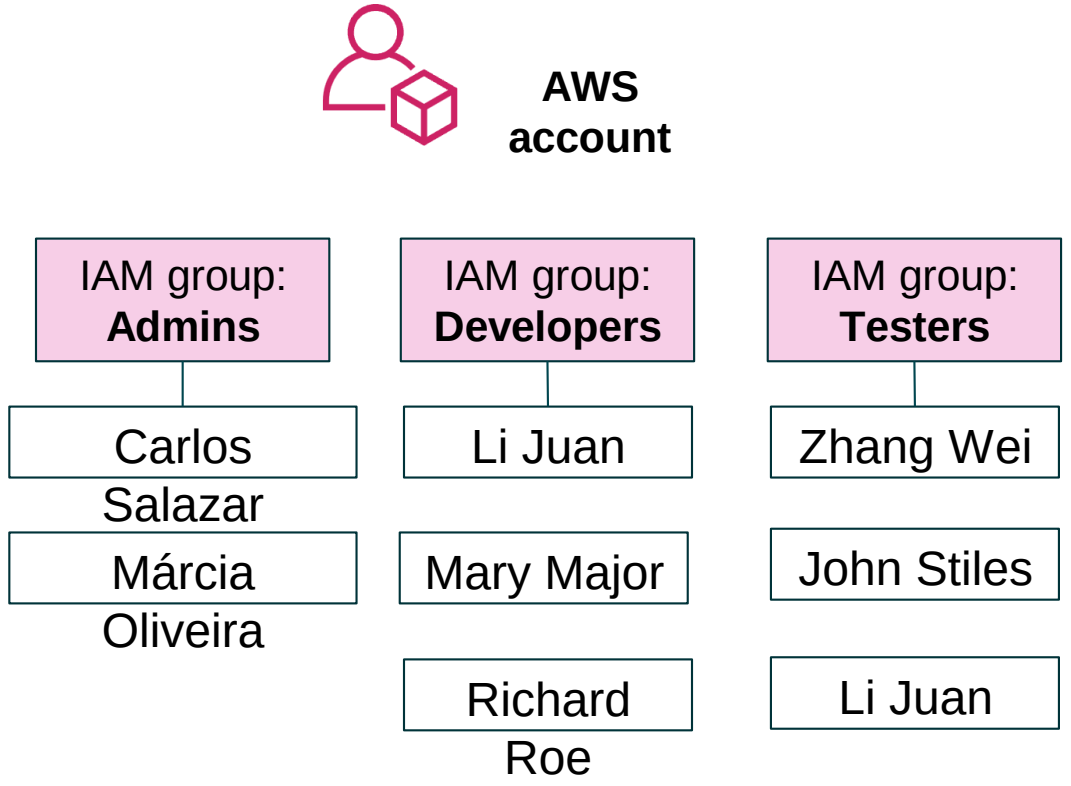
IAM permissions

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

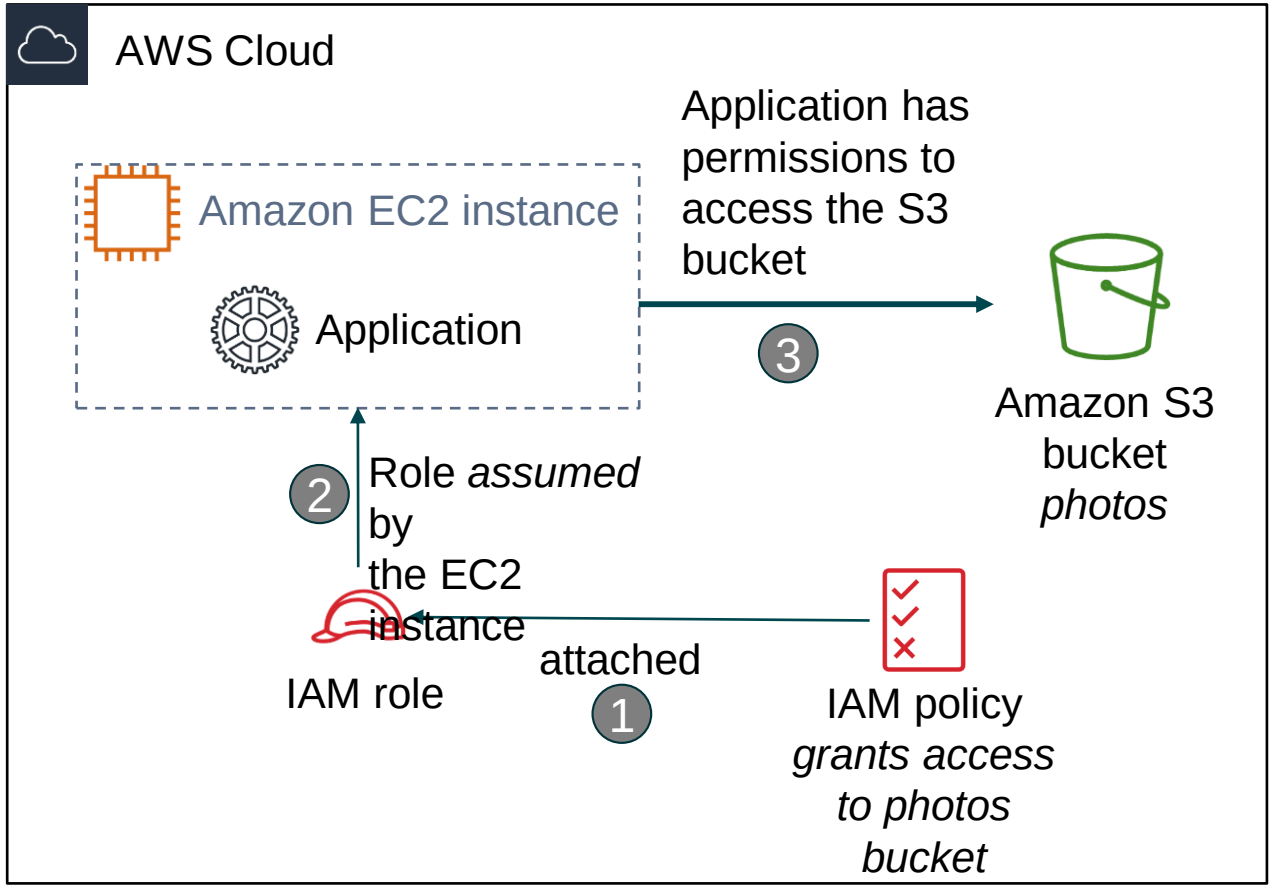
Example use of an IAM role

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



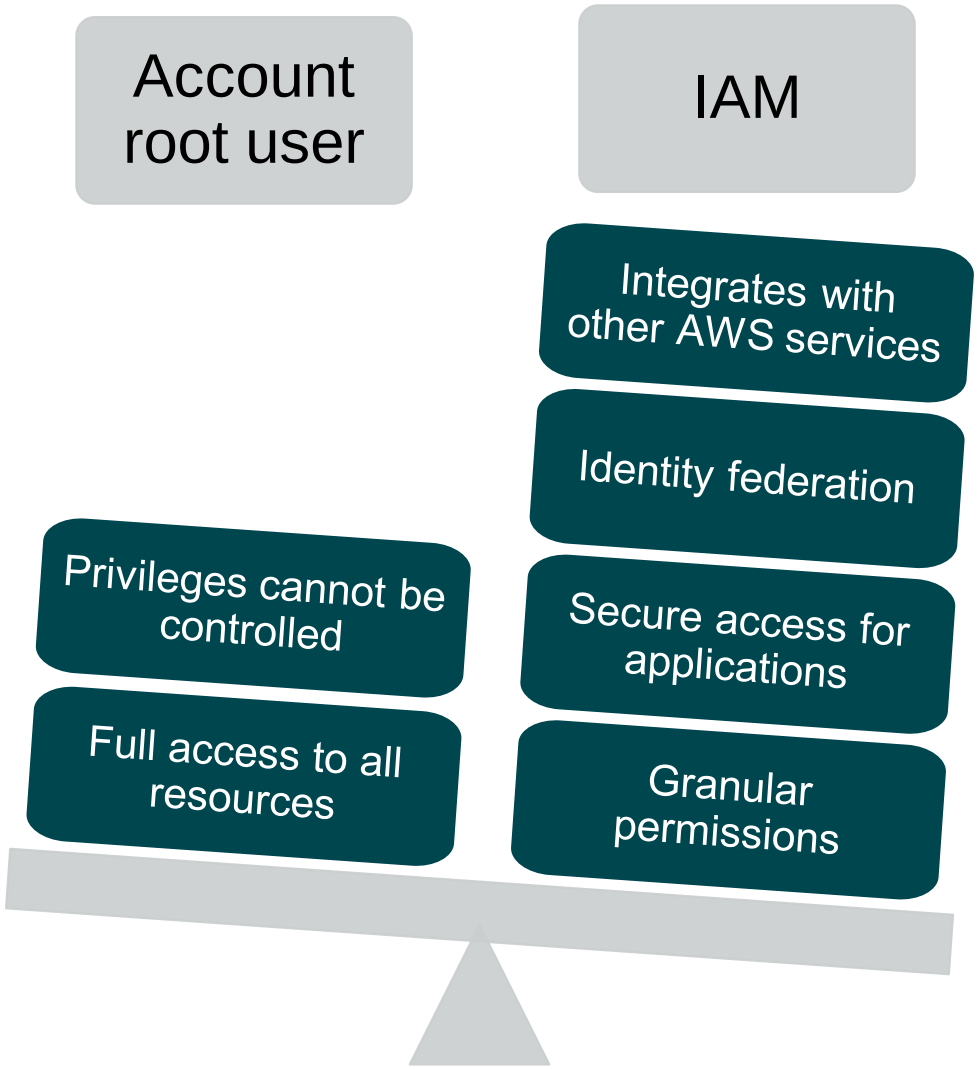
Section 2 key takeaways

- **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.

Section 3: Securing a new AWS account

Module 4: AWS Cloud Security

AWS account root user access versus IAM access



- **Best practice**: Do not use the AWS account root user except when necessary.
  - Access to the **account root user** requires logging in with the *email address* (and password) that you used to create the account.
- Example actions that can only be done with the account root user:
  - Update the account root user password
  - Change the AWS Support plan
  - Restore an IAM user's permissions
  - Change account settings (for example, contact information, allowed Regions)

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 an IAM user for yourself. 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. Disable and remove your account root user access keys, if they exist.
  4. Enable a password policy for users.
  5. Sign in with your new IAM user credentials.
  6. Store your account root user credentials in a secure place.

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).
  - U2F security key devices:
    - For example, YubiKey.
  - Hardware MFA options:
    - Key fob or display card offered by Gemalto.

MFA token

Securing a new AWS account: AWS CloudTrail

Step 3: Use AWS CloudTrail.

- CloudTrail tracks user activity on your account.
  - Logs all API requests to resources in all supported services your account.
- Basic AWS CloudTrail event history is enabled by default and is free.
  - It contains all management event data on latest 90 days of account activity.
- To access CloudTrail –
  1. Log in to the **AWS Management Console** and choose the **CloudTrail** service.
  2. Click **Event history** to view, filter, and search the last 90 days of events.
- **To enable logs beyond 90 days and enable specified event alerting, create a trail.**
  1. From the CloudTrail Console trails page, click **Create trail**.
  2. Give it a name, apply it to all Regions, and create a new Amazon S3 bucket for log storage.
  3. Configure access restrictions on the S3 bucket (for example, only admin users should have access).

Securing a new AWS account: Billing reports

Step 4: Enable a billing report, **such as the AWS Cost and Usage Report.**

- Billing reports provide information about your use of AWS resources and estimated costs for that use.
- AWS delivers the reports to an Amazon S3 bucket that you specify.
    - Report is updated at least once per day.
- The **AWS Cost and Usage Report** tracks your AWS usage and provides estimated charges associated with your AWS account, either by the hour or by the day.

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.

Section 4: Securing accounts

Module 4: AWS Cloud Security

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.

AWS Key Management Service (AWS KMS)

AWS Key Management Service (AWS KMS) features:

- Enables you to **create and manage encryption keys**
- Enables you to control the use of encryption across AWS services and in your applications.
- Integrates with AWS CloudTrail to log all key usage.
- Uses hardware security modules (HSMs) that are validated by Federal Information Processing Standards (FIPS) 140-2 to protect keys

AWS Key Management Service (AWS KMS)

Amazon Cognito

Amazon Cognito features:

- **Adds user sign-up, sign-in, and access control to your web and mobile applications**.
- Scales to millions of users.
- Supports sign-in with social identity providers, such as Facebook, Google, and Amazon; and enterprise identity providers, such as Microsoft Active Directory via Security Assertion Markup Language (SAML) 2.0.

Amazon Cognito

AWS Shield

- **AWS Shield** features:
  - Is a managed distributed denial of service (DDoS) protection service
  - Safeguards applications running on AWS
  - Provides always-on detection and automatic inline mitigations
  - *AWS Shield Standard* enabled for at no additional cost. *AWS Shield Advanced* is an optional paid service.
- Use it to **minimize application downtime and latency**.

AWS Shield

Section 5: Securing data on AWS

Module 4: AWS Cloud Security

Encryption of data *at rest*

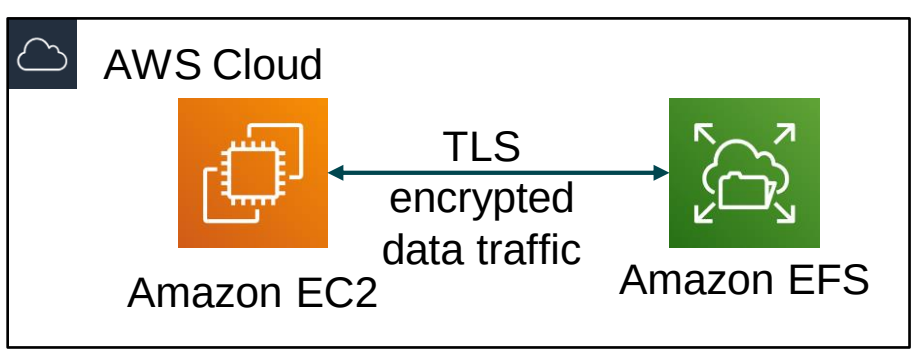
- **Encryption** encodes data with a secret key, which makes it unreadable
  - Only those who have the secret key can decode the data
  - AWS KMS can manage your secret keys

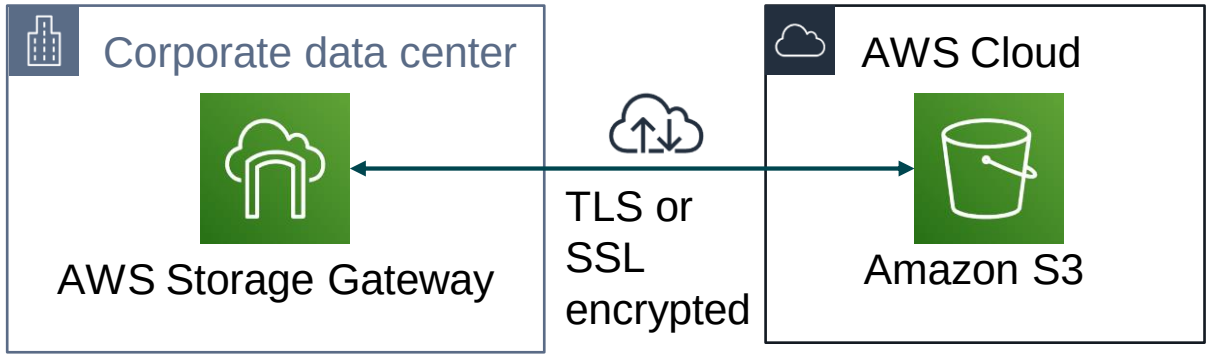
- AWS supports encryption of **data at rest**
  - Data at rest = Data stored physically (on disk or on tape)

  - 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:





Securing Amazon S3 buckets and objects

- Newly created S3 buckets and objects are private and protected by default.
- When use cases require sharing data objects on Amazon S3 –
  - It is essential to manage and control the data access.
  - Follow the **permissions that follow the principle of least privilege** and consider using Amazon S3 encryption.
- Tools and options for controlling access to S3 data include –
  - Amazon S3 Block Public Access feature: Simple to use.
  - IAM policies: A good option when the user can authenticate using IAM.
  - Bucket policies
  - Access control lists (ACLs): A legacy access control mechanism.
  - AWS Trusted Advisor bucket permission check: A free feature.

Section 6: Working to ensure compliance

Module 4: AWS Cloud Security

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

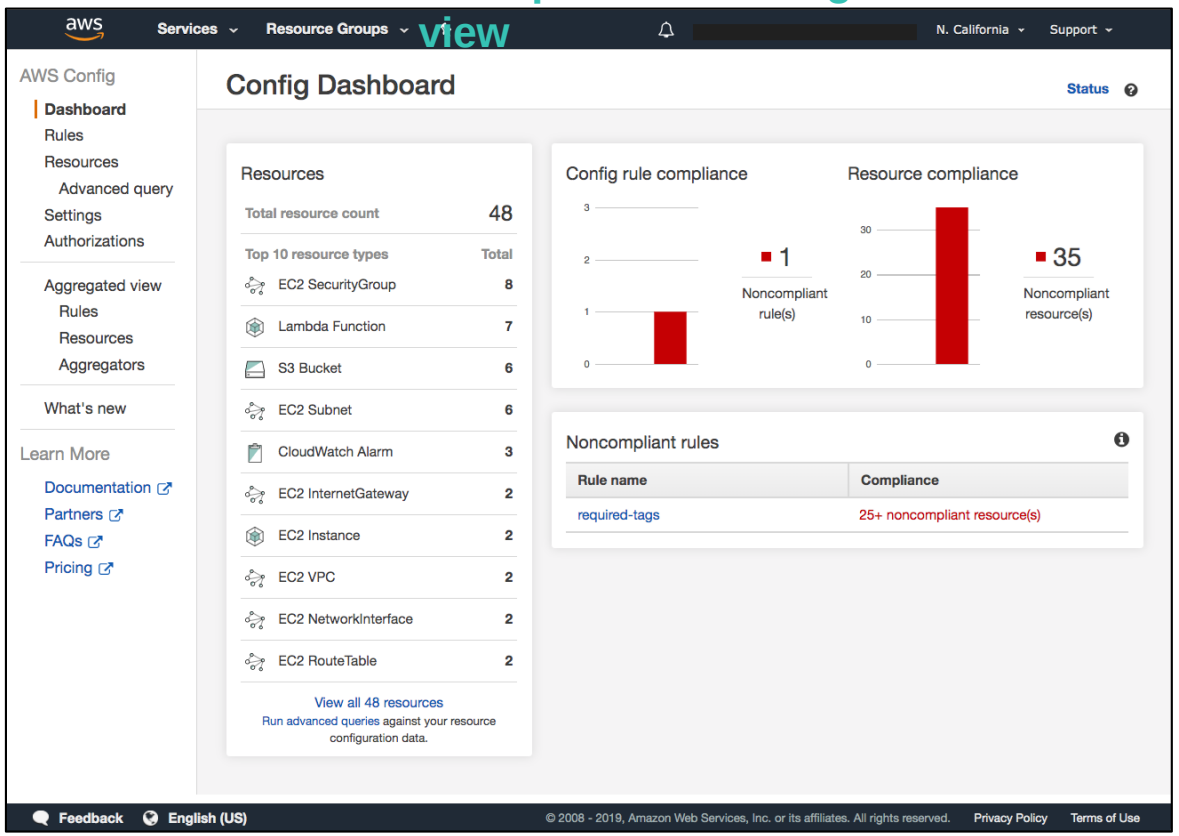
  - **Laws, regulations, and privacy**
    - AWS provides security features and legal agreements to support compliance
    - Examples: EU General Data Protection Regulation (GDPR), HIPAA

  - **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

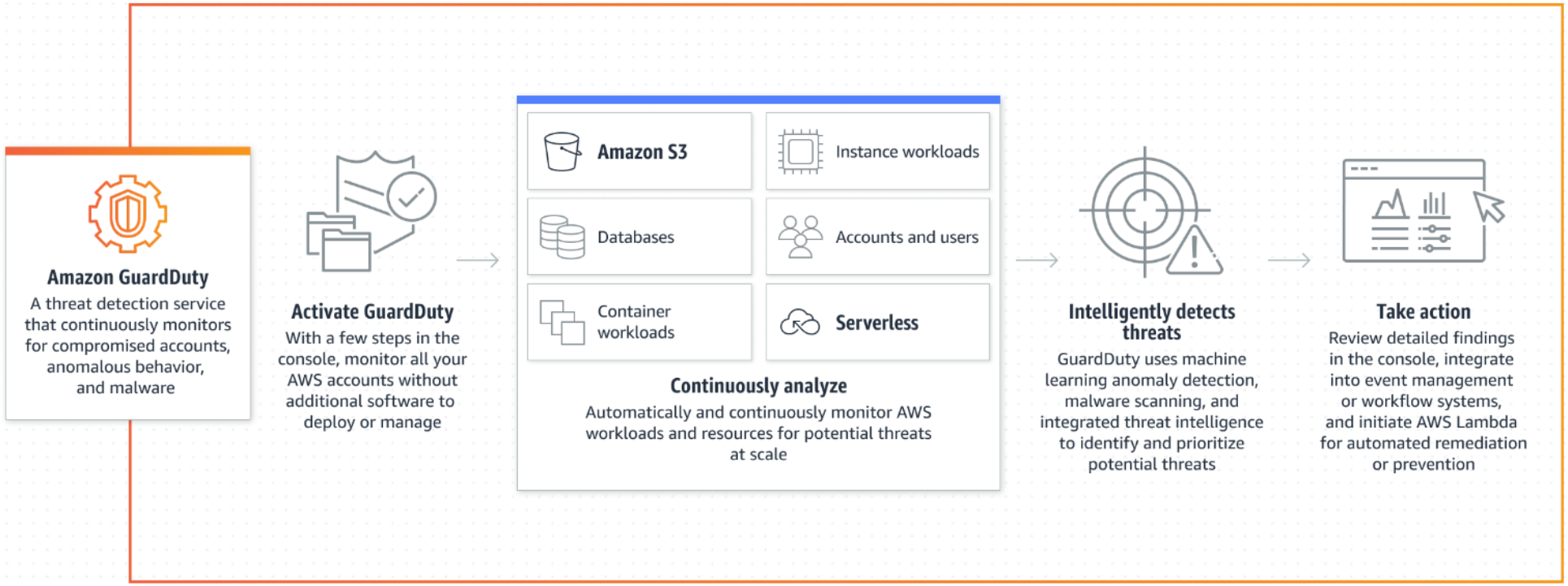
AWS Artifact

- **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**.

Amazon GuardDuty

Amazon GuardDuty
A threat detection service that continuously monitors for compromised accounts, anomalous behavior, and malware

- Amazon GuardDuty is a threat detection service that continuously monitors your AWS accounts and workloads for malicious activity and delivers detailed security findings for visibility and remediation.



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.

Module wrap-up

Module 4: AWS Cloud Security

Module summary

In summary, in this module you learned how to:

- Recognize the shared responsibility model
- Identify the responsibility of the customer and AWS
- Recognize IAM users, groups, and roles
- Describe different types of security credentials in IAM
- Identify the steps to securing a new AWS account
- Explore IAM users and groups
- Recognize how to secure AWS data
- Recognize AWS compliance programs

Complete the knowledge check

Sample exam question

Which of the following is AWS's responsibility under the AWS shared responsibility model?

Choice	Response
A	Configuring third-party applications
B	Maintaining physical hardware
C	Securing application access and data
D	Managing custom Amazon Machine Images (AMIs)

Sample exam question answer

Which of the following is AWS's responsibility under the AWS shared responsibility model?

The correct answer is B.

The keywords in the question are “AWS’s responsibility” and “AWS shared responsibility model”.

Additional resources

- AWS Cloud Security: https://aws.amazon.com/security/
- AWS Security Resources: https://aws.amazon.com/security/security-learning/?cards-top.sort-by=item.additionalFields.sortDate&cards-top.sort-order=desc&awsf.Types=*all
- AWS Security Blog: https://aws.amazon.com/blogs/security/
- Security Bulletins : https://aws.amazon.com/security/security-bulletins/?card-body.sort-by=item.additionalFields.bulletinId&card-body.sort-order=desc&awsf.bulletins-flag=*all&awsf.bulletins-year=*all
- Vulnerability and Penetration testing: https://aws.amazon.com/security/penetration-testing/
- AWS Well-Architected Framework – Security pillar: https://d1.awsstatic.com/whitepapers/architecture/AWS-Security-Pillar.pdf
- AWS documentation - IAM Best Practices: https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html

Thank you

Corrections, feedback, or other questions?
Contact us at https://support.aws.amazon.com/#/contacts/aws-academy.
All trademarks are the property of their owners.