

# CLOUD COMPUTING

## Session 5 :

### AWS Well Architected Framework - Security Pillar

**APROBAT**

Conf. univ. dr. ing. IUSTIN PRIESCU - UTM  
Dr. ing. Sebastian NICOLAESCU - Verizone Bussines-US

# AWS Well Architected Framework

The goal of this framework is to enable customers to:

- 📦 Assess and improve their architectures
- 📦 Better understand the business impact of their design decisions

It provides a **set of questions developed by AWS experts** to help customers think critically about their architecture.

It asks, "**Does your infrastructure follow best practices?**"

# AWS Well Architected Framework

Architects should leverage the AWS Well-Architected Framework in order to:

- 📦 Increase awareness of **architectural best practices**.
- 📦 Address **foundational areas** that are often neglected.
- 📦 **Evaluate** architectures using a consistent set of principles.

# AWS Well Architected Framework

The AWS Well-Architected Framework **does not** provide:

- 📦 Implementation details
- 📦 Architectural patterns
- 📦 Relevant case studies

However, it **does** provide:

- 📦 Questions centered on critically understanding architectural decisions
- 📦 Services and solutions relevant to each question
- 📦 References to relevant resources

# Pillars of the AWS Well Architected Framework

## Security

Protect and monitor systems.



## Reliability

Recover from failure and mitigate disruption.



## Performance Efficiency

Use resources sparingly.



## Cost Optimization

Eliminate unneeded expense.



# AWS Well Architected Framework: Security Pillar



- 📦 The ability to protect:
  - Information
  - Systems
  - Assets
- 📦 While delivering business value through:
  - Risk assessments
  - Mitigation strategies

# Best practice: Secure Your Infrastructure Everywhere

*Build security into every layer of your infrastructure.*

Physical data centers typically rely on security at the perimeter. AWS enables you to implement security at the perimeter as well as **within and between your resources**.

## Things to consider:

- 📦 Isolate parts of your infrastructure
- 📦 Encrypt data in transit and at rest
- 📦 Enforce access control granularly, using the principle of least privilege
- 📦 Use multi-factor authentication
- 📦 Leverage managed services
- 📦 Log access of resources
- 📦 Automate your deployments to keep security consistent

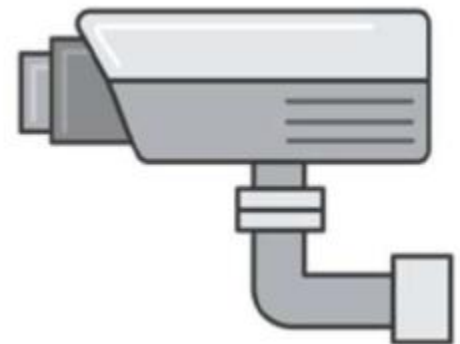
# Principles of the Security Pillar



# Security Principles: Apply Security At All Layers

Instead of just running security appliances (e.g., firewalls) at the edge of your infrastructure, **use firewalls and other security controls on all of your resources:**

- 📦 Every virtual server
- 📦 Every load balancer
- 📦 Every network subnet



# Security Principles: Enable Traceability

Log and audit all action and changes to your environment and access to your services.

# Security Principles:

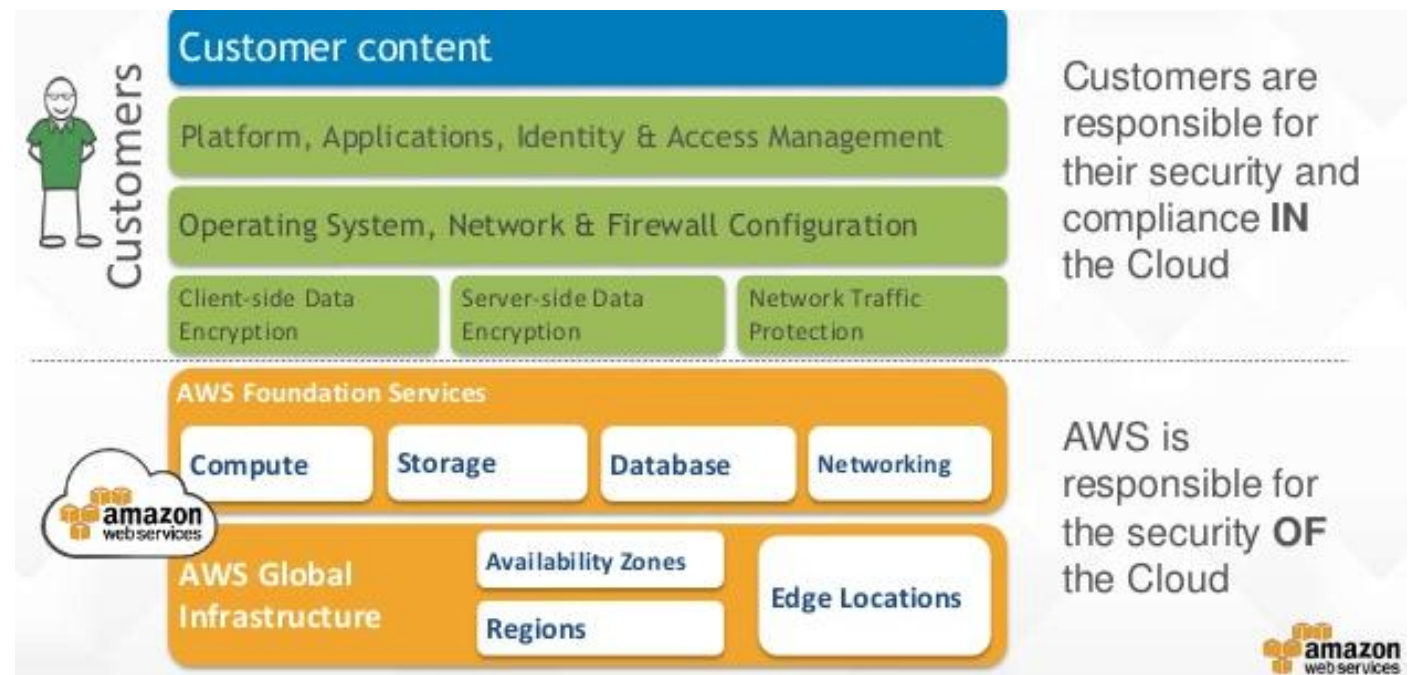
## Automate Response to Security Events

Monitor and automatically trigger responses to event-driven or condition-driven alerts.

# Security Principles: Focus on Securing **Your** System

With the AWS Shared Responsibility Model:

- 📦 AWS provides secure infrastructure and services.
- 📦 You can focus on securing your application, data, and operating systems.

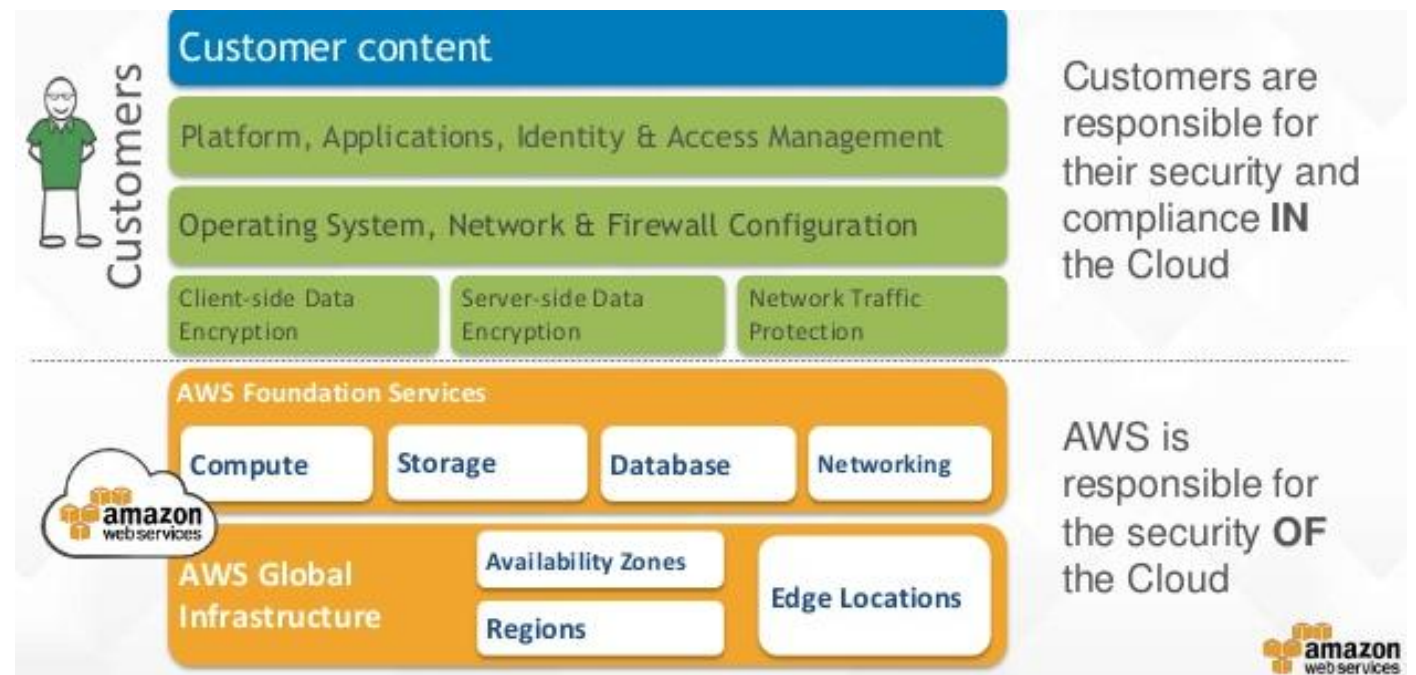


<https://aws.amazon.com/compliance/shared-responsibility-model>

# Security Principles: Focus on Securing **Your** System

With the AWS Shared Responsibility Model:

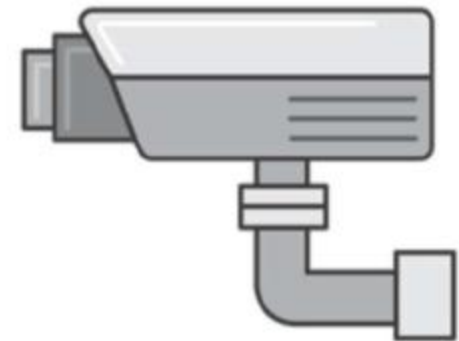
- 📦 AWS provides secure infrastructure and services.
- 📦 You can focus on securing your application, data, and operating systems.





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

# Security Principles: Automate Best Security Practices

- ❏ Use **software-based security mechanisms** to improve your ability to securely scale more rapidly and cost-effectively.
- ❏ Create and save a **custom baseline image** of a virtual server, and then use that image automatically on each new server you launch.
- ❏ Create an **entire infrastructure** that is defined and managed in a **template**.



# Key Services for Security

Areas	Key Services				
Data protection	 Elastic Load Balancing	 Amazon EBS	 Amazon S3	 Amazon RDS	 AWS Key Management Service (KMS)
Privilege management	 AWS IAM	 MFA token			
Infrastructure protection	 Amazon VPC				
Detective controls	 AWS CloudTrail	 AWS Config	 Amazon CloudWatch		

# Preventing Common Security Exploits



# DDoS Attacks

A Denial of Service (DoS) attack attempts to make your website or application **unavailable** to your end users.

To achieve this, attackers use a variety of techniques that **consume network or other resources**, thus interrupting access for legitimate end users.

The attackers use **multiple hosts** to orchestrate an attack against a target.

# DDoS Protection

Protecting against attacks is a **shared responsibility** between AWS and you.

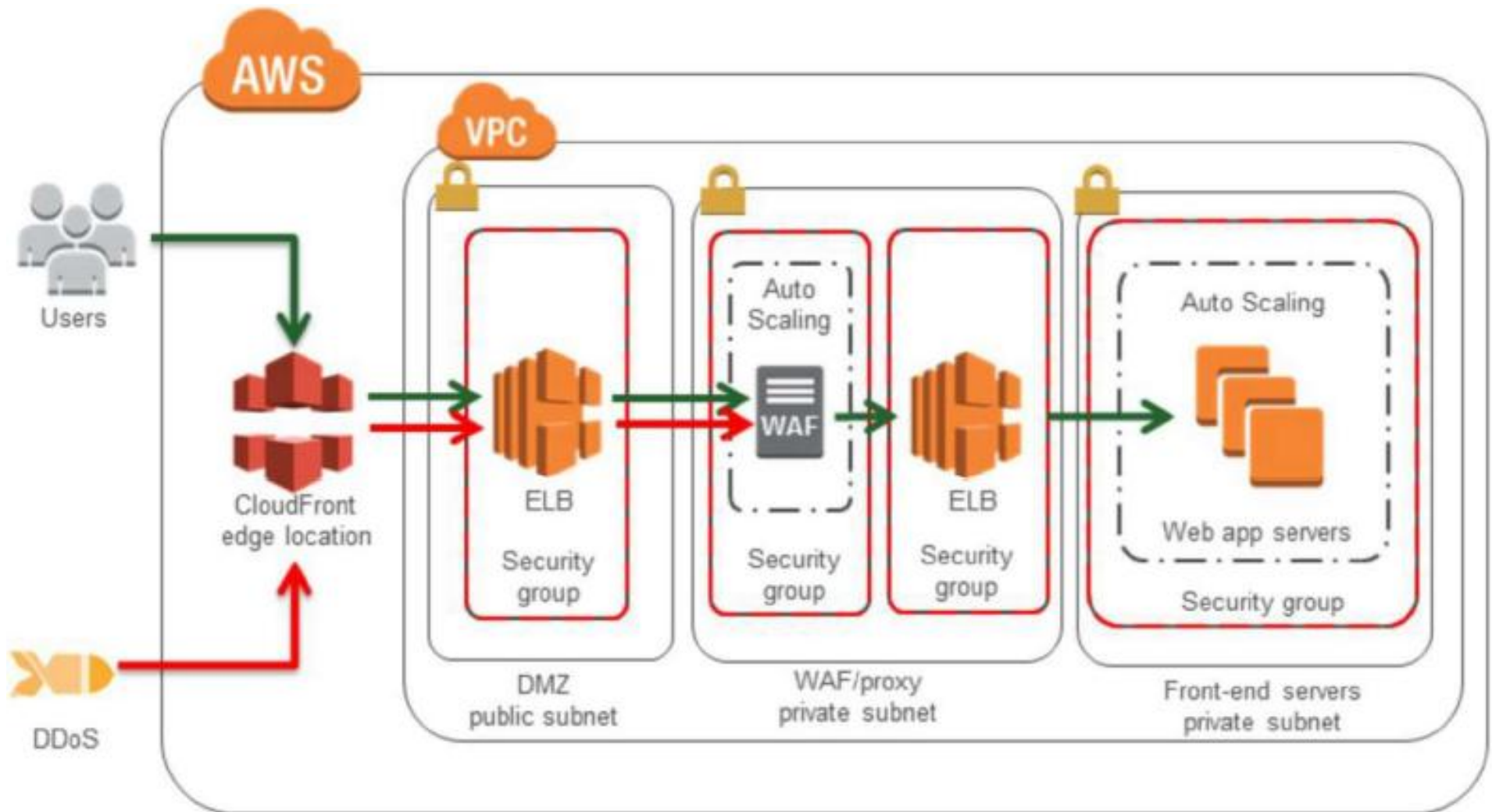
## AWS

- AWS API endpoints are hosted on large, Internet-scale, world-class infrastructure.
- Proprietary DDoS mitigation techniques are used.
- AWS networks are multi-homed across a number of providers to achieve Internet access diversity.

## Customer

- Front your application with AWS Services
- Safeguard exposed resources
- Minimize the attack surface
- Evaluate soft limits and request increases ahead of time
- Learn normal behavior
- Create a plan for attacks

# DDoS Mitigation Example



# Prevent Common Exploits with Amazon Inspector

- ❏ Is an **automated security assessment** service that assesses applications for:
  - Vulnerabilities
  - Deviations from best practices
- ❏ Produces a **detailed report with prioritized steps for remediation** after performing the assessment.



# Amazon Inspector Rules

- Amazon Inspector includes a knowledgebase with **hundreds of rules** that are
  - Mapped to:
    - Common security compliance standards
    - Vulnerability definitions
  - Regularly updated by AWS security researchers.

## Examples of Built-In Rules



Remote root login  
being enabled



Vulnerable software  
versions installed



# Amazon Inspector Prioritized List of Findings

## Amazon Inspector - Findings

Inspector findings are potential security issues discovered during Inspector's assessment of the specified application. [Learn more.](#)

Add/Edit attributes

Last updated on September 24, 2015 4:12:42 PM (20m ago)



Filter

Viewing 1-10 of

<input type="checkbox"/>	Severity	Application	Assessment	Rule package	Finding
<input type="checkbox"/>	High ⓘ	Customer Processing	Comprehensive-Assessment	Authentication Best Practices	Instance i-aac4c46f is
<input type="checkbox"/>	High ⓘ	Customer Processing	Comprehensive-Assessment	Common Vulnerabilities and Ex...	Instance i-aac4c46f is
<input type="checkbox"/>	High ⓘ	Customer Processing	Comprehensive-Assessment	Authentication Best Practices	No password complex
<input type="checkbox"/>	Informational ⓘ	Customer Processing	Initial app	PCI DSS 3.0 Readiness	Instance i-aac4c46f w
<input type="checkbox"/>	Informational ⓘ	Customer Processing	Initial app	PCI DSS 3.0 Readiness	The machine i-aac4c4
<input type="checkbox"/>	Informational ⓘ	Customer Processing	Comprehensive-Assessment	Operating System Security Best...	No potential security e
<input type="checkbox"/>	Informational ⓘ	Customer Processing	Comprehensive-Assessment	PCI DSS 3.0 Readiness	The machine i-aac4c4
<input type="checkbox"/>	Informational ⓘ	Customer Processing	Comprehensive-Assessment	Network Security Best Practices	No potential security e
<input type="checkbox"/>	Informational ⓘ	Customer Processing	Comprehensive-Assessment	PCI DSS 3.0 Readiness	Instance i-aac4c46f w
<input type="checkbox"/>	Informational ⓘ	Customer Processing	Initial app	PCI DSS 3.0 Readiness	A machine with Instan

# Amazon Inspector Detailed Remediation Recommendations

## Finding for application - Customer Processing

**Application name** Customer Processing

**Assessment name** Comprehensive-Assessment

**Assessment start** Today at 3:51 PM (GMT-4)

**Assessment end** Today at 4:12 PM (GMT-4)

**Status** COMPLETED

**Rule package** [Authentication Best Practices](#)

**Finding** Instance i-aac4c46f is configured to allow users to log in with root credentials over SSH

**Severity** High ⓘ

# Securing Data



# CloudFront Custom SSL Support

By default, **your content is delivered to viewers over HTTPS** by using a CloudFront distribution domain name such as <https://dxxxxx.cloudfront.net/image.jpg>.

**Custom SSL certificate support** features let you use your own domain name and your own SSL certificate.

## Server Name Indication (SNI) Custom SSL

- Allows multiple domains to serve SSL traffic over the same IP address.

## Dedicated IP Custom SSL

- To deliver content to browsers that do not support SNI.

# CloudFront: Advanced SSL Feature Support

## High-security ciphers

- 📦 Improve the security of HTTPS connections.

## Perfect Forward Secrecy

- 📦 Provides additional safeguards against eavesdropping of encrypted data through a unique random session key.

## OCSP Stapling

- 📦 Improves the time taken for individual SSL/TLS handshakes by moving the Online Certificate Status Protocol (OCSP) check.

## Session Tickets

- 📦 Helps speed up the time spent restarting or resuming an SSL/TLS session.

# How to make content private

- ❏ Restrict access to objects in your Amazon S3 bucket.
- ❏ Require that users use signed URLs.
  - Create CloudFront key pairs for your trusted signers.
  - Write the code that generates signed URLs.
    - Typically, you'll write an application that automatically generates signed URLs.
    - Alternatively, use a web interface to create signed URLs.
  - Add trusted signers to your distribution.
    - **Note:** Once you add a trusted signer to your distribution, users must use signed URLs to access the corresponding content.

# Origin Access Identity to Restrict Access

Restrict access to Amazon S3 content by creating an **origin access identity** (OAI), which is a special CloudFront user.

- 📦 CloudFront origin access identity gets the objects from Amazon S3 on your users' behalf.
- 📦 Direct access to the objects through Amazon S3 URLs will be denied.

Procedure:

1. Create an origin access identity and add it to your distribution.
2. Change the permissions either on your Amazon S3 bucket or the objects in your bucket so that only the origin access identity has read permission.

# Control Access to CloudFront APIs via IAM Policy

Control a user's API access to CloudFront with IAM policies.

**Group policy example:**

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": ["cloudfront:*"],
    "Resource": "*",
    "Condition": {
      "Bool": {
        "aws:SecureTransport": "true"
      }
    }
  }]
}
```

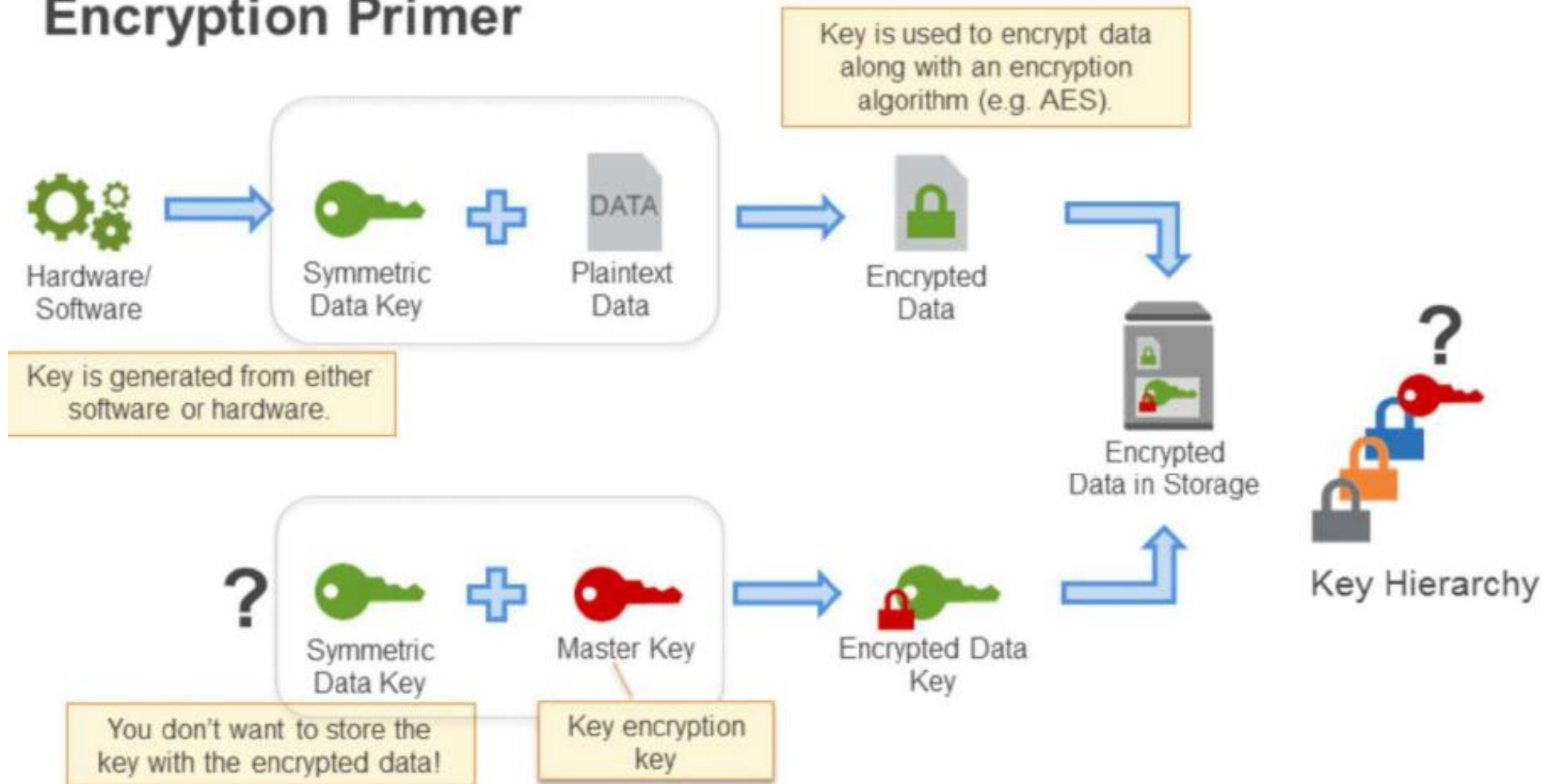
Grant permission to access all CloudFront actions for the group this policy is attached to...

...with condition that actions require use of SSL/TLS

# Encrypting Data



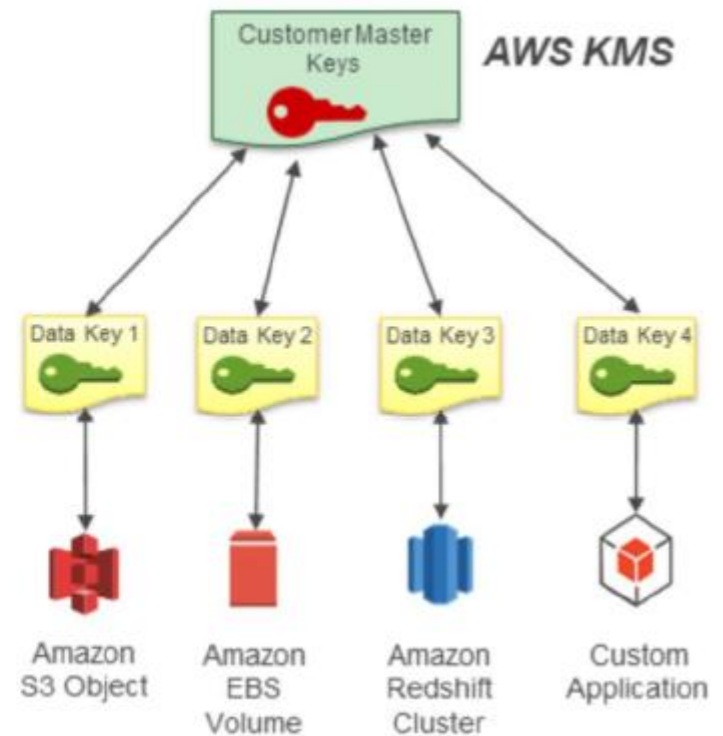
# Encryption Primer



# AWS Key Management Service (KMS)

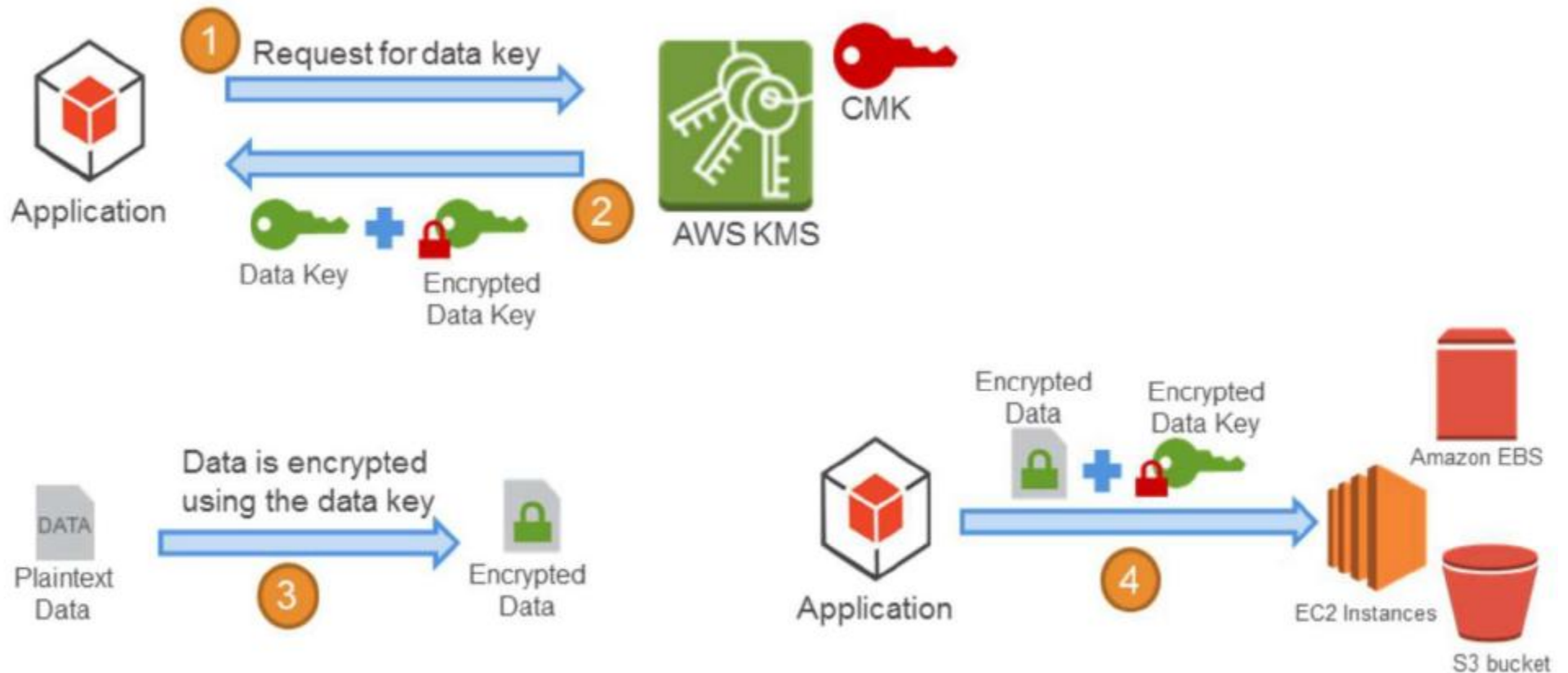
AWS KMS is a managed encryption service that enables you to easily encrypt your data.

- Two-tiered key hierarchy using envelope encryption.
- Data keys are unique.
- AWS KMS master keys encrypt data keys.
- AWS KMS master keys never leave the AWS KMS system.





# AWS KMS: CMK and Data keys



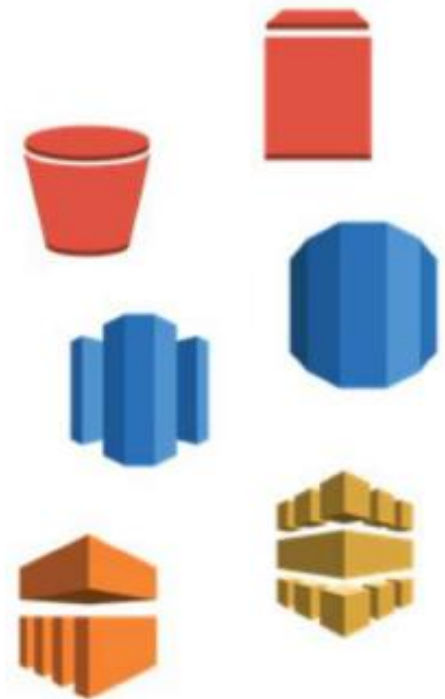
# AWS KMS Benefits

- 📦 Only data keys are available directly to the customer, and these are unique to each item encrypted.
  - If one was compromised, it would not allow decryption of other objects.
- 📦 The risk of a compromised data key is limited.
- 📦 The performance for encrypting large data is improved.
- 📦 It is easier to manage a small number of master keys than millions of data keys.

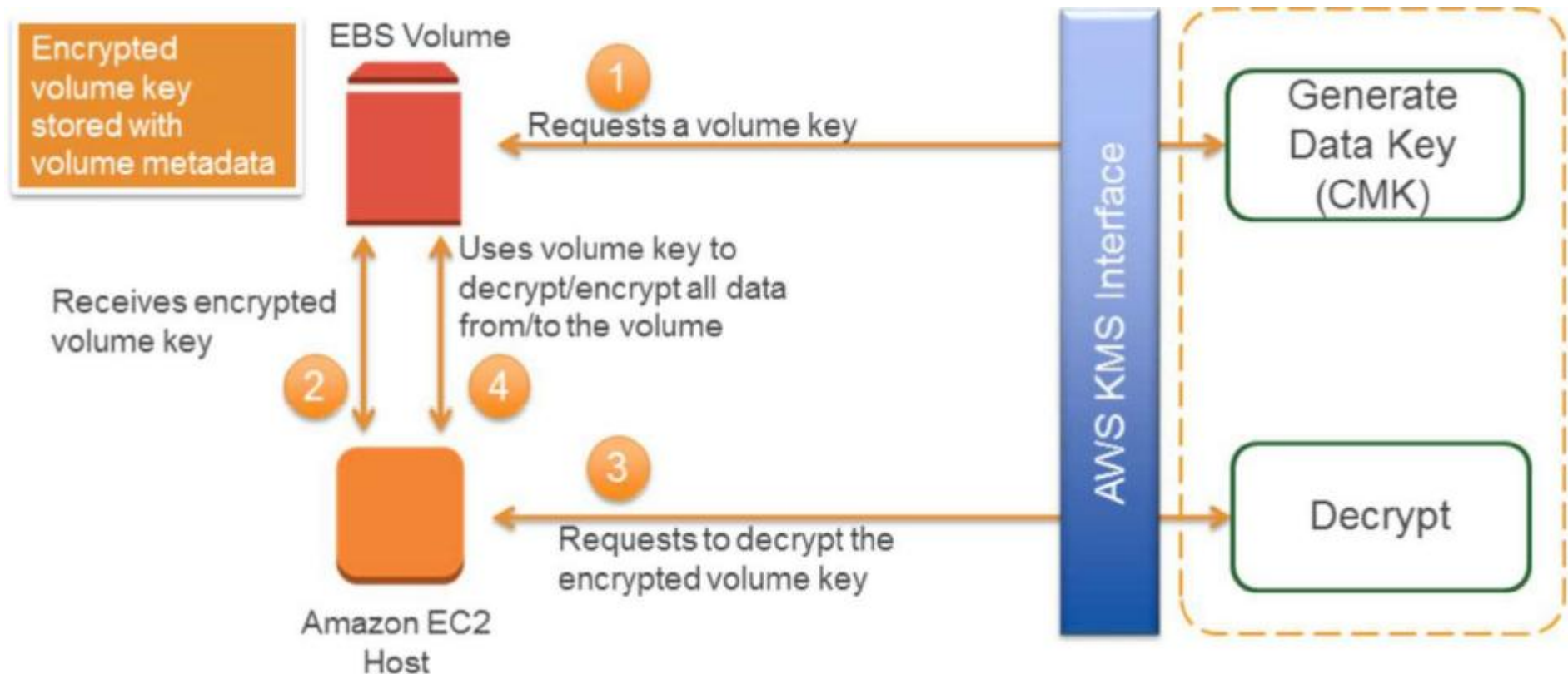
# AWS KMS Integration with Other Services

The following services are integrated with AWS KMS to simplify data encryption:

- Amazon Elastic Block Store (EBS)
- Amazon Simple Storage Service (S3)
- Amazon RDS
- Amazon Redshift
- Amazon Elastic Transcoder
- Amazon WorkMail
- Amazon EMR



# EBS Volume Encryption Using KMS



# AWS Cloud HSM

- 📦 Protects your cryptographic keys using a dedicated, tamper-resistant Hardware Security Module (HSM).
- 📦 Helps you comply with strict cryptographic key management requirements.
  - Designed to meet [FIPS 140-2](#) and [Common Criteria EAL4+](#) standards.

# AWS CloudHSM vs KMS

AWS CloudHSM	AWS KMS
<b>Single-tenant</b> HSM	Multi-tenant AWS service
Customer-managed durability and availability	Highly available and durable key storage and management
Customer managed root of trust	AWS managed root of trust
Broad third-party app support	Broad support for AWS services
Symmetric and asymmetric options	Symmetric encryption only

# S3 Encryption



# S3 Encryption Options

Data stored in Amazon S3 is private by default, requires AWS credentials for access

- 📦 Access to Amazon S3 can be over HTTP or HTTPS
- 📦 Amazon S3 logging allows auditing of access to all objects
- 📦 Amazon S3 supports access control lists and policies for every bucket, prefix (directory/folder), and object

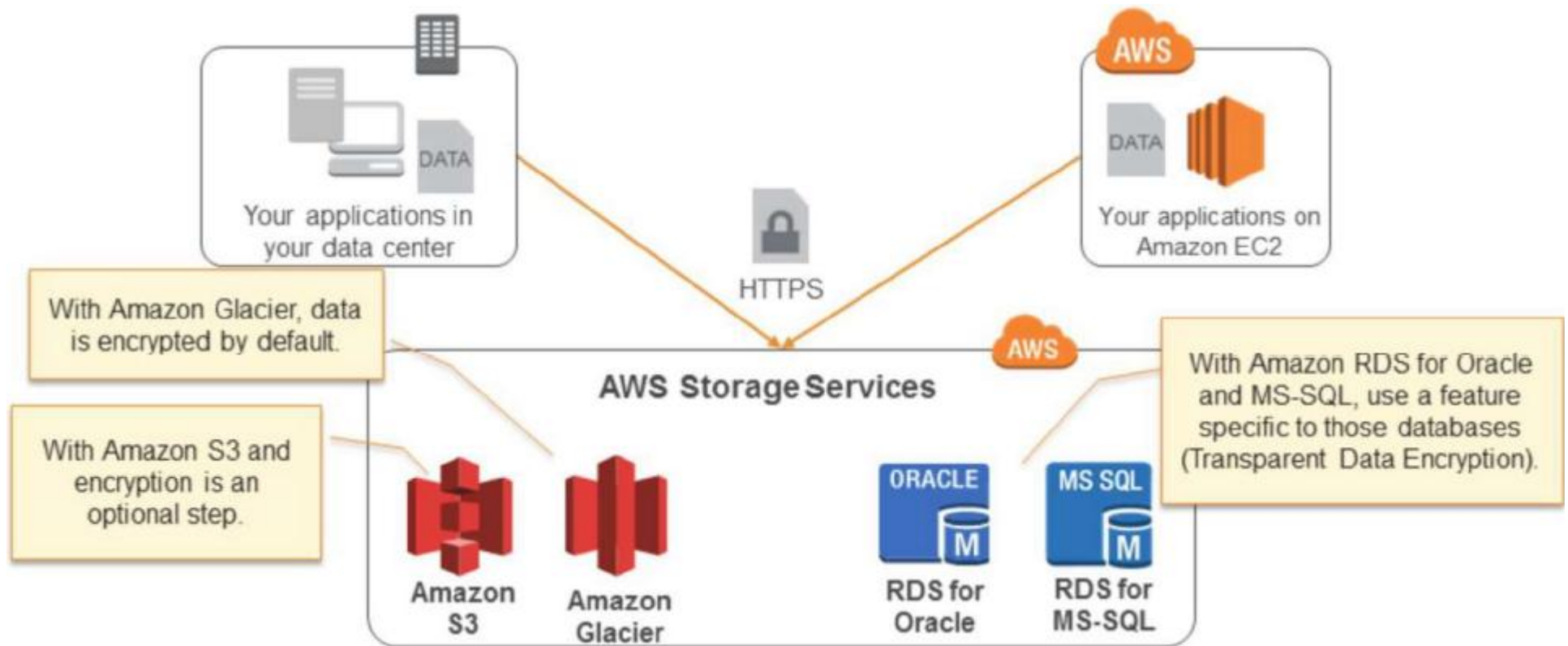
Amazon S3 provides server-side encryption (AES-256) using AWS maintained keys or customer provided keys

- 📦 AWS encryption keys are further encrypted with a rotating key

Can also encrypt data before storage in Amazon S3 (client-side encryption)



# AWS Server-Side Encryption (SSE)



# AWS RDS Security Groups

Control traffic in and out of a DB instance.

No network access by default.

Three types:

- 📦 DB security groups

- Control access to a DB instance that is not in a VPC.

- 📦 VPC security groups

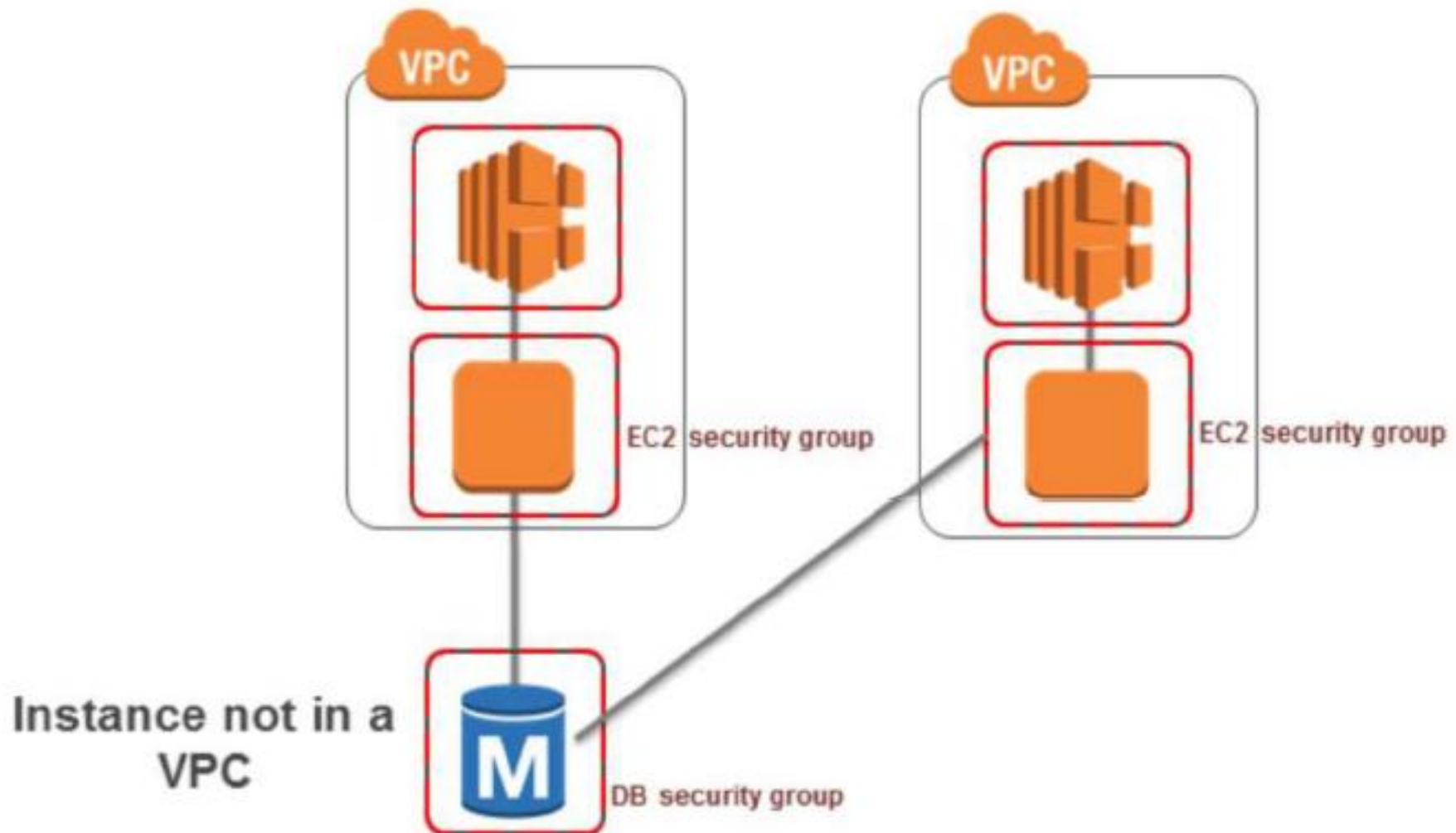
- Control access to a DB instance inside a VPC.

- 📦 EC2 security groups

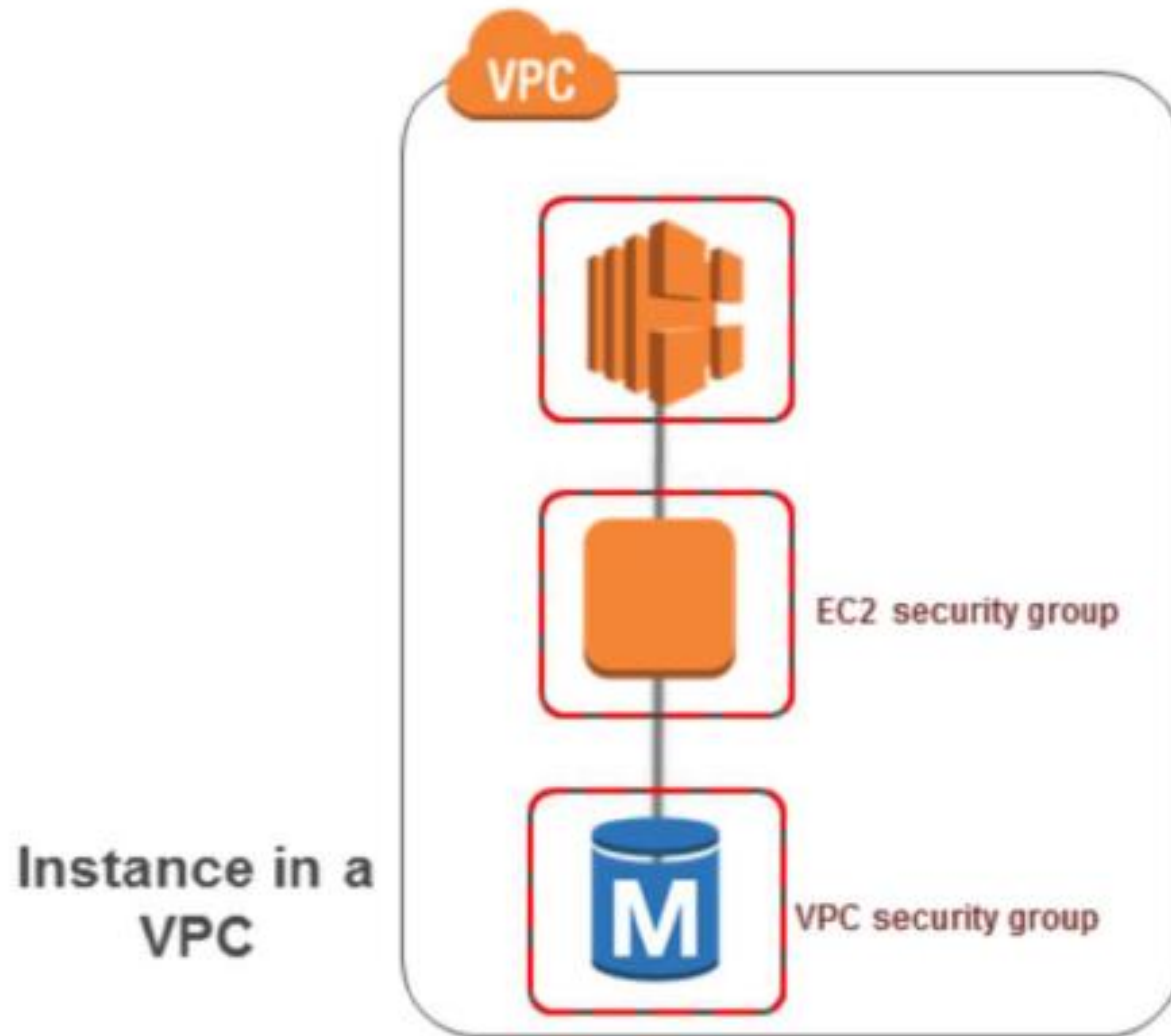
- Control access to EC2 instances.



# RDS Security Groups



# RDS Security Groups



# RDS Connections Encryption

- ❏ You are responsible for encryption of your **data in-transit**.
- ❏ Use **SSL/TLS** to encrypt connections between applications and DB Instances.
- ❏ Configure your DB instance to **accept only encrypted connections**.
- ❏ Encryption of data connections helps meet **compliance standards**.

# RDS Resources Encryption

- ❏ You are responsible for encrypting your **data at-rest**.
- ❏ Use encryption option to encrypt data at-rest.
  - Underlying storage for the instance, its backups, logs, Read Replicas, and snapshots.
  - Uses AES-256 encryption algorithm.
- ❏ Encrypting data-at-rest helps meet compliance standards.
- ❏ Automatic handling of authentication and decryption.
- ❏ Supports TDE for Oracle and SQL Server DB instances.
- ❏ Use AWS Key Management Service to manage keys.

# Authentication



# AWS Active Directory Service

AWS Directory Service is a managed service to:

- 📦 Run Microsoft AD as a managed service within AWS Directory Service
- 📦 Connect your AWS resources with an existing on-premises Microsoft Active Directory (AD Connector)
- 📦 Set up a new, stand-alone directory in the AWS Cloud (Simple AD)

AWS Directory Service allow use of existing corporate credentials for:

- 📦 Accessing AWS services (e.g. Amazon WorkSpaces, and Amazon WorkDocs)
- 📦 Accessing the AWS Management Console through IAM Roles

# AWS Active Directory Service

## Options:

- 📦 Run Microsoft AD as a managed service within AWS Directory Service:
  - Powered by Windows Server 2012 R2
  - Created as a highly available pair of domain controllers connected to your VPC.
- 📦 Use AD Connector:
  - Connect to your on-premises Active Directory via VPC VPN connection or AWS Direct Connect
  - Users access AWS applications with existing credentials
  - Integrate with existing RADIUS-based MFA solutions
- 📦 Use Simple AD:
  - Launch managed stand-alone directories powered by Samba 4 Active Directory Compatible Server
  - Supports common AD features
  - Provides audit trails and event logs

# AWS Security Token Service (STS)

STS is a lightweight web service that enables you to request **temporary, limited-privilege credentials** for IAM users or for users that you authenticate (federated users).

Allow trusted entity to assume a role by calling the `AssumeRole` APIs of STS

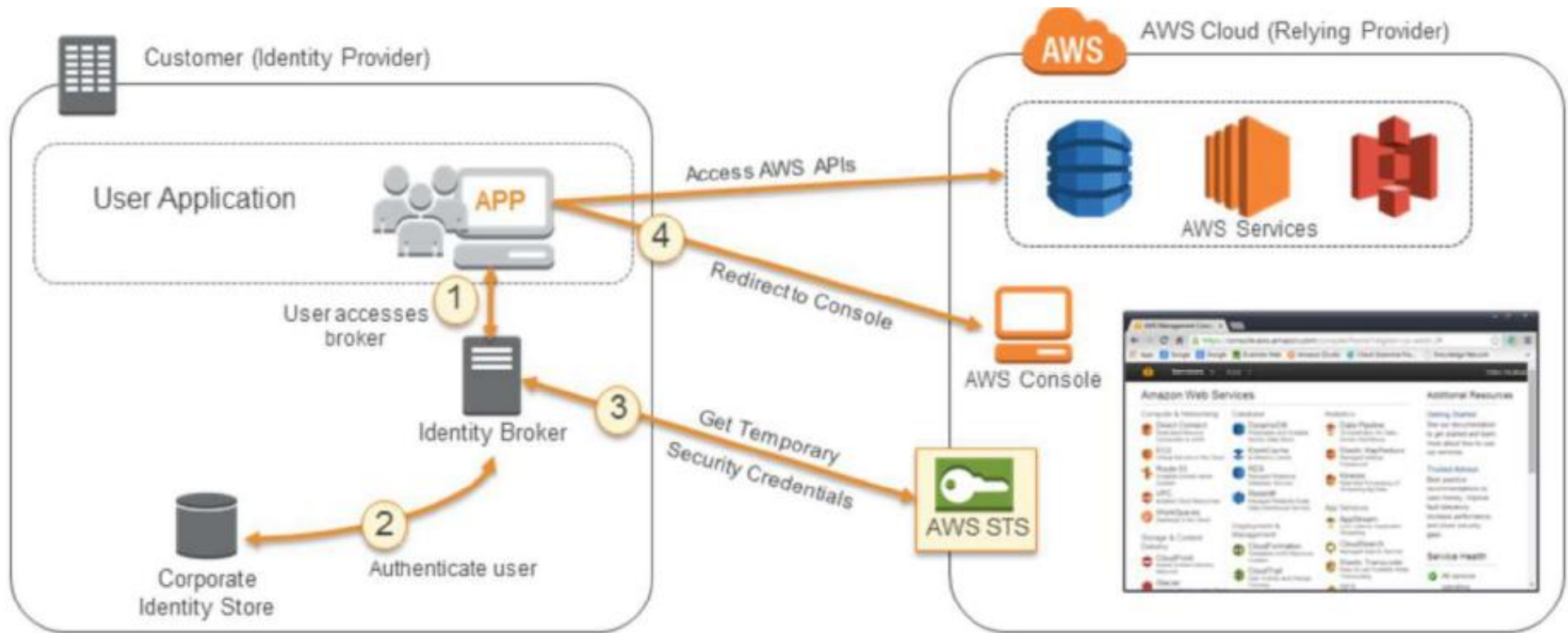
- 📦 To view security credentials of a running Amazon EC2 instance, use the following Instance Metadata Service (IMDS) URL:  
`http://169.254.169.254/latest/meta-data/iam/security-credentials/`

# Federated Users

Authenticate users to your own identity store:

- 📦 You write an “identity broker application.”
- 📦 Users authenticate to your identity broker.
- 📦 Your identity broker provisions temporary credentials via STS.
- 📦 **Single Sign-On (SSO):** Temporary credentials can be used to sign user directly into the AWS Management Console.

# Use Case: STS Identity Broker



# SSO Federation via SAML

Amazon STS supports **SAML 2.0**.

## **Benefits:**

- 📦 Open standards
- 📦 Quicker and easier to implement federation
- 📦 Leverage existing identity management software to manage access to AWS resources
- 📦 **No coding required**



# SSO Federation via SAML

Amazon STS supports **SAML 2.0**.

## Benefits:

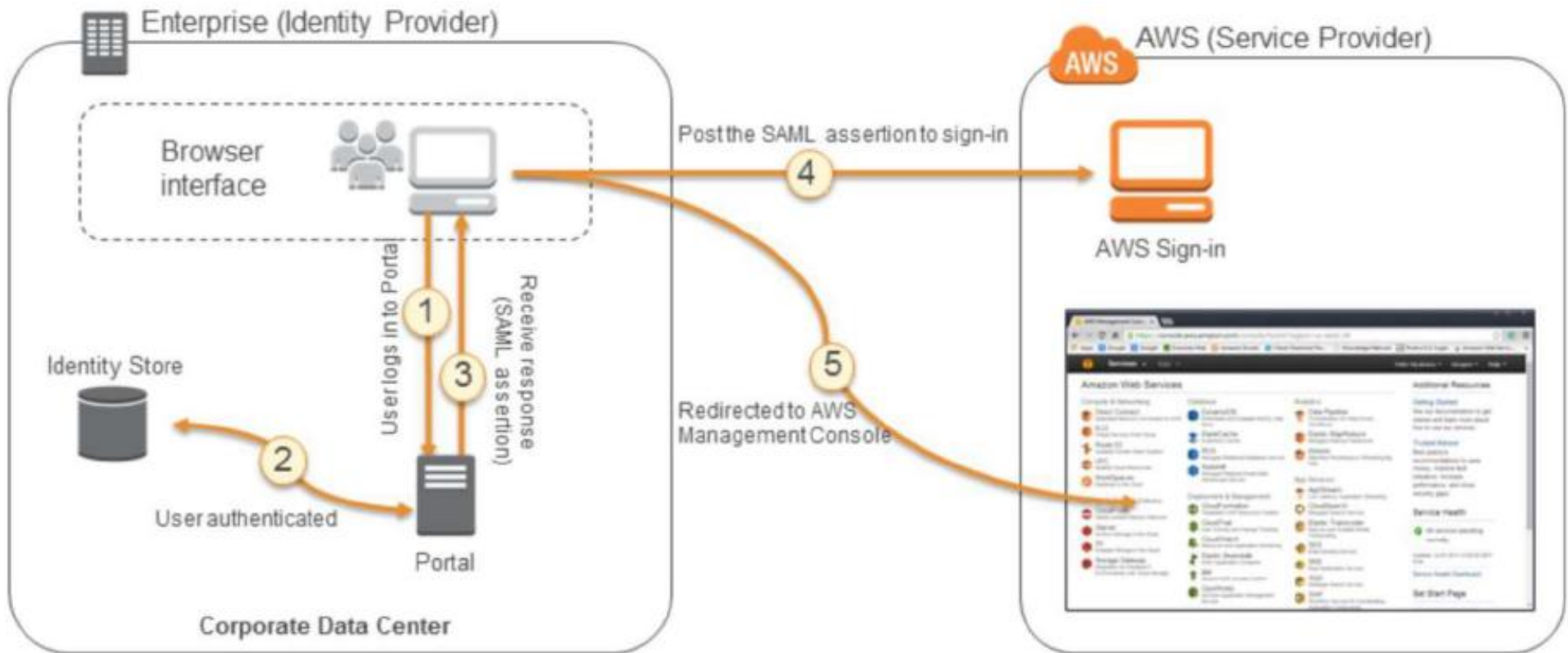
- 📦 Open standards
- 📦 Quicker and easier to implement federation
- 📦 Leverage existing identity management software to manage access to AWS resources
- 📦 **No coding required**

## AWS Management Console SSO:

- 📦 IdP-initiated web SSO via SAML 2.0 using the HTTP-POST binding (web SSO profile)
- 📦 New sign-in URL that greatly simplifies SSO:  
[https://signin.aws.amazon.com/saml<SAML\\_AuthN\\_response>](https://signin.aws.amazon.com/saml<SAML_AuthN_response>)
- 📦 API federation using new `assumeRoleWithSAML` operation



# SSO Federation via SAML



# Web Identity Federation

Use STS API, `AssumeRoleWithWebIdentity`

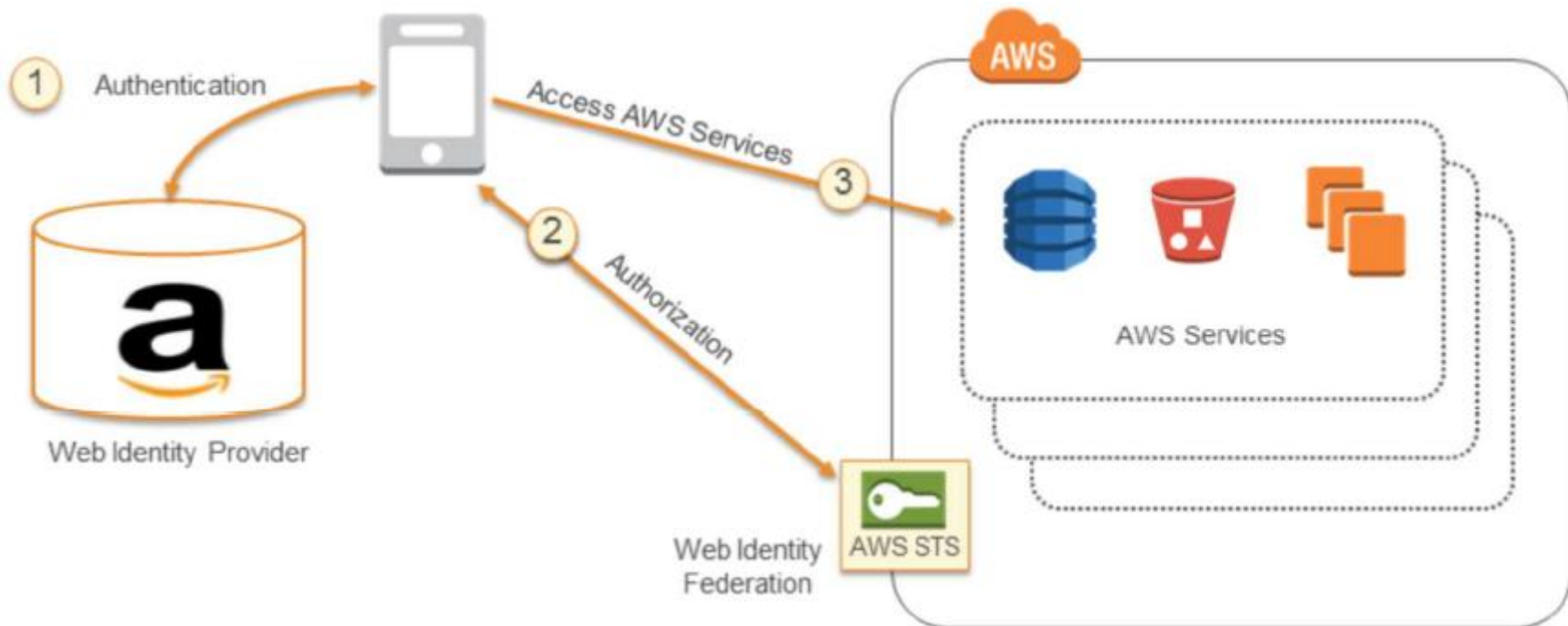
- 📦 Lets you request temporary security credentials to access AWS resources.

Supported web identity providers:

- 📦 Amazon
- 📦 Google
- 📦 Facebook

A mobile app can be developed without server-side code and without distributing long-term credentials with the mobile app.

# Web Identity Federation



# Questions

*How do you encrypt and protect  
your data at rest?*

# How do you encrypt and protect your data at rest?

## Best practice

- 📦 Use AWS service-specific controls, such as:
  - Amazon S3 SSE
  - Amazon EBS Encrypted Volumes
  - Amazon RDS Transparent Data Encryption (TDE)
- 📦 Use client-side techniques, such as:
  - SDK-supported
  - OS-supported
  - Windows Bitlocker
  - dm-crypt
- 📦 Use a solution from the AWS Marketplace or APN Partner

*How do you encrypt and protect  
your data in transit?*



# How do you encrypt and protect your data in transit?

## Best practice

- 📦 SSL/TLS enabled AWS APIs
- 📦 SSL/TLS or equivalent is used for communication
- 📦 VPN based solution
- 📦 Private connectivity (AWS Direct Connect)
- 📦 AWS Marketplace solution

*How do you protect AWS root  
account credentials?*

# How do you protect AWS root account credentials?

## Best practice

- 📦 Only use AWS root account credentials for minimal required activities
- 📦 Associate an MFA hardware device with the AWS root account
- 📦 Use an AWS Marketplace solution

*How are you defining roles and responsibilities of system users  
to control human access to the AWS Management Console  
and API?*

*How are you defining roles and responsibilities of system users to control human access to the AWS Management console and APIs?*

### Best practice

- ❏ IAM users and groups
- ❏ SAML integration
- ❏ Web Identity Federation
- ❏ AWS Security Token Service (STS)
- ❏ IAM roles for cross-account access
- ❏ AWS Marketplace solution
- ❏ Define and enforce employee life-cycle policies
- ❏ Clearly define users, groups, and roles.
- ❏ Grants only the minimum privileges needed to accomplish business requirements.

*How are you limiting automated access to AWS resources?  
(e.g. applications, scripts, and/or third-party tools or services)*

*How are you limiting automated access to AWS resources?  
(e.g. applications, scripts, and/or third-party tools and services)*

### **Anti-pattern**

- 📦 Hard-coding the credential into scripts and source code

### **Best practice**

- 📦 IAM roles for Amazon EC2
- 📦 IAM user credential
- 📦 SAML Integration
- 📦 AWS Security Token Services (STS)
- 📦 OS-specific controls for EC2 instances
- 📦 AWS Marketplace solutions



*How are you managing keys and credentials?*

## *How are you managing keys and credentials?*

### **Anti-pattern**

- 📦 Hard-coding secret keys and credentials into scripts and source code

### **Best practice**

- 📦 Use:
  - An appropriate key and credential rotation policy
  - AWS CloudHSM
  - AWS server-side techniques with AWS managed keys
  - AWS Marketplace solutions

*How are you enforcing network and host-level boundary protection?*

## *How are you enforcing network and host-level boundary protection?*

### **Best practice**

- 📦 Enforce role-based access using security groups with minimal authorizations.
- 📦 Run the system in one or more VPCs.
- 📦 Trusted VPC access is via a private mechanism, such as
  - Virtual Private Network (VPN)
  - Ipsec tunnel
  - AWS Direct Connect
  - AWS Marketplace solution
- 📦 Define and enforce employee life-cycle policies
- 📦 Clearly define users, groups, and roles.
- 📦 Grants only the minimum privileges needed to accomplish business requirements.

*How are you enforcing AWS service level protection?*

## *How are you enforcing AWS service level protection?*

### **Best practice**

- ❏ Configure credentials with **least privilege**.
- ❏ Have a **separation of duties**.
- ❏ **Audit permissions** periodically.
- ❏ Define and use **service-specific requirements**.
- ❏ Define **resource requirements** for sensitive API calls, such as requiring:
  - MFA authentication
  - Encryption
- ❏ Use an **AWS Marketplace** solution

*How are you protecting the integrity of the operating system on your Amazon EC2 instances?*

## *How are you protecting the integrity of the operating system on your Amazon EC2 instances?*

### **Best practice**

- 📦 Use controls for EC2 instances, including:
  - File integrity
  - Host-based intrusion detection
- 📦 Use a solution from:
  - The AWS Marketplace
  - An APN Partner
- 📦 Use custom AMIs or configuration management tools (i.e., Puppet or Chef) that are secured by default.



*How are you capturing and analyzing AWS logs?*

# *How are you capturing and analyzing AWS logs?*

## **Best practice**

### Capture logs using:

- AWS CloudTrail
- Amazon CloudWatch logs
- Elastic Load Balancing (ELB) logs
- Amazon Virtual Private Cloud (VPC) Flow Logs
- Amazon S3 bucket logs
- Other AWS service-specific log sources
- Operating system or third-party application logs
- Solutions from the AWS Marketplace

# Resources

- AWS Best Practices:  
[https://d0.awsstatic.com/whitepapers/AWS\\_Cloud\\_Best\\_Practices.pdf](https://d0.awsstatic.com/whitepapers/AWS_Cloud_Best_Practices.pdf)
- AWS Well Architected Framework:  
[https://d0.awsstatic.com/whitepapers/architecture/AWS\\_Well-Architected\\_Framework.pdf](https://d0.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf)
- AWS SlideShare Channel:  
<https://www.slideshare.net/AmazonWebServices/>
- AWS Risk and Compliance Whitepaper  
[https://d0.awsstatic.com/whitepapers/compliance/AWS\\_Risk\\_and\\_Compliance\\_Whitepaper.pdf](https://d0.awsstatic.com/whitepapers/compliance/AWS_Risk_and_Compliance_Whitepaper.pdf)
- AWS Security Best Practices:  
<https://d0.awsstatic.com/whitepapers/aws-security-best-practices.pdf>