

AWS Cloud Security Foundations– AWS-CSF

Dr. Ibrahim Gomaa

Lecture 6 – AWS Cloud Security



Lecture 6: AWS Cloud Security

Topics

- Security in the AWS Cloud
- Security design principles
- Shared responsibility model
- Securing a new AWS account
- Securing accounts
- Securing data on AWS
- Working to ensure compliance

Activities

- AWS shared responsibility model activity

Lab

- Introduction to AWS IAM

Security in the AWS Cloud



Security is familiar



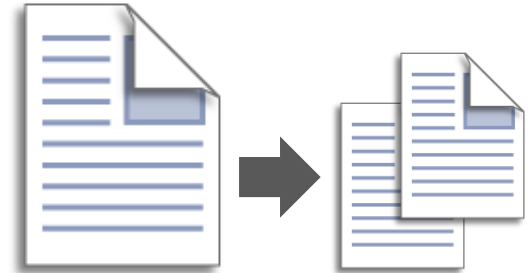
Confidentiality

- Limit access and disclosure to authorized users
- Prevent access by unauthorized people



Integrity

- Maintain data consistency during its lifecycle
- Preserve data at rest and data in transit



Availability

- Have access to information resources when needed

AWS Cloud security: Objectives

- Controllability
- Auditability
- Visibility
- Agility
- Automation

AWS Cloud security: Controllability



AWS Identity and Access Management (IAM)

- Can I effectively manage users?
- How can I provide temporary credentials?
- Can I use my own keys?
- The AWS IAM service helps you securely control access to AWS resources for your users. Use IAM to control who can use your AWS resources ([authentication](#)), what resources they can use, and in what ways ([authorization](#)).
- You can use the AWS Security Token Service ([AWS STS](#)) to create and provide trusted users with [temporary security credentials](#) that can control access to your AWS resources
- With [AWS CloudHSM](#), you can protect your encryption keys within hardware security modules (HSMs) that are designed and validated to government standards for secure key management. You can securely generate, store, and manage the cryptographic keys used for data encryption in a way that ensures that only you have access to the keys

AWS Cloud security: Visibility

- What is in my environment?
- What impact did a particular action have?
- What has changed?
- Where is the evidence?



AWS Config

by using [AWS Config](#), you can:

- Discover existing AWS resources
- Export a complete inventory of your AWS resources with all configuration details
- Determine how a resource was configured at any point in time.

AWS Cloud security: Visibility

- How do I ensure high availability?
- Can I automatically deploy applications with security and compliance-related settings?
- How can I apply security checks in a reproducible manner?



AWS CloudFormation

Security design principles



Apply the principle of least privilege



- Grant access as needed
- Enforce separation of duties
- Avoid long-term credentials



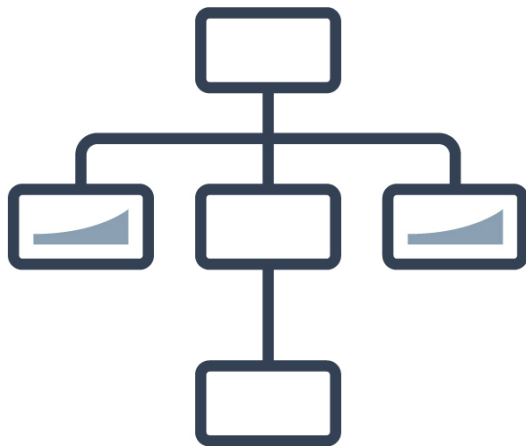
Enable traceability



- Monitor actions and changes
- Use logs and metrics
- Audit your cloud resources



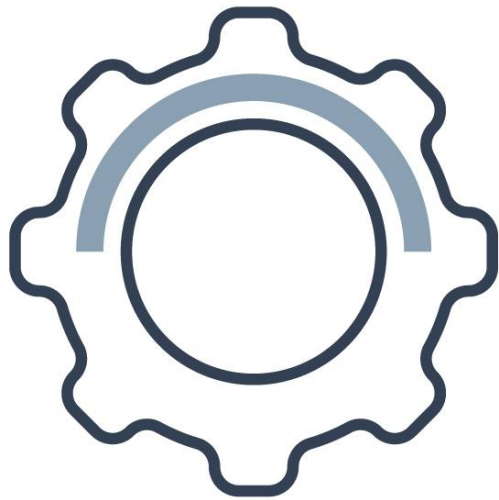
Secure all layers



- Use a defense in depth approach
- Use different AWS services



Automate security



- Automate routine security tasks with APIs
- Implement infrastructure as code



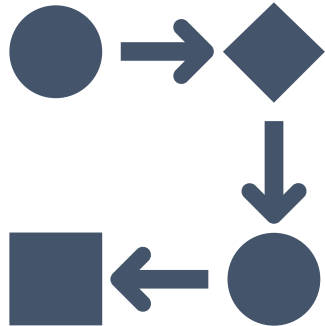
Protect data in transit and data at rest



- Use encryption and access controls
- Classify your data with tags
- Use VPN and TLS connections



Prepare for security events



- Mitigate the impact of security incidents
- Create processes to isolate incidents and restore operations



Minimize the attack surface



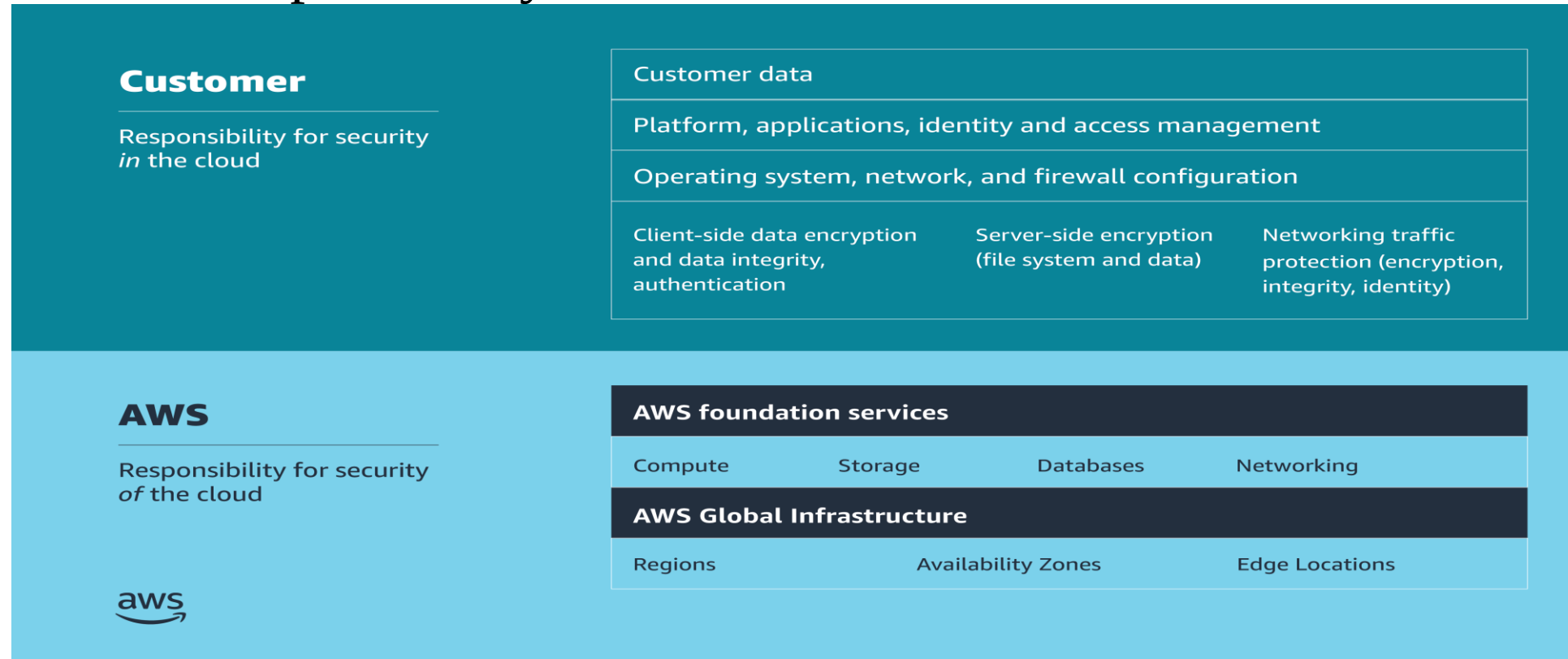
- Be ready to scale and absorb the attack
- Safeguard exposed resources



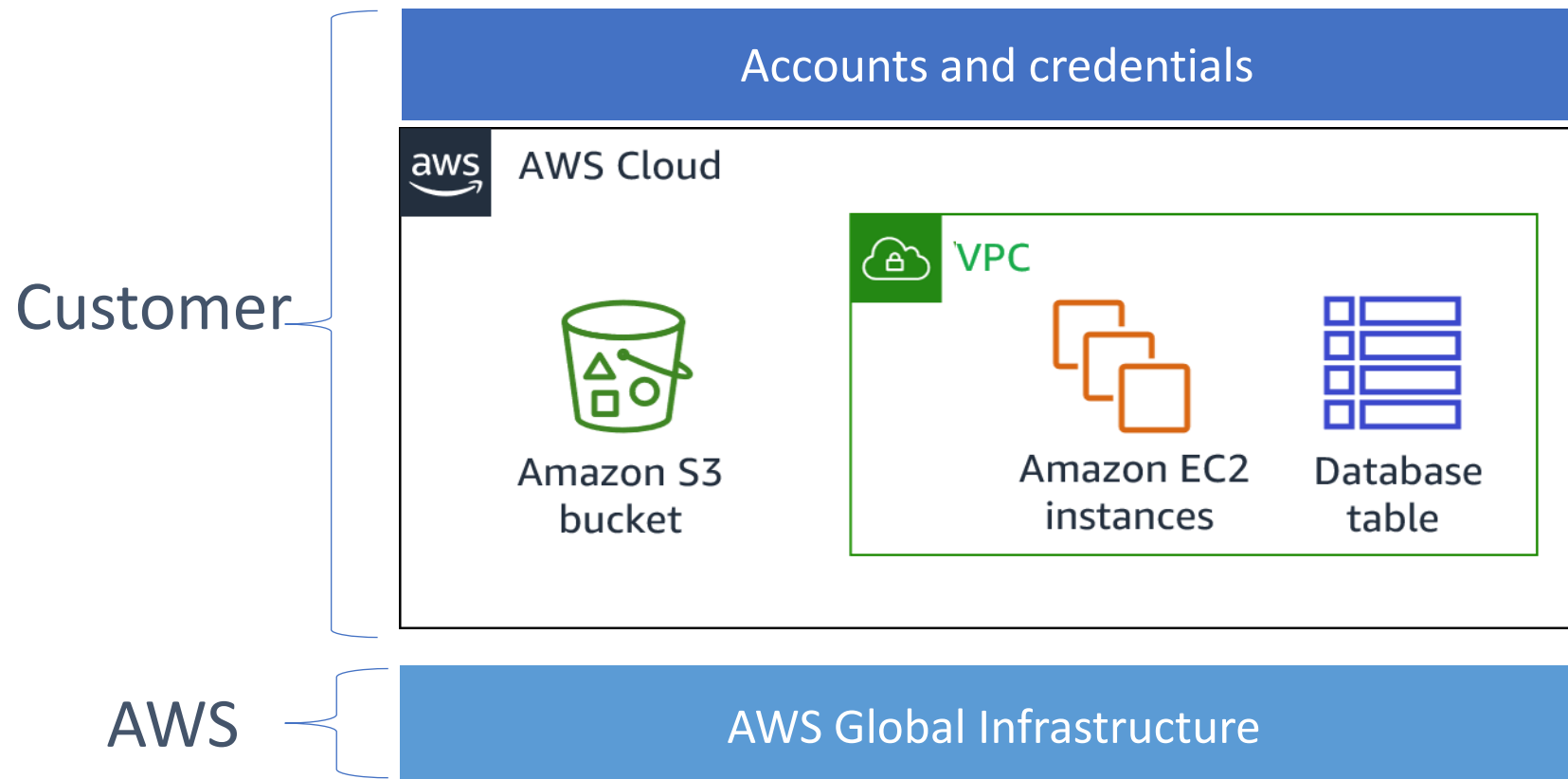
Shared responsibility model



AWS shared responsibility model



Shared responsibility example



Customer responsibility examples:

- Guest OS configuration
- Application-level security
- Security group configuration

Security *in* the cloud

Customer

Customer data

Platform, applications, identity and access management

Operating system, network and firewall configuration

Client-side data encryption and
data integrity, authentication

Server-side encryption
(file system and/or data)

Network traffic protection
(encryption/integrity/identity)

Considerations

- What you should store
- Which AWS services you should use
- Which Region to store data in
- What content format and structure to use
- Who has access

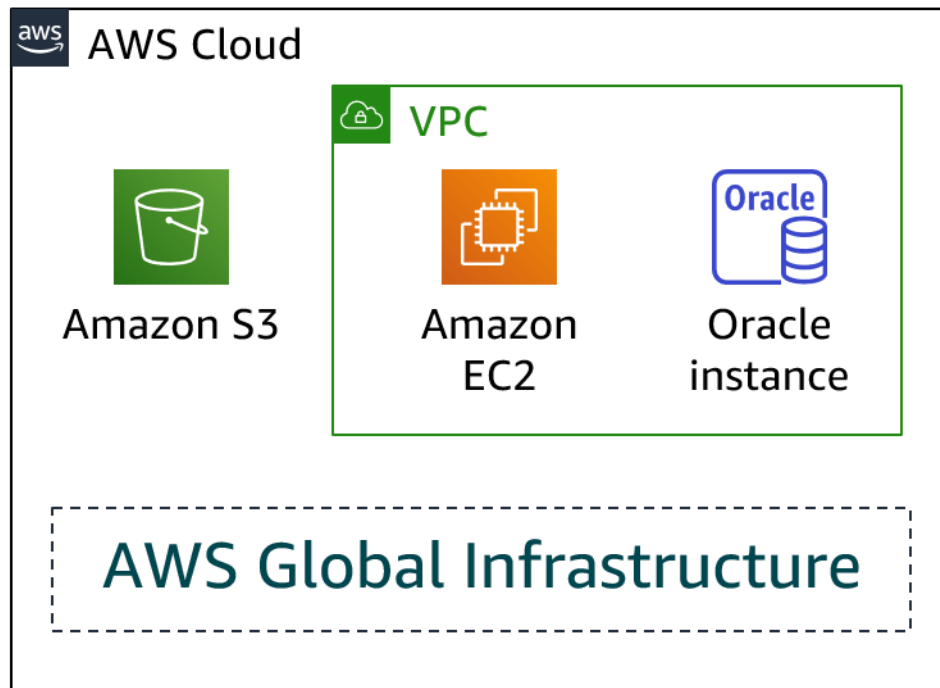


Activity: Shared Responsibility Model



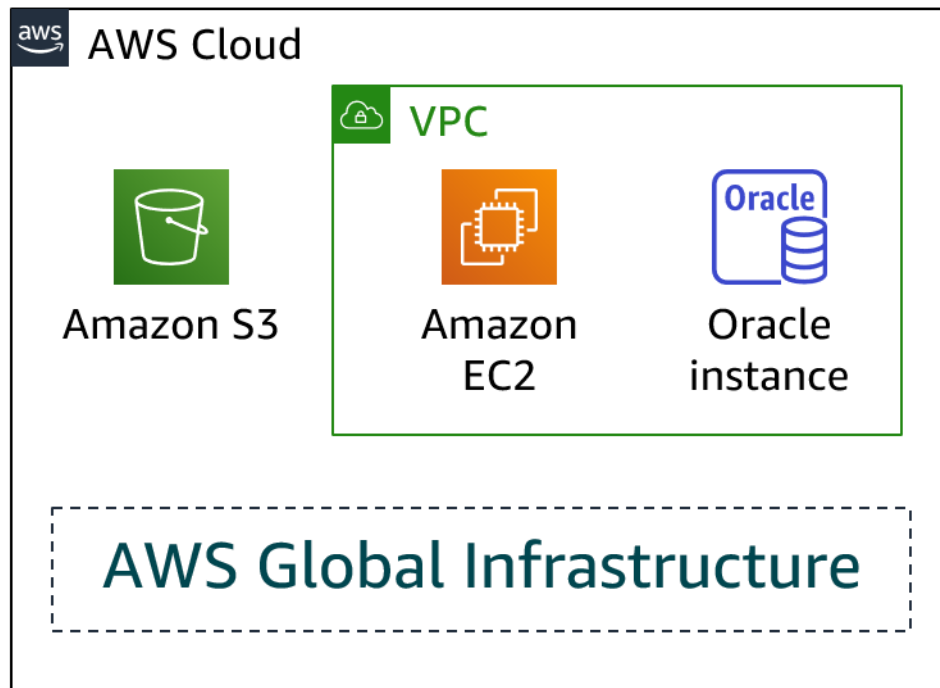
Activity: Scenario 1

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



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?

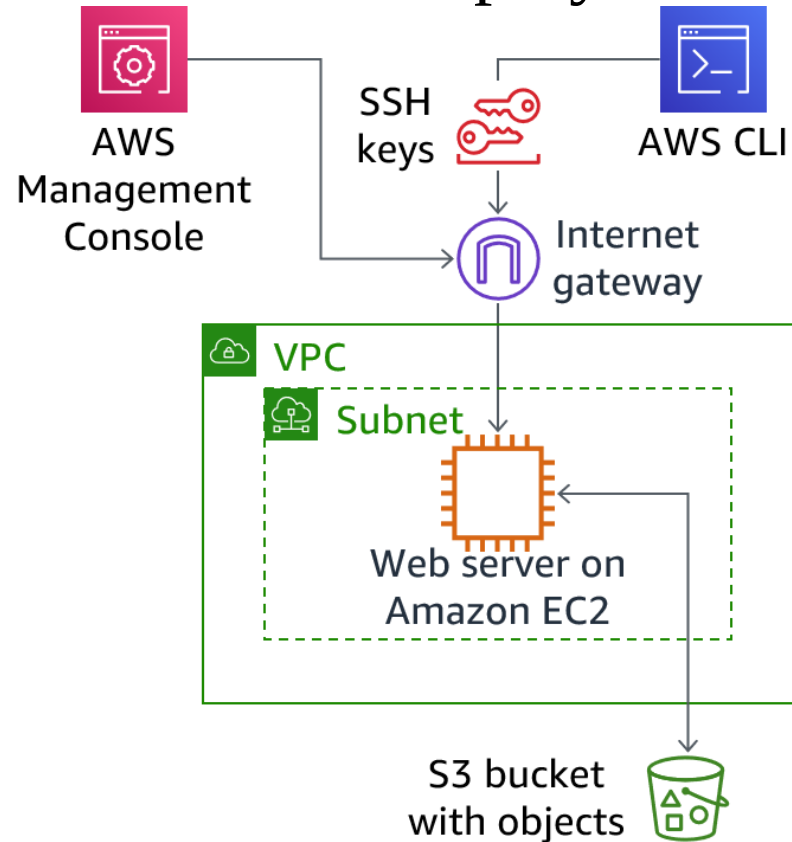
Answer: The customer

8. S3 bucket access configuration?

Answer: The customer

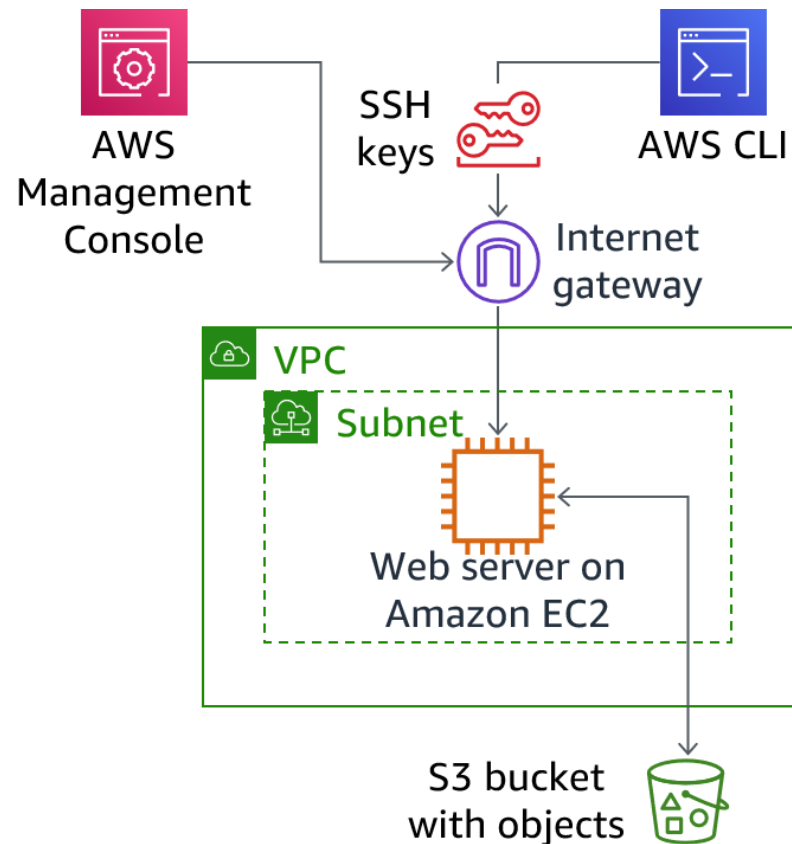
Activity: Scenario 2

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



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

Managed Services Organization

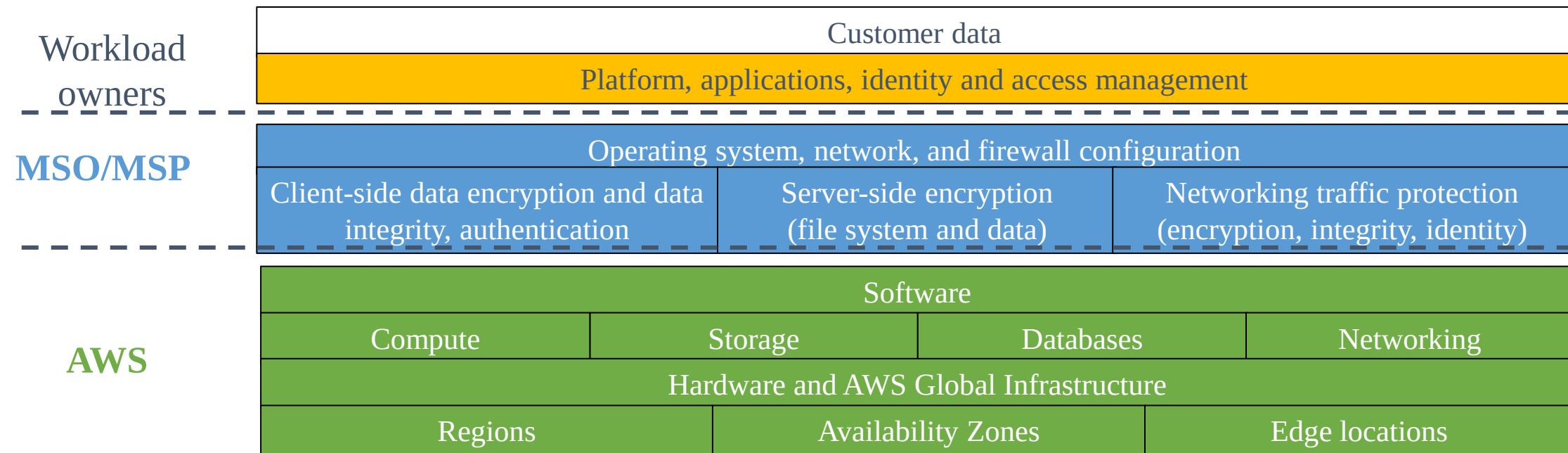
- One approach to implement security and governance is to create a centralized team that is responsible for establishing repeatable processes and templates to deploy applications to AWS.
- Such a team could be either internal or external (third-party vendors), and is often referred to as **a provisioning team, center of excellence, or managed services organization (MSO)**.
- External vendors are commonly referred to as **managed service providers (MSPs)**.



MSO responsibility model

workload owners handle the actual deployment, development, and maintenance of applications. (system administrators, developers, and others who are directly responsible for one or more applications)

MSO ensures that applications are deployed in a secure and compliant fashion through the automated implementation of organizational security requirements.

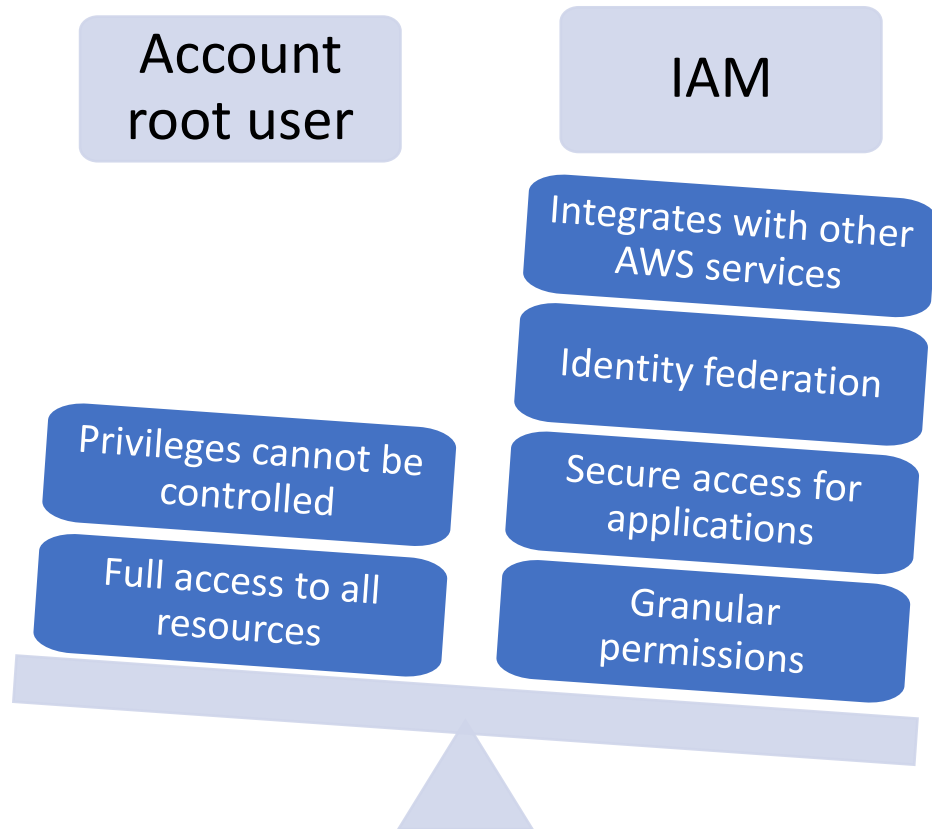


AWS Cloud Security Foundations– AWS-CSF

Securing a new AWS account



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.

Securing accounts



AWS Organizations

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

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



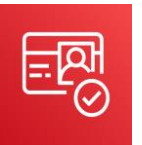
AWS Key Management
Service (AWS KMS)

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.

(SAML is an open standard for exchanging identity and security information with applications and service providers.)



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

Securing data on AWS



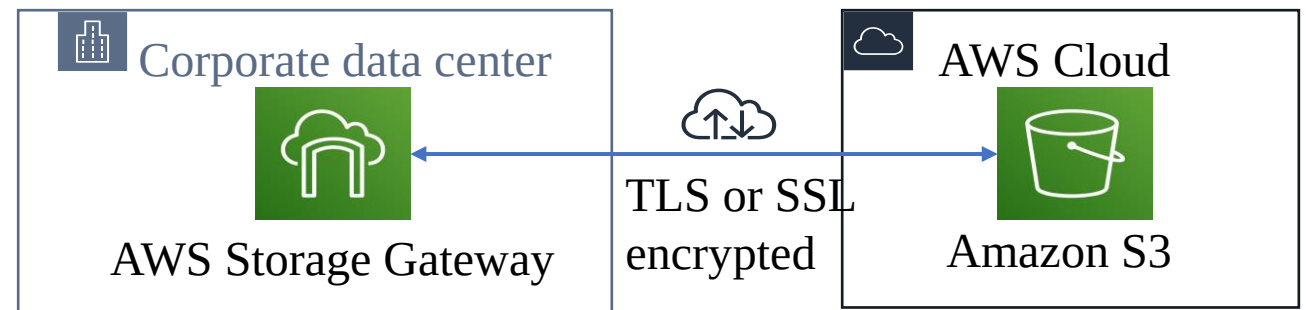
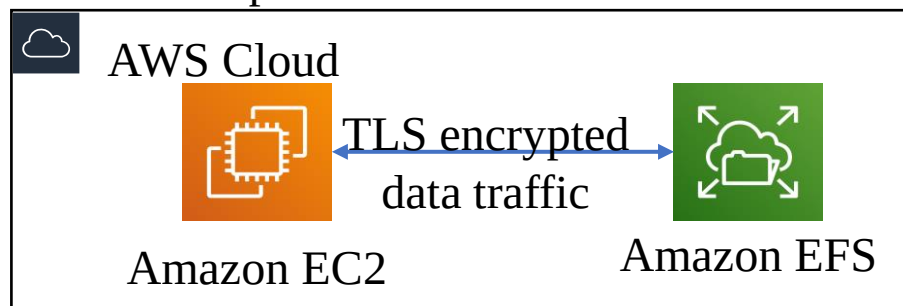
Encryption of data at rest

- **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:



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.

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

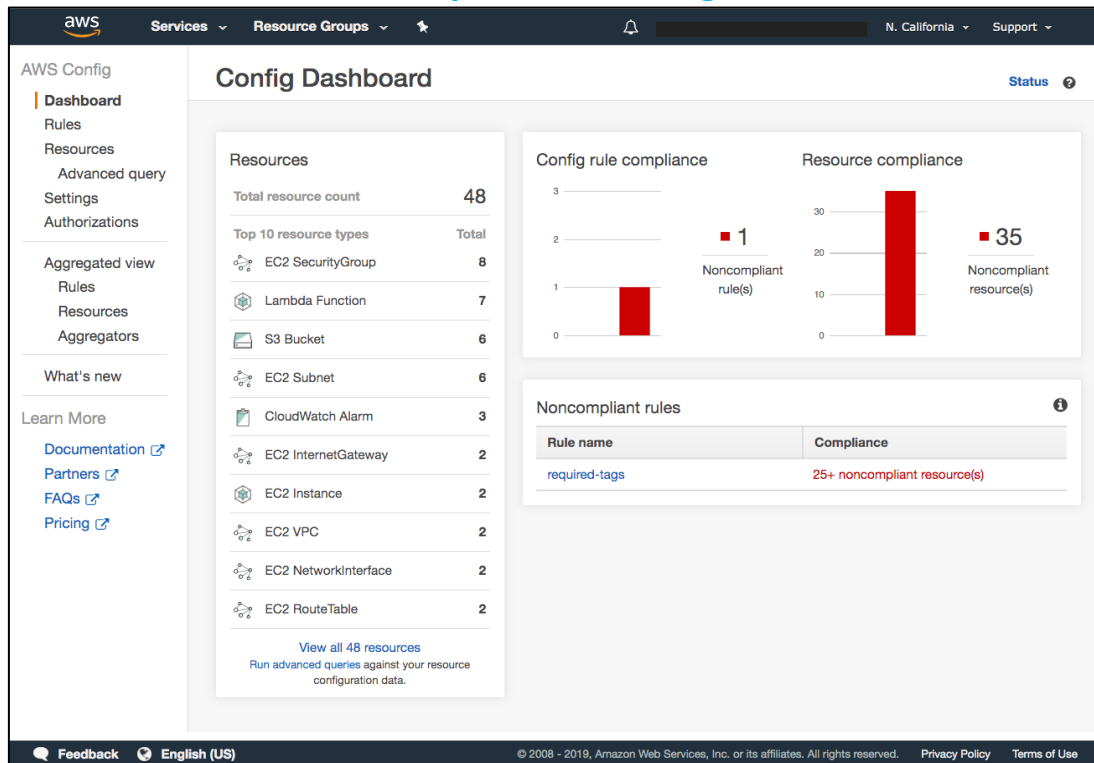




AWS Config

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**.

Thanks