Cloud Security

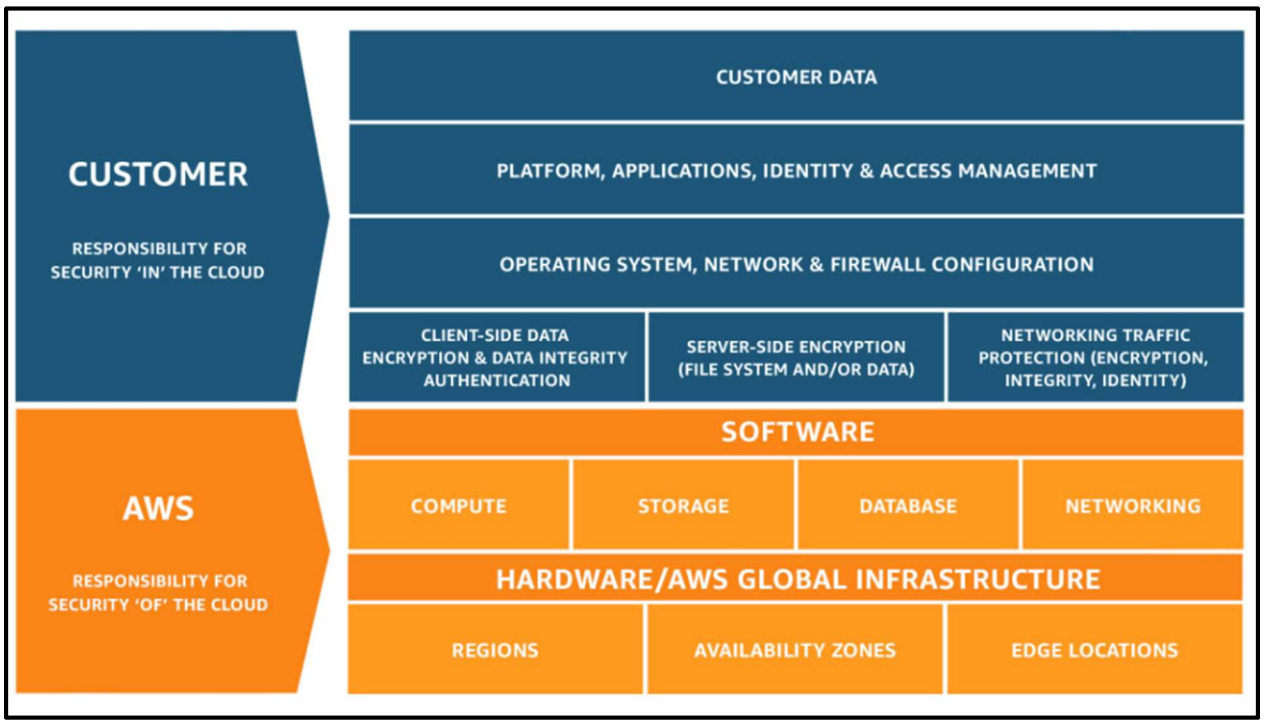
- Now a day, Security threats have become more advanced and keeping data and applications safe has become critical.
- **Cloud security refers to the technologies, policies, controls, and services that protect cloud data, applications, and infrastructure from internal and external threats.**

Defense in Depth

- Defense in Depth (DiD) is an approach to cybersecurity in which a series of defensive mechanisms are layered in order to protect valuable data and information.
- If one mechanism fails, another steps up immediately to thwart an attack.
- This multi-layered approach with intentional redundancies increases the security of a system as a whole and addresses many different attack vectors.

AWS Security and Encryption

- AWS operates under a shared security responsibility model, where AWS is responsible for the security of the underlying cloud infrastructure and you are responsible for securing workloads you deploy in AWS (*Figure 1*).



- AWS responsibility "Security of the Cloud" - AWS is responsible for protecting the infrastructure that runs all of the services offered in the AWS Cloud. This infrastructure is composed of the hardware, software, networking, and facilities that run AWS Cloud services.

- Customer responsibility "Security in the Cloud" – Customer responsibility will be determined by the AWS Cloud services that a customer selects. This determines the amount of configuration work the customer must perform as part of their security responsibilities. For example, a service such as Amazon Elastic Compute Cloud (Amazon EC2) is categorized as Infrastructure as a Service (IaaS) and, as such, requires the customer to perform all of the necessary security configuration and management tasks.
- AWS offers you the ability to add a layer of security to your data at rest in the cloud, providing scalable and efficient encryption features.

AWS KMS

- AWS Key Management Service (AWS KMS) is a managed service that makes it easy for you to create and control the cryptographic keys that are used to protect your data.
- AWS KMS uses hardware security modules (HSM) to protect and validate your AWS KMS keys.
- AWS KMS integrates with most other AWS services that encrypt your data.
- You can use the AWS KMS API to create and manage KMS keys and special features, such as custom key stores, and use KMS keys in cryptographic operations.
- You can create and manage your AWS KMS keys:
    - ✓ Create, edit, and view symmetric and asymmetric KMS keys, including HMAC keys.
    - ✓ Control access to your KMS keys by using key policies, IAM policies, and grants. AWS KMS supports attribute-based access control (ABAC). You can also refine policies by using condition keys.
    - ✓ Create, delete, list, and update aliases, friendly names for your KMS keys. You can also use aliases to control access to your KMS keys.
    - ✓ Tag your KMS keys for identification, automation, and cost tracking. You can also use tags to control access to your KMS keys.
    - ✓ Enable and disable KMS keys.
    - ✓ Enable and disable automatic rotation of the cryptographic material in a KMS key.
    - ✓ Delete KMS keys to complete the key lifecycle

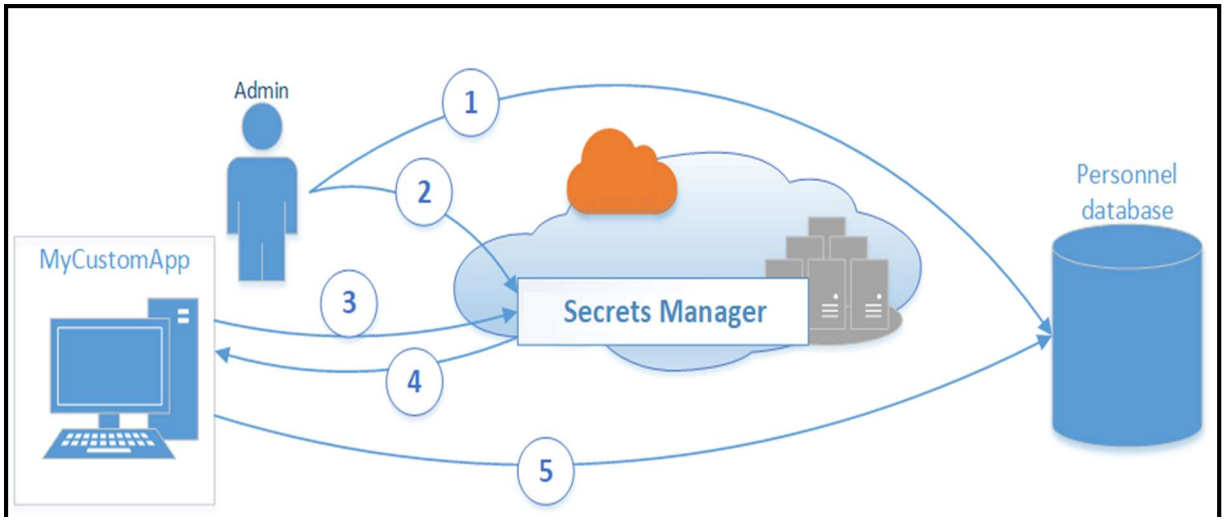
AWS SSM parameter store

- Parameter Store, a capability of AWS Systems Manager, provides secure, hierarchical storage for configuration data management and secrets management.
- You can store data such as passwords, database strings, Amazon Machine Image (AMI) IDs, and license codes as parameter values.
- You can store values as plain text or encrypted data.

- You can reference Systems Manager parameters in your scripts, commands, SSM documents, and configuration and automation workflows by using the unique name that you specified when you created the parameter.
- Parameter Store offers these benefits:
  - ✓ Use a secure, scalable, hosted secrets management service with no servers to manage.
  - ✓ Improve your security posture by separating your data from your code.
  - ✓ Store configuration data and encrypted strings in hierarchies and track versions.
  - ✓ Control and audit access at granular levels.
  - ✓ Store parameters reliably because Parameter Store is hosted in multiple Availability Zones in an AWS Region.

AWS Secret Manager

- Secrets Manager enables you to replace hardcoded credentials in your code, including passwords, with an API call to Secrets Manager to retrieve the secret programmatically.
- This helps ensure the secret can't be compromised by someone examining your code, because the secret no longer exists in the code.
- Also, you can configure Secrets Manager to automatically rotate the secret for you according to a specified schedule.
- **Basic AWS Secrets Manager scenario:**
  The following diagram illustrates the most basic scenario. The diagram displays you can store credentials for a database in Secrets Manager, and then use those credentials in an application to access the database.



1. The database administrator creates a set of credentials on the Personnel database for use by an application called MyCustomApp. The administrator also configures those

credentials with the permissions required for the application to access the Personnel database.

2. The database administrator stores the credentials as a secret in Secrets Manager named *MyCustomAppCreds*. Then, Secrets Manager encrypts and stores the credentials within the secret as the *protected secret text*.
3. When MyCustomApp accesses the database, the application queries Secrets Manager for the secret named *MyCustomAppCreds*.
4. Secrets Manager retrieves the secret, decrypts the protected secret text, and returns the secret to the client app over a secured (HTTPS with TLS) channel.
5. The client application parses the credentials, connection string, and any other required information from the response and then uses the information to access the database server.

Cloud HSM

- AWS CloudHSM provides hardware security modules in the AWS Cloud.
- A hardware security module (HSM) is a computing device that processes cryptographic operations and provides secure storage for cryptographic keys.
- When you use an HSM from AWS CloudHSM, you can perform a variety of cryptographic tasks:
    - ✓ Generate, store, import, export, and manage cryptographic keys, including symmetric keys and asymmetric key pairs.
    - ✓ Use symmetric and asymmetric algorithms to encrypt and decrypt data.
    - ✓ Use cryptographic hash functions to compute message digests and hash-based message authentication codes (HMACs).
    - ✓ Cryptographically sign data (including code signing) and verify signatures.
    - ✓ Generate cryptographically secure random data.

AWS Shield

- A DDoS attack is an attack in which multiple compromised systems try to flood a target with traffic.
- A DDoS attack can prevent legitimate end users from accessing the target services and can cause the target to crash due to overwhelming traffic volume.
- Protection against Distributed Denial of Service (DDoS) attacks is of primary importance for your internet-facing applications.
- When you build your application on AWS, you can make use of protections that AWS provides at no additional cost.

- Additionally, you can use the AWS Shield Advanced managed threat protection service to improve your security posture with additional DDoS detection, mitigation, and response capabilities.
- AWS Shield provides protection against a wide range of known DDoS attack vectors and zero-day attack vectors.
- Shield detection and mitigation is designed to provide coverage against threats even if they are not explicitly known to the service at the time of detection.

AWS Web Application Firewall (WAF)

- AWS WAF is a web application firewall that lets you monitor the HTTP(S) requests that are forwarded to your protected web application resources.
- You can protect the following resource types:
    - Amazon CloudFront distribution
    - Amazon API Gateway REST API
    - Application Load Balancer
    - AWS AppSync GraphQL API
    - Amazon Cognito user pool
- AWS WAF also lets you control access to your content.
- Based on criteria that you specify, such as the IP addresses that requests originate from or the values of query strings, the service associated with your protected resource responds to requests either with the requested content, with an HTTP 403 status code (Forbidden), or with a custom response.

AWS Guardduty

- Amazon GuardDuty is a threat detection service that continuously monitors for malicious activity and unauthorized behavior to protect your Amazon Web Services accounts, workloads, and data stored in Amazon S3.
- With the cloud, the collection and aggregation of account and network activities is simplified, but it can be time consuming for security teams to continuously analyze event log data for potential threats.
- With GuardDuty, you now have an intelligent and cost-effective option for continuous threat detection in Amazon Web Services Cloud.
- The service uses machine learning, anomaly detection, and integrated threat intelligence to identify and prioritize potential threats.
- GuardDuty analyzes tens of billions of events across multiple Amazon Web Services data sources, such as Amazon CloudTrail event logs, Amazon VPC Flow Logs, and DNS logs.
- With a few clicks in the Amazon Web Services Management Console, GuardDuty can be enabled with no software or hardware to deploy or maintain.

- By integrating with Amazon CloudWatch Events, GuardDuty alerts are actionable, easy to aggregate across multiple accounts, and straightforward to push into existing event management and workflow systems.
- Amazon GuardDuty is a security monitoring service that analyzes and processes data sources, such as AWS CloudTrail data events for Amazon S3 logs, CloudTrail management event logs, DNS logs, Amazon EBS volume data, Kubernetes audit logs, Amazon VPC flow logs, and RDS login activity.
- It uses threat intelligence feeds, such as lists of malicious IP addresses and domains, and machine learning to identify unexpected, potentially unauthorized, and malicious activity within your AWS environment.
- This can include issues like escalation of privileges, use of exposed credentials, or communication with malicious IP addresses, domains, presence of malware on your Amazon EC2 instances and container workloads, or discovery of unusual patterns of login events on your database.
- For example, GuardDuty can detect compromised EC2 instances and container workloads serving malware, or mining bitcoin.
- It also monitors AWS account access behavior for signs of compromise, such as unauthorized infrastructure deployments, like instances deployed in a Region that hasn't been used before, or unusual API calls like a password policy change to reduce password strength.
- GuardDuty informs you of the status of your AWS environment by producing security findings that you can view in the GuardDuty console or through Amazon CloudWatch Events.
- GuardDuty also provides support for you to export your findings to an Amazon Simple Storage Service (S3) bucket, and integrate with other services such as Amazon Detective and Amazon Inspector.

AWS Inspector

- Amazon Inspector is a vulnerability management service that continuously scans your AWS workloads for software vulnerabilities and unintended network exposure.
- Amazon Inspector automatically discovers and scans running Amazon EC2 instances, container images in Amazon Elastic Container Registry (Amazon ECR), and AWS Lambda functions for known software vulnerabilities and unintended network exposure.
- Amazon Inspector creates a *finding* when it discovers a software vulnerability or network configuration issue.
- A finding describes the vulnerability, identifies the affected resource, rates the severity of the vulnerability, and provides remediation guidance.
- You can analyze findings using the Amazon Inspector console, or view and process your findings through other AWS services.

AWS Macie

- Amazon Macie is a data security service that discovers sensitive data by using machine learning and pattern matching, provides visibility into data security risks, and enables automated protection against those risks.
- To help you manage the security posture of your organization's Amazon Simple Storage Service (Amazon S3) data estate, Macie provides you with an inventory of your S3 buckets, and automatically evaluates and monitors the buckets for security and access control.
- If Macie detects a potential issue with the security or privacy of your data, such as a bucket that becomes publicly accessible, Macie generates a finding for you to review and remediate as necessary.
- Macie also automates discovery and reporting of sensitive data to provide you with a better understanding of the data that your organization stores in Amazon S3.
- To detect sensitive data, you can use built-in criteria and techniques that Macie provides, custom criteria that you define, or a combination of the two.
- If Macie detects sensitive data in an S3 object, Macie generates a finding to notify you of the sensitive data that Macie found.

AWS Well Architected Framework with more focus on Security

- The AWS Well-Architected Framework helps you understand the pros and cons of decisions you make while building systems on AWS.
- By using the Framework you will learn architectural best practices for designing and operating reliable, secure, efficient, cost-effective, and sustainable systems in the cloud.
- The framework is based on six pillars:
    - Operational Excellence
    - Security
    - Reliability
    - Performance Efficiency
    - Cost Optimization
    - Sustainability
- The security pillar describes how to take advantage of cloud technologies to protect data, systems, and assets in a way that can improve your security posture.
- Design principles that can help you strengthen your cloud security:
    - **Implement a strong identity foundation:** Implement the principle of least privilege and enforce separation of duties with appropriate authorization for each interaction with your AWS resources.
    - **Enable traceability:** Monitor, alert, and audit actions and changes to your environment in real time.
    - **Apply security at all layers:** Apply a defense in depth approach with multiple security controls.
    - **Automate security best practices:** Automated software-based security mechanisms improve your ability to securely scale more rapidly and cost-effectively.

- **Protect data in transit and at rest**: Classify your data into sensitivity levels and use mechanisms, such as encryption, tokenization, and access control where appropriate.
- **Keep people away from data:** Use mechanisms and tools to reduce or eliminate the need for direct access or manual processing of data.
- **Prepare for security events:** Prepare for an incident by having incident management and investigation policy and processes that align to your organizational requirements.