AWS Security Management

[Edition 02]

[Last Update 210508]

Contents

1 DOCUMENTATION LINK

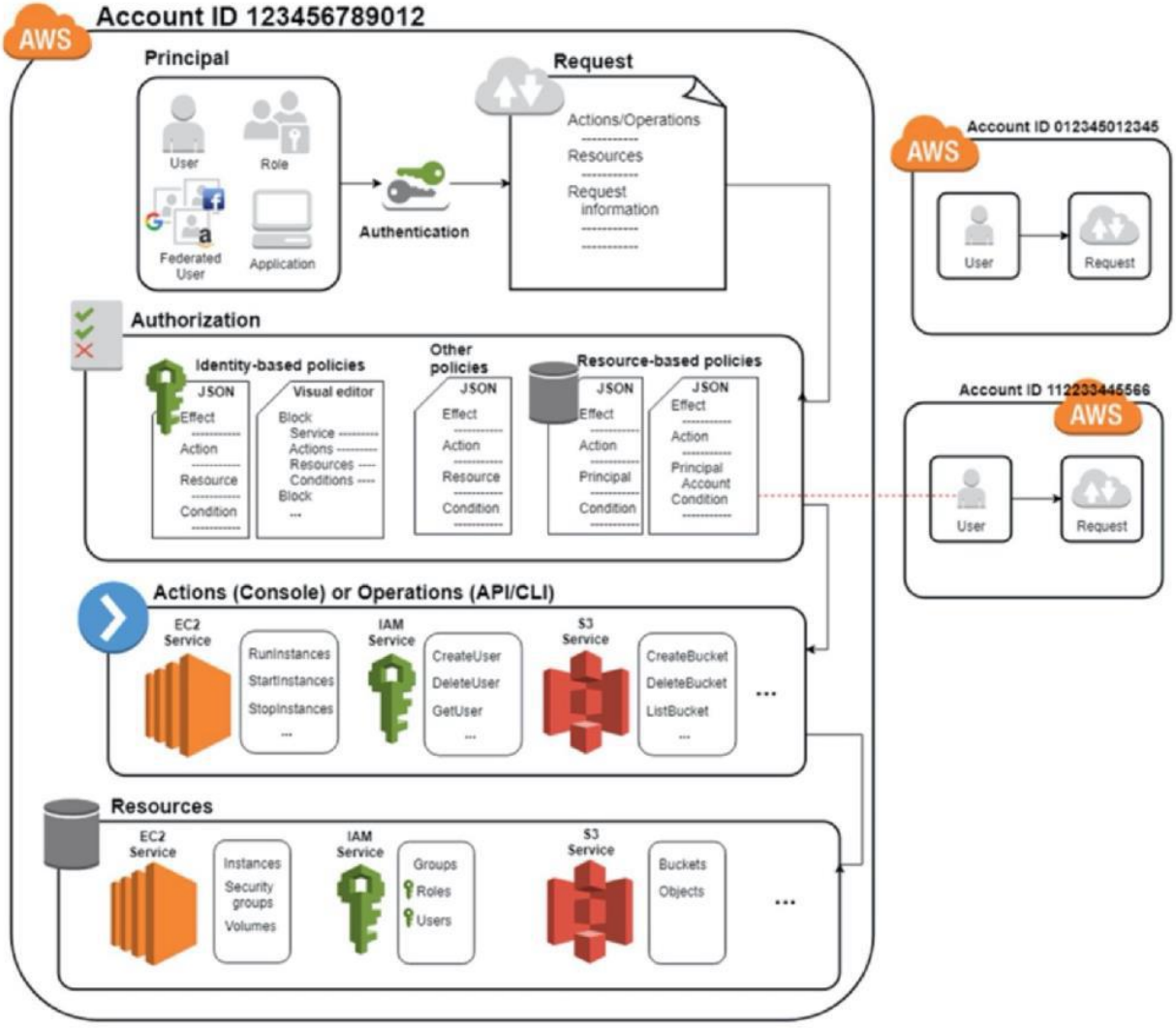
1. Setting Password Policy for IAM Users
   https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_account-policy.html
2. Create IAM groups
   https://docs.aws.amazon.com/IAM/latest/UserGuide/id_groups_create.html
3. Create an IAM user in your AWS account
   https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html
4. Managing IAM Policies
   https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_manage.html
5. IAM Roles in AWS
   https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html
6. Changing the AWS account root user password
   https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_change-root.html
7. AWS WAF - Web Application Firewall
   https://aws.amazon.com/waf/
8. AWS WAF Pricing
   https://aws.amazon.com/waf/pricing/
9. AWS Key Management Service (KMS)
   https://aws.amazon.com/kms/
10. AWS Key Management Service pricing
    https://aws.amazon.com/kms/pricing/

2 AWS IAM

- IAM is used to securely control individual and group access to AWS resources.
- IAM makes it easy to provide multiple users secure access to AWS resources.
- **IAM can be used to manage:**
  - Users
  - Groups
  - Access policies
  - Roles
  - User credentials
  - User password policies
  - Multi-factor authentication (MFA)
  - API keys for programmatic access (CLI)
- Provides centralized control of your AWS account.
- Enables shared access to your AWS account.
- By default, new users are created with NO access to any AWS services – they can only login to the AWS console.
- Permission must be explicitly granted to allow a user to access an AWS service.
- IAM users are individuals who have been granted access to an AWS account.
- **Each IAM user has three main components:**
  - A user-name
  - A password
  - Permissions to access various resources
- You can apply granular permissions with IAM.
- You can assign users individual security credentials such as access keys, passwords, and multi-factor authentication devices.
- IAM is not used for application-level authentication.
- Identity Federation (including AD, Facebook etc.) can be configured allowing secure access to resources in an AWS account without creating an IAM user account.
- Multi-factor authentication (MFA) can be enabled/enforced for the AWS account and for individual users under the account.

- MFA uses an authentication device that continually generates random, six-digit, single-use authentication codes.
- You can authenticate using an MFA device in the following three ways:
  - Through the AWS Management Console – the user is prompted for a user name, password and authentication code.
  - Using the AWS API – restrictions are added to IAM policies and developers can request temporary security credentials and pass MFA parameters in their AWS STS API requests.
  - Using the AWS CLI by obtaining temporary security credentials from STS (aws sts get-session-token).
- It is a best practice to use MFA for all users and to use U2F or hardware MFA devices for all privileged users.
- IAM is universal (global) and does not apply to regions.
- IAM is eventually consistent.
- IAM replicates data across multiple data centers around the world.
- The “root account” is the account created when you setup the AWS account. It has complete Admin access and is the only account that has this access by default.
- It is a best practice to not use the root account for anything other than billing.
- Power user access allows all permissions except the management of groups and users in IAM.
- Temporary security credentials consist of the AWS access key ID, secret access key, and security token.
- IAM can assign temporary security credentials to provide users with temporary access to services/resources.
- To sign-in you must provide your account ID or account alias in addition to a user name and password.
- The sign-in URL includes the account ID or account alias, e.g.: https://My_AWS_Account_ID.signin.aws.amazon.com/console/.
- Alternatively, you can sign-in at the following URL and enter your account ID or alias manually:
- https://console.aws.amazon.com/.
- IAM integrates with many different AWS services.
- IAM supports PCI DSS compliance.
- AWS recommend that you use the AWS SDKs to make programmatic API calls to IAM.
- However, you can also use the IAM Query API to make direct calls to the IAM web service.

2.1 IAM Infrastructure Elements



Principals:

- An entity that can take an action on an AWS resource.
- Your administrative IAM user is your first principal.
- You can allow users and services to assume a role.
- IAM supports federated users.
- IAM supports programmatic access to allow an application to access your AWS account.
- IAM users, roles, federated users, and applications are all AWS principals

Requests:

- Principals send requests via the Console, CLI, SDKs, or APIs.

Requests are:

- Actions (or operations) that the principal wants to perform.
- Resources upon which the actions are performed.
- Principal information including the environment from which the request was made.

Request context - AWS gathers the request information:

- Principal (requester).
- Aggregate permissions associated with the principal.
- Environment data, such as IP address, user agent, SSL status etc.
- Resource data, or data that is related to the resource being requested.

Authentication:

- A principal sending a request must be authenticated to send a request to AWS.
- To authenticate from the console, you must sign in with your user name and password.
- To authenticate from the API or CLI, you must provide your access key and secret key.

Authorization:

- IAM uses values from the request context to check for matching policies and determines whether to allow or deny the request.
- IAM policies are stored in IAM as JSON documents and specify the permissions that are allowed or denied.

IAM policies can be:

- User (identity) based policies
- Resource-based policies
- IAM checks each policy that matches the context of your request.
- If a single policy has a deny action IAM denies the request and stops evaluating (explicit deny).

Evaluation logic:

- By default, all requests are denied (implicit deny).
- An explicit allow overrides the implicit deny.
- An explicit deny overrides any explicit allows.
- Only the root user has access to all resources in the account by default.

Actions:

- Actions are defined by a service.
- Actions are the things you can do to a resource such as viewing, creating, editing, deleting.
- Any actions on resources that are not explicitly allowed are denied.
- To allow a principal to perform an action you must include the necessary actions in a policy that applies to the principal or the affected resource.

Resources:

- A resource is an entity that exists within a service.
  E.g. EC2 instances, S3 buckets, IAM users, and DynamoDB tables.
- Each AWS service defines a set of actions that can be performed on the resource.
- After AWS approves the actions in your request, those actions can be performed on the related resources within your account.

2.2 Authentication Methods

Console password:

- A password that the user can enter to sign into interactive sessions such as the AWS Management Console.
- You can allow users to change their own passwords.
- You can allow selected IAM users to change their passwords by disabling the option for all users and using an IAM policy to grant permissions for the selected users.

Access Keys:

- A combination of an access key ID and a secret access key.
- You can assign two active access keys to a user at a time.
- These can be used to make programmatic calls to AWS when using the API in program code or at a command prompt when using the AWS CLI or the AWS PowerShell tools.

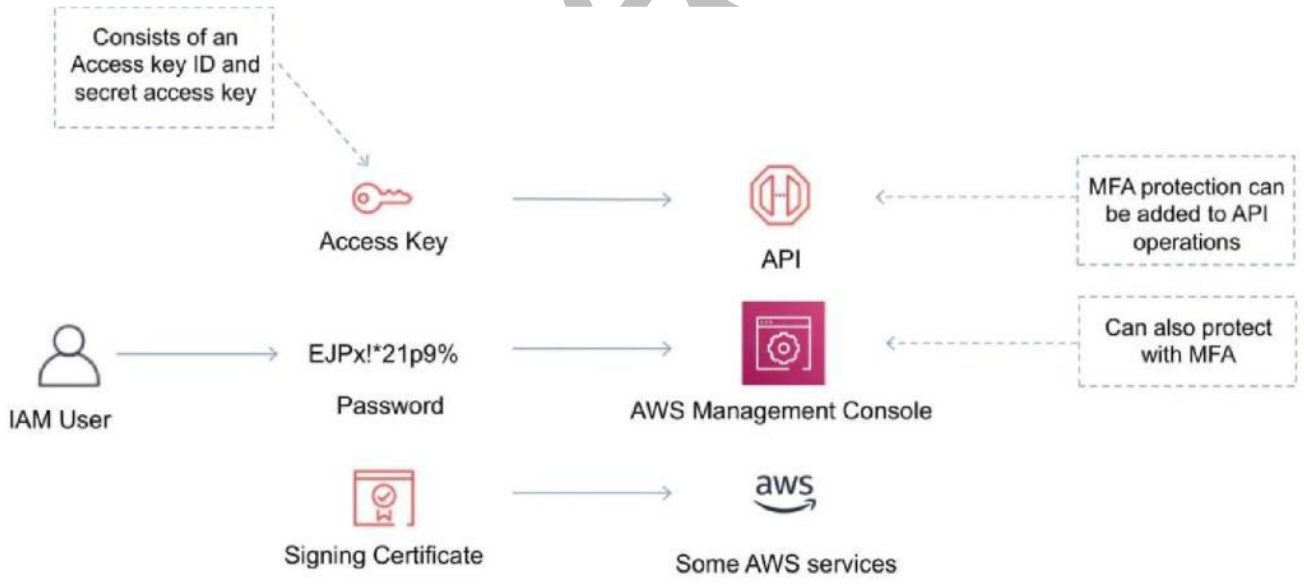
- You can create, modify, view or rotate access keys.
- When created IAM returns the access key ID and secret access key.
- The secret access is returned only at creation time and if lost a new key must be created.
- Ensure access keys and secret access keys are stored securely.
- Users can be given access to change their own keys through IAM policy (not from the console).

You can disable a user's access key which prevents it from being used for API calls.

Server certificates:

- SSL/TLS certificates that you can use to authenticate with some AWS services.
- AWS recommends that you use the AWS Certificate Manager (ACM) to provision, manage and deploy your server certificates.
- Use IAM only when you must support HTTPS connections in a region that is not supported by ACM.

The following diagram shows the different methods of authentication available with IAM:



2.3 IAM Users

- An IAM user is an entity that represents a person or service.

Can be assigned:

- An access key ID and secret access key for programmatic access to the AWS API, CLI, SDK, and other development tools.
- password for access to the management console.

- By default, users cannot access anything in your account.
- The account root user credentials are the email address used to create the account and a password.
- The root account has full administrative permissions, and these cannot be restricted.

Best practice for root accounts:

- Don't use the root user credentials.
- Don't share the root user credentials.
- Create an IAM user and assign administrative permissions as required.
- Enable MFA.

- IAM users can be created to represent applications and these are known as "service accounts".
- You can have up to 5000 users per AWS account.
- Each user account has a friendly name and an ARN which uniquely identifies the user across AWS.
- A unique ID is also created which is returned only when you create the user using the API, Tools for Windows PowerShell or the AWS CLI.
- You should create individual IAM accounts for users (best practice not to share accounts).
- The Access Key ID and Secret Access Key are not the same as a password and cannot be used to login to the AWS console.
- The Access Key ID and Secret Access Key can only be generated once and must be regenerated if lost.
- A password policy can be defined for enforcing password length, complexity etc. (applies to all users).

- You can allow or disallow the ability to change passwords using an IAM policy.
- Access keys and passwords should be changed regularly.

2.4 Groups

- Groups are collections of users and have policies attached to them.
- A group is not an identity and cannot be identified as a principal in an IAM policy.
- Use groups to assign permissions to users.
- Use the principal of least privilege when assigning permissions.
- You cannot nest groups (groups within groups).

2.5 Roles

- Roles are created and then “assumed” by trusted entities and define a set of permissions for making AWS service requests.
- With IAM Roles you can delegate permissions to resources for users and services without using permanent credentials (e.g. user name and password).
- IAM users or AWS services can assume a role to obtain temporary security credentials that can be used to make AWS API calls.
- You can delegate using roles.
- There are no credentials associated with a role (password or access keys).
- IAM users can temporarily assume a role to take on permissions for a specific task.
- A role can be assigned to a federated user who signs in using an external identity provider.
- Temporary credentials are primarily used with IAM roles and automatically expire.
- Roles can be assumed temporarily through the console or programmatically with the AWS CLI, Tools for Windows PowerShell or API.

IAM roles with EC2 instances:

- IAM roles can be used for granting applications running on EC2 instances permissions to AWS API requests using instance profiles.
- Only one role can be assigned to an EC2 instance at a time.
- A role can be assigned at the EC2 instance creation time or at any time afterwards.
- When using the AWS CLI or API instance profiles must be created manually (it’s automatic and transparent through the console).

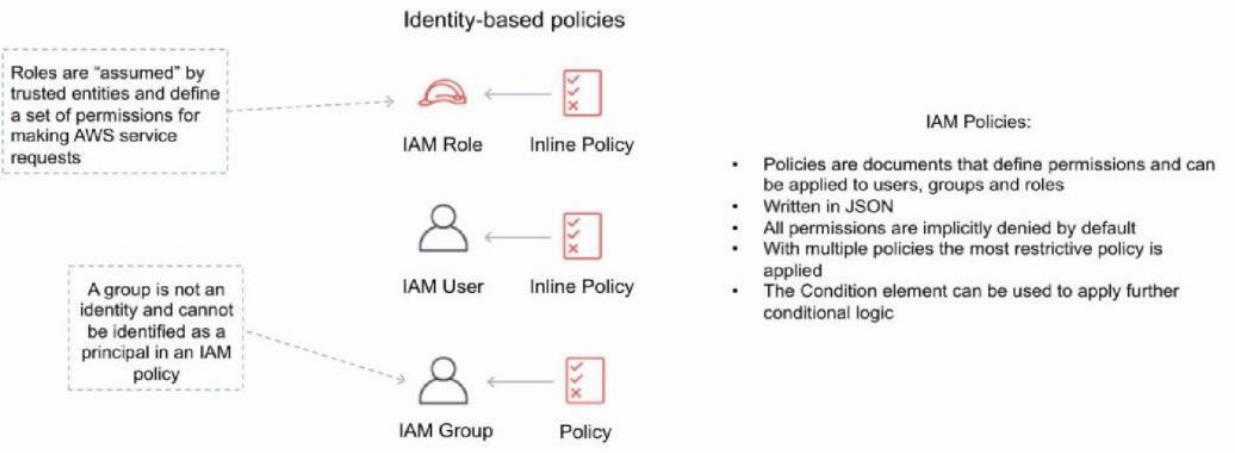
- Applications retrieve temporary security credentials from the instance metadata.

Role Delegation:

- Create an IAM role with two policies:
- Permissions policy – grants the user of the role the required permissions on a resource.
- Trust policy – specifies the trusted accounts that are allowed to assume the role.
- Wildcards (*) cannot be specified as a principal.
- A permissions policy must also be attached to the user in the trusted account.

2.6 Policies

- Policies are documents that define permissions and can be applied to users, groups and roles.
- Policy documents are written in JSON (key value pair that consists of an attribute and a value).
- All permissions are implicitly denied by default.
- The most restrictive policy is applied.
- The IAM policy simulator is a tool to help you understand, test, and validate the effects of access control policies.
- The Condition element can be used to apply further conditional logic.
- The diagram below provides some more information on the relationship between IAM roles, users, groups and policies.



2.7 IAM Best Practices

- Lock Away Your AWS Account Root User Access Keys.
- Create Individual IAM Users.
- Use Groups to Assign Permissions to IAM Users.
- Grant Least Privilege.
- Get Started Using Permissions with AWS Managed Policies.
- Use Customer Managed Policies Instead of Inline Policies.
- Use Access Levels to Review IAM Permissions.
- Configure a Strong Password Policy for Your Users.
- Enable MFA.
- Use Roles for Applications That Run on Amazon EC2 Instances.
- Use Roles to Delegate Permissions.
- Do Not Share Access Keys.
- Rotate Credentials Regularly.
- Remove Unnecessary Credentials.
- Use Policy Conditions for Extra Security.
- Monitor Activity in Your AWS Account.

2.8 Sample Questions

Q1: What does IAM stand for?

a. Improved Access Management

b. Identity and Access Management

c. Information and Access Management

d. Identity and Authorization Management

Answer: B

Explanation: IAM stands for Identity and Access Management.

Q2 : Which of the following does IAM manage? (Choose two.)

a. Management of users accessing the AWS platform

b. Management of permissions for hosted application features

c. Management of roles affecting resources within AWS

d. Management of cost controls for user actions

Answer:A, C

Explanation: IAM only applies to permissions for users, roles, and groups and does not affect billing or cost or specific application feature accessibility.

Q3 : Which of the following AWS services is associated with privilege management?

a. AWS Config

b. RDS

c. IAM

d. VPC

Answer: C

Explanation: IAM provides access management through users, roles, and permissions, all of which are related to privileges.

For more Questions Please check Certification Sample Quiz under each module
Link: https://k21academy.com/awssaquizm03

3 AWS KEY MANAGEMENT STORE (KMS)

- AWS Key Management Store (KMS) is a managed service that enables you to easily encrypt your data.
- AWS KMS provides a highly available key storage, management, and auditing solution for you to encrypt data within your own applications and control the encryption of stored data across AWS services.
- AWS KMS allows you to centrally manage and securely store your keys. These are known as customer master keys or CMKs.
- You can generate CMKs in KMS, in an AWS CloudHSM cluster, or import them from your own key management infrastructure.
- These master keys are protected by hardware security modules (HSMs) and are only ever used within those modules.
- You can submit data directly to KMS to be encrypted or decrypted using these master keys.
- You set usage policies on these keys that determine which users can use them to encrypt and decrypt data and under which conditions.
- KMS is tightly integrated into many AWS services like Lambda, S3, EBS, EFS, DynamoDB, SQS etc.
- AWS KMS is integrated with AWS services and client-side toolkits that use a method known as envelope encryption to encrypt your data.
- Under this method, KMS generates data keys which are used to encrypt data and are themselves encrypted using your master keys in KMS.
- Data keys are not retained or managed by KMS.
- AWS services encrypt your data and store an encrypted copy of the data key along with the data it protects.
- When a service needs to decrypt your data they request KMS to decrypt the data key using your master key.
- If the user requesting data from the AWS service is authorized to decrypt under your master key policy, the service will receive the decrypted data key from KMS with which it can decrypt the data and return it in plaintext.
- All requests to use your master keys are logged in AWS CloudTrail so you can understand who used which key under which context and when they used it.
- You can control who manages and accesses keys via IAM users and roles.
- You can audit the use of keys via CloudTrail.
- KMS differs from Secrets Manager as its purpose-built for encryption key management.

- KMS is validated by many compliance schemes (e.g. PCI DSS Level 1, FIPS 140-2 Level 2).
- You can perform the following key management functions in AWS KMS:
  - Create keys with a unique alias and description.
  - Import your own key material.
  - Define which IAM users and roles can manage keys.
  - Define which IAM users and roles can use keys to encrypt and decrypt data.
  - Choose to have AWS KMS automatically rotate your keys on an annual basis.
  - Temporarily disable keys so they cannot be used by anyone.
  - Re-enable disabled keys.
  - Delete keys that you no longer use.
  - Audit use of keys by inspecting logs in AWS CloudTrail.
  - Create custom key stores*.
  - Connect and disconnect custom key stores*.
  - Delete custom key stores*.
  - The use of custom key stores requires CloudHSM resources to be available in your account.
- **Typically, data is encrypted in one of the following three scenarios:**

1. You can use KMS APIs directly to encrypt and decrypt data using your master keys stored in KMS.

2. You can choose to have AWS services encrypt your data using your master keys stored in KMS. In this case data is encrypted using data keys that are protected by your master keys in KMS.

3. You can use the AWS Encryption SDK that is integrated with AWS KMS to perform encryption within your own applications, whether they operate in AWS or not.

- **Custom Key Store:**
  - The AWS KMS custom key store feature combines the controls provided by AWS CloudHSM with the integration and ease of use of AWS KMS.
  - You can configure your own CloudHSM cluster and authorize KMS to use it as a dedicated key store for your keys rather than the default KMS key store.
  - When you create keys in KMS you can chose to generate the key material in your CloudHSM cluster. Master keys that are generated in your custom key store never leave the HSMs in the CloudHSM cluster in plaintext and all KMS operations that use those keys are only performed in your HSMs.

- In all other respects master keys stored in your custom key store are consistent with other KMS CMKs.

Key deletion:

- You can schedule a customer master key and associated metadata that you created in AWS KMS for deletion, with a configurable waiting period from 7 to 30 days.
- This waiting period allows you to verify the impact of deleting a key on your applications and users that depend on it.
- The default waiting period is 30 days.
- You can cancel key deletion during the waiting period.

Limits:

- You can create up to 1000 customer master keys per account per region.
- As both enabled and disabled customer master keys count towards the limit, AWS recommend deleting disabled keys that you no longer use.
- AWS managed master keys created on your behalf for use within supported AWS services do not count against this limit.
- There is no limit to the number of data keys that can be derived using a master key and used in your application or by AWS services to encrypt data on your behalf.

3.1 Sample Questions

Q1 : A Solutions Architect is developing an encryption solution. The solution requires that data keys are encrypted using envelope protection before they are written to disk.

Which solution option can assist with this requirement?

a) API Gateway with STS
b) IAM Access Key
c) AWS Certificate Manager
d) AWS KMS API

Answer:D

Explanation: When you encrypt your data, your data is protected, but you have to protect your encryption key. One strategy is to encrypt it. Envelope encryption is the practice of encrypting plaintext data with a data key, and then encrypting the data key under another key.

Q2 : An application analyzes images of people that are uploaded to an Amazon S3 bucket. The application determines demographic data which is then saved to a .CSV file in another S3 bucket. The data must be encrypted at rest and then queried using SQL. The solution should be fully serverless.

Which actions should a Solutions Architect take to encrypt and query the data?

a. Use Amazon S3 server-side encryption and Amazon QuickSight to query the data
b. Use Amazon S3 server-side encryption and use Amazon RedShift Spectrum to query the data
c. Use AWS KMS encryption keys for the S3 bucket and use Amazon Kinesis Data Analytics to query the data
d. Use AWS KMS encryption keys for the S3 bucket and use Amazon Athena to query the data

Answer: D

Explanation: Amazon Athena is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL. Athena is serverless, so there is no infrastructure to manage, and you pay only for the queries that you run. Amazon Athena supports encrypted data for both the source data and query results, for example, using Amazon S3 with AWS KMS.

For more Questions Please check Certification Sample Quiz under each module
Link: https://k21academy.com/awssaquizm03

4 AWS WAF & SHIELD

- AWS WAF and AWS Shield help protect your AWS resources from web exploits and DDoS attacks.
- AWS WAF is a web application firewall service that helps protect your web apps from common exploits that could affect app availability, compromise security, or consume excessive resources.
- AWS Shield provides expanded DDoS attack protection for your AWS resources. Get 24/7 support from the DDoS response team and detailed visibility into DDoS events.
- We'll now go into more detail on each service.
- AWS WEB APPLICATION FIREWALL (WAF)
- AWS WAF is a web application firewall that helps protect your web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources.
- AWS WAF helps protect web applications from attacks by allowing you to configure rules that allow, block, or monitor (count) web requests based on conditions that you define.
- These conditions include IP addresses, HTTP headers, HTTP body, URI strings, SQL injection and cross-site scripting.
- AWS WAF gives you control over which traffic to allow or block to your web applications by defining customizable web security rules.
- New rules can be deployed within minutes, letting you respond quickly to changing traffic patterns.
- When AWS services receive requests for web sites, the requests are forwarded to AWS WAF for inspection against defined rules.
- Once a request meets a condition defined in the rules, AWS WAF instructs the underlying service to either block or allow the request based on the action you define.
- With AWS WAF you pay only for what you use.
- AWS WAF pricing is based on how many rules you deploy and how many web requests your web application receives.
- There are no upfront commitments.
- AWS WAF is tightly integrated with Amazon CloudFront and the Application Load Balancer (ALB), services.
- When you use AWS WAF on Amazon CloudFront, rules run in all AWS Edge Locations, located around the world close to end users.
- This means security doesn't come at the expense of performance.

- Blocked requests are stopped before they reach your web servers.
- When you use AWS WAF on an Application Load Balancer, your rules run in region and can be used to protect internet-facing as well as internal load balancers.

4.1 WEB Traffic Filtering

- AWS WAF lets you create rules to filter web traffic based on conditions that include IP addresses, HTTP headers and body, or custom URIs.
- This gives you an additional layer of protection from web attacks that attempt to exploit vulnerabilities in custom or third-party web applications.
- In addition, AWS WAF makes it easy to create rules that block common web exploits like SQL injection and cross site scripting.
- AWS WAF allows you to create a centralized set of rules that you can deploy across multiple websites.
- This means that in an environment with many websites and web applications you can create a single set of rules that you can reuse across applications rather than recreating that rule on every application you want to protect.

4.2 Full Feature API

- AWS WAF can be completely administered via APIs.
- This provides organizations with the ability to create and maintain rules automatically and incorporate them into the development and design process.
- For example, a developer who has detailed knowledge of the web application could create a security rule as part of the deployment process.
- This capability to incorporate security into your development process avoids the need for complex handoffs between application and security teams to make sure rules are kept up to date.
- AWS WAF can also be deployed and provisioned automatically with AWS CloudFormation sample templates that allow you to describe all security rules you would like to deploy for your web applications delivered by Amazon CloudFront.
- AWS WAF is integrated with Amazon CloudFront, which supports custom origins outside of AWS – this means you can protect web sites not hosted in AWS.
- Support for IPv6 allows the AWS WAF to inspect HTTP/S requests coming from both IPv6 and IPv4 addresses.

4.3 Real-Time Visibility

- AWS WAF provides real-time metrics and captures raw requests that include details about IP addresses, geo locations, URIs, User-Agent and Referers.
- AWS WAF is fully integrated with Amazon CloudWatch, making it easy to setup custom alarms when thresholds are exceeded, or particular attacks occur.
- This information provides valuable intelligence that can be used to create new rules to better protect applications.

4.4 AWS Shield

- AWS Shield is a managed Distributed Denial of Service (DDoS) protection service that safeguards applications running on AWS.
- AWS Shield provides always-on detection and automatic inline mitigations that minimize application downtime and latency, so there is no need to engage AWS Support to benefit from DDoS protection.

There are two tiers of AWS Shield – Standard and Advanced.

1. AWS SHIELD STANDARD

- All AWS customers benefit from the automatic protections of AWS Shield Standard, at no additional charge.
- AWS Shield Standard defends against most common, frequently occurring network and transport layer DDoS attacks that target web sites or applications.
- When using AWS Shield Standard with Amazon CloudFront and Amazon Route 53, you receive comprehensive availability protection against all known infrastructure (Layer 3 and 4) attacks.

2. AWS SHIELD ADVANCED

- Provides higher levels of protection against attacks targeting applications running on Amazon Elastic Compute Cloud (EC2), Elastic Load Balancing (ELB), Amazon CloudFront, AWS Global Accelerator and Amazon Route 53 resources.
- In addition to the network and transport layer protections that come with Standard, AWS Shield Advanced provides additional detection and mitigation against large and sophisticated DDoS attacks, near real-time visibility into attacks, and integration with AWS WAF, a web application firewall.
- AWS Shield Advanced also gives you 24×7 access to the AWS DDoS Response Team (DRT) and protection against DDoS related spikes in your Amazon Elastic Compute Cloud (EC2), Elastic Load Balancing (ELB), Amazon CloudFront, AWS Global Accelerator and Amazon Route 53 charges.

- AWS Shield Advanced is available globally on all Amazon CloudFront, AWS Global Accelerator, and Amazon Route 53 edge locations.
- Origin servers can be Amazon S3, Amazon Elastic Compute Cloud (EC2), Elastic Load Balancing (ELB), or a custom server outside of AWS.
- AWS Shield Advanced includes DDoS cost protection, a safeguard from scaling charges as a result of a DDoS attack that causes usage spikes on protected Amazon EC2, Elastic Load Balancing (ELB), Amazon CloudFront, AWS Global Accelerator, or Amazon Route 53.
- If any of the AWS Shield Advanced protected resources scale up in response to a DDoS attack, you can request credits via the regular AWS Support channel.

4.5 Sample Questions

Q1 : A website runs on Amazon EC2 instances in an Auto Scaling group behind an Application Load Balancer (ALB) which serves as an origin for an Amazon CloudFront distribution. An AWS WAF is being used to protect against SQL injection attacks. A review of security logs revealed an external malicious IP that needs to be blocked from accessing the website.

What should a solutions architect do to protect the application?

a. Modify the security groups for the EC2 instances in the target groups behind the ALB to deny the malicious IP address.

b. Modify the network ACL for the EC2 instances in the target groups behind the ALB to deny the malicious IP address

c. Modify the configuration of AWS WAF to add an IP match condition to block the malicious IP address

d. Modify the network ACL on the CloudFront distribution to add a deny rule for the malicious IP address

Answer : C

Explanation:

A new version of the AWS Web Application Firewall was released in November 2019. With AWS WAF classic you create “IP match conditions”, whereas with AWS WAF (new version) you create “IP set match statements”. Look out for wording on the exam.

The IP match condition / IP set match statement inspects the IP address of a web request’s origin against a set of IP addresses and address ranges. Use this to allow or block web requests based on the IP addresses that the requests originate from.

AWS WAF supports all IPv4 and IPv6 address ranges. An IP set can hold up to 10,000 IP addresses or IP address ranges to check.

Q2 : You need a service that can provide you with control over which traffic to allow or block to your web applications by defining customizable web security rules. You need to block common attack patterns, such as SQL injection and cross-site scripting, as well as creating custom rules for your own applications.

Which AWS service fits these requirements?

a) Route 53
b) CloudFront
c) Security Groups
d) AWS WAF

Answer:D

Explanation:

AWS WAF is a web application firewall that helps detect and block malicious web requests targeted at your web applications. AWS WAF allows you to create rules that can help protect against common web exploits like SQL injection and cross-site scripting. With AWS WAF you first identify the resource (either an Amazon CloudFront distribution or an Application Load Balancer) that you need to protect. You then deploy the rules and filters that will best protect your applications.

The other services listed do not enable you to create custom web security rules that can block known malicious attacks.

For more Questions Please check Certification Sample Quiz under each module
Link: https://k21academy.com/awssaquizm03