Free Questions for SAA-C03 by go4braindumps

Shared by Valentine on 29-01-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

A company wants to run its payment application on AWS The application receives payment notifications from mobile devices Payment notifications require a basic validation before they are sent for further processing

The backend processing application is long running and requires compute and memory to be adjusted The company does not want to manage the infrastructure

Which solution will meet these requirements with the LEAST operational overhead?

Options:

A- Create an Amazon Simple Queue Service (Amazon SQS) queue Integrate the queue with an Amazon EventBndge rule to receive payment notifications from mobile devices Configure the rule to validate payment notifications and send the notifications to the backend application Deploy the backend application on Amazon Elastic Kubernetes Service (Amazon EKS) Anywhere Create a standalone cluster

B- Create an Amazon API Gateway API Integrate the API with anAWS Step Functions state machine to receive payment notifications from mobile devices Invoke the state
machine to validate payment notifications and send the notifications to the backend application Deploy the backend application on Amazon Elastic Kubernetes Sen/ice (Amazon EKS). Configure an EKS cluster with self-managed nodes.

C- Create an Amazon Simple Queue Sen/ice (Amazon SQS) queue Integrate the queue with an Amazon EventBridge rule to receive payment notifications from mobile devices Configure the rule to validate payment notifications and send the notifications to the backend

application Deploy the backend application on Amazon EC2 Spot Instances Configure a Spot Fleet with a default allocation strategy.

D- Create an Amazon API Gateway API Integrate the API with AWS Lambda to receive payment notifications from mobile devices Invoke a Lambda function to validate payment notifications and send the notifications to the backend application Deploy the backend application on Amazon Elastic Container Service (Amazon ECS). Configure Amazon ECS with an AWS Fargate launch type.

Answer:

D

Explanation:

This option is the best solution because it allows the company to run its payment application on AWS with minimal operational overhead and infrastructure management. By using Amazon API Gateway, the company can create a secure and scalable API to receive payment notifications from mobile devices. By using AWS Lambda, the company can run a serverless function to validate the payment notifications and send them to the backend application. Lambda handles the provisioning, scaling, and security of the function, reducing the operational complexity and cost. By using Amazon ECS with AWS Fargate, the company can run the backend application on a fully managed container service that scales the compute resources automatically and does not require any EC2 instances to manage. Fargate allocates the right amount of CPU and memory for each container and adjusts them as needed.

A) Create an Amazon Simple Queue Service (Amazon SQS) queue Integrate the queue with an Amazon EventBndge rule to receive payment notifications from mobile devices Configure the rule to validate payment notifications and send the notifications to the backend application Deploy the backend application on Amazon Elastic Kubernetes Service (Amazon EKS) Anywhere Create a standalone cluster. This option is not optimal because it requires the company to manage the Kubernetes cluster that runs the backend application. Amazon EKS Anywhere is a deployment option that allows the company to create and operate Kubernetes clusters on-premises or in

other environments outside AWS. The company would need to provision, configure, scale, patch, and monitor the cluster nodes, which can increase the operational overhead and complexity. Moreover, the company would need to ensure the connectivity and security between the AWS services and the EKS Anywhere cluster, which can also add challenges and risks.

B) Create an Amazon API Gateway API Integrate the API with anAWS Step Functions state ma-chine to receive payment notifications from mobile devices Invoke the state machine to validate payment notifications and send the notifications to the backend application Deploy the backend application on Amazon Elastic Kubernetes Sen/ice (Amazon EKS). Configure an EKS cluster with self-managed nodes. This option is not ideal because it requires the company to manage the EC2 instances that host the Kubernetes cluster that runs the backend application. Amazon EKS is a fully managed service that runs Kubernetes on AWS, but it still requires the company to manage the worker nodes that run the containers. The company would need to provision, configure, scale, patch, and monitor the EC2 instances, which can increase the operational overhead and infrastructure costs. Moreover, using AWS Step Functions to validate the payment notifications may be unnecessary and complex, as the validation logic can be implemented in a simpler way with Lambda or other services.

C) Create an Amazon Simple Queue Sen/ice (Amazon SQS) queue Integrate the queue with an Amazon EventBridge rule to receive payment notifications from mobile devices Configure the rule to validate payment notifications and send the notifications to the backend application Deploy the backend application on Amazon EC2 Spot Instances Configure a Spot Fleet with a default al-location strategy. This option is not cost-effective because it requires the company to manage the EC2 instances that run the backend application. The company would need to provision, configure, scale, patch, and monitor the EC2 instances, which can increase the operational overhead and infrastructure costs. Moreover, using Spot Instances can introduce the risk of interruptions, as Spot Instances are reclaimed by AWS when the demand for On-Demand Instances increases. The company would need to handle the interruptions gracefully and ensure the availability and reliability of the backend application.

1Amazon API Gateway - Amazon Web Services

2AWS Lambda - Amazon Web Services

3Amazon Elastic Container Service - Amazon Web Services

4AWS Fargate - Amazon Web Services

Question 2

Question Type: MultipleChoice

A company is deploying an application that processes streaming data in near-real time The company plans to use Amazon EC2 instances for the workload The network architecture must be configurable to provide the lowest possible latency between nodes

Which combination of network solutions will meet these requirements? (Select TWO)

Options:

A- Enable and configure enhanced networking on each EC2 instance

B- Group the EC2 instances in separate accounts

C- Run the EC2 instances in a cluster placement group

D- Attach multiple elastic network interfaces to each EC2 instance

E- Use Amazon Elastic Block Store (Amazon EBS) optimized instance types.

Answer:

A, C

Explanation:

These options are the most suitable ways to configure the network architecture to provide the lowest possible latency between nodes. Option A enables and configures enhanced networking on each EC2 instance, which is a feature that improves the network performance of the instance by providing higher bandwidth, lower latency, and lower jitter. Enhanced networking uses single root I/O virtualization (SR-IOV) or Elastic Fabric Adapter (EFA) to provide direct access to the network hardware. You can enable and configure enhanced networking by choosing a supported instance type and a compatible operating system, and installing the required drivers. Option C runs the EC2 instances in a cluster placement group, which is a logical grouping of instances within a single Availability Zone that are placed close together on the same underlying hardware. Cluster placement groups provide the lowest network latency and the highest network throughput among the placement group options. You can run the EC2 instances in a cluster placement group by creating a placement group and launching the instances into it.

Option B is not suitable because grouping the EC2 instances in separate accounts does not provide the lowest possible latency between nodes. Separate accounts are used to isolate and organize resources for different purposes, such as security, billing, or compliance. However, they do not affect the network performance or proximity of the instances. Moreover, grouping the EC2 instances in separate accounts would incur additional costs and complexity, and it would require setting up cross-account networking and permissions.

Option D is not suitable because attaching multiple elastic network interfaces to each EC2 instance does not provide the lowest possible latency between nodes. Elastic network interfaces are virtual network interfaces that can be attached to EC2 instances to provide

additional network capabilities, such as multiple IP addresses, multiple subnets, or enhanced security. However, they do not affect the network performance or proximity of the instances. Moreover, attaching multiple elastic network interfaces to each EC2 instance would consume additional resources and limit the instance type choices.

Option E is not suitable because using Amazon EBS optimized instance types does not provide the lowest possible latency between nodes. Amazon EBS optimized instance types are instances that provide dedicated bandwidth for Amazon EBS volumes, which are block storage volumes that can be attached to EC2 instances. EBS optimized instance types improve the performance and consistency of the EBS volumes, but they do not affect the network performance or proximity of the instances. Moreover, using EBS optimized instance types would incur additional costs and may not be necessary for the streaming data workload.Reference:

Enhanced networking on Linux

Placement groups

Elastic network interfaces

Amazon EBS-optimized instances

Question 3

Question Type: MultipleChoice

A company's website hosted on Amazon EC2 instances processes classified data stored in Amazon S3 Due to security concerns, the company requires a pnvate and secure connection between its EC2 resources and Amazon S3.

Which solution meets these requirements?

Options:

A- Set up S3 bucket policies to allow access from a VPC endpomt.

B- Set up an 1AM policy to grant read-write access to the S3 bucket.

C- Set up a NAT gateway to access resources outside the private subnet.

D- Set up an access key ID and a secret access key to access the S3 bucket.

Answer:

A

Explanation:

This solution meets the following requirements:

It is private and secure, as it allows the EC2 instances to access the S3 bucket without using the public internet. A VPC endpoint is a gateway that enables you to create a private connection between your VPC and another AWS service, such as S3, within the same Region. A VPC endpoint for S3 provides secure and direct access to S3 buckets and objects using private IP addresses from your VPC. You can also use VPC endpoint policies and S3 bucket policies to control the access to the S3 resources based on the endpoint, the IAM user, the IAM role, or the source IP address.

It is simple and scalable, as it does not require any additional AWS services, gateways, or NAT devices. A VPC endpoint for S3 is a fully managed service that scales automatically with the network traffic. You can create a VPC endpoint for S3 with a few clicks in the VPC console or with a simple API call. You can also use the same VPC endpoint to access multiple S3 buckets in the same Region.

VPC Endpoints - Amazon Virtual Private Cloud

Gateway VPC endpoints - Amazon Virtual Private Cloud

Using Amazon S3 with interface VPC endpoints - Amazon Simple Storage Service

Using Amazon S3 with gateway VPC endpoints - Amazon Simple Storage Service

Question 4

Question Type: MultipleChoice

A company website hosted on Amazon EC2 instances processes classified data stored in The application writes data to Amazon Elastic Block Store (Amazon EBS) volumes The company needs to ensure that all data that is written to the EBS volumes is encrypted at rest.

Which solution will meet this requirement?

Options:

A- Create an 1AM role that specifies EBS encryption Attach the role to the EC2 instances

B- Create the EBS volumes as encrypted volumes Attach the EBS volumes to the EC2 instances

C- Create an EC2 instance tag that has a key of Encrypt and a value of True Tag all instances that require encryption at the EBS level

D- Create an AWS Key Management Service (AWS KMS) key policy that enforces EBS encryption in the account Ensure that the key policy is active

Answer:

B

Explanation:

The simplest and most effective way to ensure that all data that is written to the EBS volumes is encrypted at rest is to create the EBS volumes as encrypted volumes. You can do this by selecting the encryption option when you create a new EBS volume, or by copying an existing unencrypted volume to a new encrypted volume. You can also specify the AWS KMS key that you want to use for encryption, or use the default AWS-managed key. When you attach the encrypted EBS volumes to the EC2 instances, the data will be automatically encrypted and decrypted by the EC2 host. This solution does not require any additional IAM roles, tags, or policies.

Amazon EBS encryption

Creating an encrypted EBS volume

Encrypting an unencrypted EBS volume

Question 5

Question Type: MultipleChoice

A company runs a container application on a Kubernetes cluster in the company's data center The application uses Advanced Message Queuing Protocol (AMQP) to communicate with a message queue The data center cannot scale fast enough to meet the company's expanding business needs The company wants to migrate the workloads to AWS

Which solution will meet these requirements with the LEAST operational overhead? \

Options:

A- Migrate the container application to Amazon Elastic Container Service (Amazon ECS) Use Amazon Simple Queue Service (Amazon SQS) to retrieve the messages.

B- Migrate the container application to Amazon Elastic Kubernetes Service (Amazon EKS) Use Amazon MQ to retrieve the messages.

C- Use highly available Amazon EC2 instances to run the application Use Amazon MQ to retrieve the messages.

D- Use AWS Lambda functions to run the application Use Amazon Simple Queue Service (Amazon SQS) to retrieve the messages.

Answer:

B

Explanation:

This option is the best solution because it allows the company to migrate the container application to AWS with minimal changes and leverage a managed service to run the Kubernetes cluster and the message queue. By using Amazon EKS, the company can run the container application on a fully managed Kubernetes control plane that is compatible with the existing Kubernetes tools and plugins. Amazon EKS handles the provisioning, scaling, patching, and security of the Kubernetes cluster, reducing the operational overhead and complexity. By using Amazon MQ, the company can use a fully managed message broker service that supports AMQP and other popular messaging protocols. Amazon MQ handles the administration, maintenance, and scaling of the message broker, ensuring high availability, durability, and security of the messages.

A) Migrate the container application to Amazon Elastic Container Service (Amazon ECS) Use Amazon Simple Queue Service (Amazon SQS) to retrieve the messages. This option is not optimal because it requires the company to change the container orchestration platform from Kubernetes to ECS, which can introduce additional complexity and risk. Moreover, it requires the company to change the messaging protocol from AMQP to SQS, which can also affect the application logic and performance. Amazon ECS and Amazon SQS are both fully managed services that simplify the deployment and management of containers and messages, but they may not be compatible with the existing application architecture and requirements.

C) Use highly available Amazon EC2 instances to run the application Use Amazon MQ to retrieve the messages. This option is not ideal because it requires the company to manage the EC2 instances that host the container application. The company would need to provision, configure, scale, patch, and monitor the EC2 instances, which can increase the operational overhead and infrastructure costs. Moreover, the company would need to install and maintain the Kubernetes software on the EC2 instances, which can also add complexity and risk. Amazon MQ is a fully managed message broker service that supports AMQP and other popular messaging protocols, but it cannot compensate for the lack of a managed Kubernetes service.

D) Use AWS Lambda functions to run the application Use Amazon Simple Queue Service (Amazon SQS) to retrieve the messages. This option is not feasible because AWS Lambda does not support running container applications directly. Lambda functions are executed in

a sandboxed environment that is isolated from other functions and resources. To run container applications on Lambda, the company would need to use a custom runtime or a wrapper library that emulates the container API, which can introduce additional complexity and overhead. Moreover, Lambda functions have limitations in terms of available CPU, memory, and runtime, which may not suit the application needs. Amazon SQS is a fully managed message queue service that supports asynchronous communication, but it does not support AMQP or other messaging protocols.

1Amazon Elastic Kubernetes Service - Amazon Web Services

2Amazon MQ - Amazon Web Services

3Amazon Elastic Container Service - Amazon Web Services

4AWS Lambda FAQs - Amazon Web Services

Question 6

Question Type: **MultipleChoice**

A solutions architect is designing a shared storage solution for a web application that is deployed across multiple Availability Zones The web application runs on Amazon EC2 instances that are in an Auto Scaling group The company plans to make frequent changes to the content The solution must have strong consistency in returning the new content as soon as the changes occur.

Which solutions meet these requirements? (Select TWO)

Options:

A- Use AWS Storage Gateway Volume Gateway Internet Small Computer Systems Interface (iSCSI) block storage that is mounted to the individual EC2 instances

B- Create an Amazon Elastic File System (Amazon EFS) file system Mount the EFS file system on the individual EC2 instances

C- Create a shared Amazon Elastic Block Store (Amazon EBS) volume. Mount the EBS volume on the individual EC2 instances.

D- Use AWS DataSync to perform continuous synchronization of data between EC2 hosts in the Auto Scaling group

E- Create an Amazon S3 bucket to store the web content Set the metadata for the Cache-Control header to no-cache Use Amazon CloudFront to deliver the content

Answer:

B, E

Explanation:

These options are the most suitable ways to design a shared storage solution for a web application that is deployed across multiple Availability Zones and requires strong consistency. Option B uses Amazon Elastic File System (Amazon EFS) as a shared file system that can be mounted on multiple EC2 instances in different Availability Zones. Amazon EFS provides high availability, durability, scalability, and performance for file-based workloads. It also supports strong consistency, which means that any changes made to the file system are immediately visible to all clients. Option E uses Amazon S3 as a shared object store that can store the web content and serve it through Amazon CloudFront, a content delivery network (CDN). Amazon S3 provides high availability, durability, scalability, and performance for object-based workloads. It also supports strong consistency for read-after-write and list operations, which means that

any changes made to the objects are immediately visible to all clients. By setting the metadata for the Cache-Control header to no-cache, the web content can be prevented from being cached by the browsers or the CDN edge locations, ensuring that the latest content is always delivered to the users.

Option A is not suitable because using AWS Storage Gateway Volume Gateway as a shared storage solution for a web application is not efficient or scalable. AWS Storage Gateway Volume Gateway is a hybrid cloud storage service that provides block storage volumes that can be mounted on-premises or on EC2 instances as iSCSI devices. It is useful for migrating or backing up data to AWS, but it is not designed for serving web content or providing strong consistency. Moreover, using Volume Gateway would incur additional costs and complexity, and it would not leverage the native AWS storage services.

Option C is not suitable because creating a shared Amazon EBS volume and mounting it on multiple EC2 instances is not possible or reliable. Amazon EBS is a block storage service that provides persistent and high-performance volumes for EC2 instances. However, EBS volumes can only be attached to one EC2 instance at a time, and they are constrained to a single Availability Zone. Therefore, creating a shared EBS volume for a web application that is deployed across multiple Availability Zones is not feasible. Moreover, EBS volumes do not support strong consistency, which means that any changes made to the volume may not be immediately visible to other clients.

Option D is not suitable because using AWS DataSync to perform continuous synchronization of data between EC2 hosts in the Auto Scaling group is not efficient or scalable. AWS DataSync is a data transfer service that helps you move large amounts of data to and from AWS storage services. It is useful for migrating or archiving data, but it is not designed for serving web content or providing strong consistency. Moreover, using DataSync would incur additional costs and complexity, and it would not leverage the native AWS storage services.Reference:

What Is Amazon Elastic File System?

What Is Amazon Simple Storage Service?

What Is Amazon CloudFront?

What Is AWS Storage Gateway?

What Is Amazon Elastic Block Store?

What Is AWS DataSync?

Question 7

Question Type: MultipleChoice

A company has multiple AWS accounts with applications deployed in the us-west-2 Region Application logs are stored within Amazon S3 buckets in each account The company wants to build a centralized log analysis solution that uses a single S3 bucket Logs must not leave us-west-2, and the company wants to incur minimal operational overhead

Which solution meets these requirements and is MOST cost-effective?

Options:

A- Create an S3 Lifecycle policy that copies the objects from one of the application S3 buckets to the centralized S3 bucket

B- Use S3 Same-Region Replication to replicate logs from the S3 buckets to another S3 bucket in us-west-2 Use this S3 bucket for log

analysis.

C- Write a script that uses the PutObject API operation every day to copy the entire contents of the buckets to another S3 bucket in us-west-2 Use this S3 bucket for log analysis.

D- Write AWS Lambda functions in these accounts that are triggered every time logs are delivered to the S3 buckets (s3 ObjectCreated a event) Copy the logs to another S3 bucket in us-west-2. Use this S3 bucket for log analysis.

Answer:

B

Explanation:

This solution meets the following requirements:

It is cost-effective, as it only charges for the storage and data transfer of the replicated objects, and does not require any additional AWS services or custom scripts. S3 Same-Region Replication (SRR) is a feature that automatically replicates objects across S3 buckets within the same AWS Region. SRR can help you aggregate logs from multiple sources to a single destination for analysis and auditing. SRR also preserves the metadata, encryption, and access control of the source objects.

It is operationally efficient, as it does not require any manual intervention or scheduling. SRR replicates objects as soon as they are uploaded to the source bucket, ensuring that the destination bucket always has the latest log data. SRR also handles any updates or deletions of the source objects, keeping the destination bucket in sync. SRR can be enabled with a few clicks in the S3 console or with a simple API call.

It is secure, as it does not allow the logs to leave the us-west-2 Region. SRR only replicates objects within the same AWS Region, ensuring that the data sovereignty and compliance requirements are met. SRR also supports encryption of the source and destination objects, using either server-side encryption with AWS KMS or S3-managed keys, or client-side encryption.

Same-Region Replication - Amazon Simple Storage Service

How do I replicate objects across S3 buckets in the same AWS Region?

Centralized Logging on AWS | AWS Solutions | AWS Solutions Library

Question 8

Question Type: MultipleChoice

A company is creating an application The company stores data from tests of the application in multiple on-premises locations

The company needs to connect the on-premises locations to VPCs in an AWS Region in the AWS Cloud The number of accounts and VPCs will increase during the next year The network architecture must simplify the administration of new connections and must provide the ability to scale.

Which solution will meet these requirements with the LEAST administrative overhead'?

Options:

A- Create a peering connection between the VPCs Create a VPN connection between the VPCs and the on-premises locations.

B- Launch an Amazon EC2 instance On the instance, include VPN software that uses a VPN connection to connect all VPCs and on-premises locations.

C- Create a transit gateway Create VPC attachments for the VPC connections Create VPN attachments for the on-premises connections.

D- Create an AWS Direct Connect connection between the on-premises locations and a central VPC. Connect the central VPC to other VPCs by using peering connections.

Answer:

C

Explanation:

A transit gateway is a network transit hub that enables you to connect your VPCs and on-premises networks in a centralized and scalable way. You can create VPC attachments to connect your VPCs to the transit gateway, and VPN attachments to connect your on-premises networks to the transit gateway over the internet. The transit gateway acts as a router between the attached networks, and simplifies the administration of new connections by reducing the number of peering or VPN connections required. You can also use transit gateway route tables to control the routing of traffic between the attached networks. By creating a transit gateway and using VPC and VPN attachments, you can meet the requirements of the company with the least administrative overhead.

AWS Transit Gateway

Transit gateway attachments

Transit gateway route tables

Question 9

Question Type: MultipleChoice

A company built an application with Docker containers and needs to run the application in the AWS Cloud The company wants to use a managed sen/ice to host the application

The solution must scale in and out appropriately according to demand on the individual container services The solution also must not result in additional operational overhead or infrastructure to manage

Which solutions will meet these requirements? (Select TWO)

Options:

A- Use Amazon Elastic Container Service (Amazon ECS) with AWS Fargate.

B- Use Amazon Elastic Kubernetes Service (Amazon EKS) with AWS Fargate.

C- Provision an Amazon API Gateway API Connect the API to AWS Lambda to run the containers.

D- Use Amazon Elastic Container Service (Amazon ECS) with Amazon EC2 worker nodes.

E- Use Amazon Elastic Kubernetes Service (Amazon EKS) with Amazon EC2 worker nodes.

Answer:

A, B

Explanation:

These options are the best solutions because they allow the company to run the application with Docker containers in the AWS Cloud using a managed service that scales automatically and does not require any infrastructure to manage. By using AWS Fargate, the company can launch and run containers without having to provision, configure, or scale clusters of EC2 instances. Fargate allocates the right amount of compute resources for each container and scales them up or down as needed. By using Amazon ECS or Amazon EKS, the company can choose the container orchestration platform that suits its needs. Amazon ECS is a fully managed service that integrates with other AWS services and simplifies the deployment and management of containers. Amazon EKS is a managed service that runs Kubernetes on AWS and provides compatibility with existing Kubernetes tools and plugins.

C) Provision an Amazon API Gateway API Connect the API to AWS Lambda to run the containers. This option is not feasible because AWS Lambda does not support running Docker containers directly. Lambda functions are executed in a sandboxed environment that is isolated from other functions and resources. To run Docker containers on Lambda, the company would need to use a custom runtime or a wrapper library that emulates the Docker API, which can introduce additional complexity and overhead.

D) Use Amazon Elastic Container Service (Amazon ECS) with Amazon EC2 worker nodes. This option is not optimal because it requires the company to manage the EC2 instances that host the containers. The company would need to provision, configure, scale, patch, and monitor the EC2 instances, which can increase the operational overhead and infrastructure costs.

E) Use Amazon Elastic Kubernetes Service (Amazon EKS) with Amazon EC2 worker nodes. This option is not ideal because it requires the company to manage the EC2 instances that host the containers. The company would need to provision, configure, scale, patch, and monitor the EC2 instances, which can increase the operational overhead and infrastructure costs.

1AWS Fargate - Amazon Web Services

2Amazon Elastic Container Service - Amazon Web Services

3Amazon Elastic Kubernetes Service - Amazon Web Services

4AWS Lambda FAQs - Amazon Web Services

Question 10

Question Type: MultipleChoice

A company is building an Amazon Elastic Kubernetes Service (Amazon EKS) cluster for its workloads. All secrets that are stored in Amazon EKS must be encrypted in the Kubernetes etcd key-value store.

Which solution will meet these requirements?

Options:

A- Create a new AWS Key Management Service (AWS KMS) key Use AWS Secrets Manager to manage rotate, and store all secrets in Amazon EKS.

B- Create a new AWS Key Management Service (AWS KMS) key Enable Amazon EKS KMS secrets encryption on the Amazon EKS cluster.

C- Create the Amazon EKS cluster with default options Use the Amazon Elastic Block Store (Amazon EBS) Container Storage Interface (CSI) driver as an add-on.

D- Create a new AWS Key Management Service (AWS KMS) key with the ahas/aws/ebs alias Enable default Amazon Elastic Block Store (Amazon EBS) volume encryption for the account.

Answer:

B

Explanation:

This option is the most secure and simple way to encrypt the secrets that are stored in Amazon EKS. AWS Key Management Service (AWS KMS) is a service that allows you to create and manage encryption keys that can be used to encrypt your data. Amazon EKS KMS secrets encryption is a feature that enables you to use a KMS key to encrypt the secrets that are stored in the Kubernetes etcd key-value store. This provides an additional layer of protection for your sensitive data, such as passwords, tokens, and keys. You can create a new KMS key or use an existing one, and then enable the Amazon EKS KMS secrets encryption on the Amazon EKS cluster. You can also use IAM policies to control who can access or use the KMS key.

Option A is not correct because using AWS Secrets Manager to manage, rotate, and store all secrets in Amazon EKS is not necessary or efficient. AWS Secrets Manager is a service that helps you securely store, retrieve, and rotate your secrets, such as database

credentials, API keys, and passwords. You can use it to manage secrets that are used by your applications or services outside of Amazon EKS, but it is not designed to encrypt the secrets that are stored in the Kubernetes etcd key-value store. Moreover, using AWS Secrets Manager would incur additional costs and complexity, and it would not leverage the native Kubernetes secrets management capabilities.

Option C is not correct because using the Amazon EBS Container Storage Interface (CSI) driver as an add-on does not encrypt the secrets that are stored in Amazon EKS. The Amazon EBS CSI driver is a plugin that allows you to use Amazon EBS volumes as persistent storage for your Kubernetes pods. It is useful for providing durable and scalable storage for your applications, but it does not affect the encryption of the secrets that are stored in the Kubernetes etcd key-value store. Moreover, using the Amazon EBS CSI driver would require additional configuration and resources, and it would not provide the same level of security as using a KMS key.

Option D is not correct because creating a new AWS KMS key with the alias aws/ebs and enabling default Amazon EBS volume encryption for the account does not encrypt the secrets that are stored in Amazon EKS. The alias aws/ebs is a reserved alias that is used by AWS to create a default KMS key for your account. This key is used to encrypt the Amazon EBS volumes that are created in your account, unless you specify a different KMS key. Enabling default Amazon EBS volume encryption for the account is a setting that ensures that all new Amazon EBS volumes are encrypted by default. However, these features do not affect the encryption of the secrets that are stored in the Kubernetes etcd key-value store. Moreover, using the default KMS key or the default encryption setting would not provide the same level of control and security as using a custom KMS key and enabling the Amazon EKS KMS secrets encryption feature.Reference:

Encrypting secrets used in Amazon EKS

What Is AWS Key Management Service?

What Is AWS Secrets Manager?

Amazon EBS CSI driver

Encryption at rest