IAM

timeouts are usually caused by blocked firewall access.

IAM roles and resource-based policies delegate access across accounts only within a single partition

**Use Envelope Encryption and reference the data as file within the code**

While AWS KMS does support sending data up to 4 KB to be encrypted directly, envelope encryption can offer significant performance benefits. When you encrypt data directly with AWS KMS it must be transferred over the network. Envelope encryption reduces the network load since only the request and delivery of the much smaller data key go over the network. The data key is used locally in your application or encrypting AWS service, avoiding the need to send the entire block of data to AWS KMS and suffer network latency.

AWS Lambda environment variables can have a maximum size of 4 KB. Additionally, the direct 'Encrypt' API of KMS also has an upper limit of 4 KB for the data payload. To encrypt 1 MB, you need to use the Encryption SDK and pack the encrypted file with the lambda function.

**AWS Certificate Manager** - AWS Certificate Manager (ACM) is the preferred tool to provision, manage, and deploy server certificates. With ACM you can request a certificate or deploy an existing ACM or external certificate to AWS resources. Certificates provided by ACM are free and automatically renew. In a supported Region, you can use ACM to manage server certificates from the console or programmatically.

**IAM** - IAM is used as a certificate manager only when you must support HTTPS connections in a Region that is not supported by ACM. IAM securely encrypts your private keys and stores the encrypted version in IAM SSL certificate storage. IAM supports deploying server certificates in all Regions, but you must obtain your certificate from an external provider for use with AWS. You cannot upload an ACM certificate to IAM. Additionally, you cannot manage your certificates from the IAM Console.

**AWS Secrets Manager** - AWS Secrets Manager helps you protect secrets needed to access your applications, services, and IT resources. The service enables you to easily rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle. Users and applications retrieve secrets with a call to Secrets Manager APIs, eliminating the need to hardcode sensitive information in plain text. It cannot be used to discover and protect your sensitive data in AWS.

**AWS Systems Manager** - AWS Systems Manager gives you visibility and control of your infrastructure on AWS. Systems Manager provides a unified user interface so you can view operational data from multiple AWS services and allows you to automate operational tasks such as running commands, managing patches, and configuring servers across AWS Cloud as well as on-premises infrastructure.

**SMS text message-based MFA** - A type of MFA in which the IAM user settings include the phone number of the user's SMS-compatible mobile device. When the user signs in, AWS sends a six-digit numeric code by SMS text message to the user's mobile device. The user is required to type that code on a second webpage during sign-in. SMS-based MFA is available only for IAM users, you cannot use this type of MFA with the AWS account root user.

**Hardware MFA device** - This hardware device generates a six-digit numeric code. The user must type this code from the device on a second webpage during sign-in. Each MFA device assigned to a user must be unique. A user cannot type a code from another user's device to be authenticated. Can be used for root user authentication.

**U2F security key** - A device that you plug into a USB port on your computer. U2F is an open authentication standard hosted by the FIDO Alliance. When you enable a U2F security key, you sign in by entering your credentials and then tapping the device instead of manually entering a code.

**Virtual MFA devices** - A software app that runs on a phone or other device and emulates a physical device. The device generates a six-digit numeric code. The user must type a valid code from the device on a second webpage during sign-in. Each virtual MFA device assigned to a user must be unique. A user cannot type a code from another user's virtual MFA device to authenticate.

AWS Budgets lets customers set custom budgets and receive alerts if their costs or usage exceed (or are forecasted to exceed) their budgeted amount.

**AWS requires approximately 5 weeks of usage data to generate budget forecasts** - AWS requires approximately 5 weeks of usage data to generate budget forecasts. If you set a budget to alert based on a forecasted amount, this budget alert isn't triggered until you have enough historical usage information.

**Budget forecast has been created from an account that does not have enough privileges** - This is an incorrect statement. If the user account does not have enough privileges, the user will not be able to create the budget at all.

**Access Advisor feature on IAM console**- To help identify the unused roles, IAM reports the last-used timestamp that represents when a role was last used to make an AWS request. Your security team can use this information to identify, analyze, and then confidently remove unused roles. This helps improve the security posture of your AWS environments. Additionally, by removing unused roles, you can simplify your monitoring and auditing efforts by focusing only on roles that are in use.

**AWS Trusted Advisor** - AWS Trusted Advisor is an online tool that provides you real-time guidance to help you provision your resources following AWS best practices on cost optimization, security, fault tolerance, service limits, and performance improvement.

**IAM Access Analyzer** - AWS IAM Access Analyzer helps you identify the resources in your organization and accounts, such as Amazon S3 buckets or IAM roles, that are shared with an external entity. This lets you identify unintended access to your resources and data, which is a security risk.

**Amazon Inspector** - Amazon Inspector is an automated security assessment service that helps improve the security and compliance of applications deployed on AWS. Amazon Inspector automatically assesses applications for exposure, vulnerabilities, and deviations from best practices.

* IAM roles and resource-based policies delegate access across accounts only within a single partition
* **Create an IAM role in account B with access to DynamoDB. Modify the trust policy of the role in Account B to allow the execution role of Lambda to assume this role. Update the Lambda function code to add the AssumeRole API call.**

**Trust policy** - Trust policies define which principal entities (accounts, users, roles, and federated users) can assume the role. An IAM role is both an identity and a resource that supports resource-based policies. For this reason, you must attach both a trust policy and an identity-based policy to an IAM role. The IAM service supports only one type of resource-based policy called a role trust policy, which is attached to an IAM role.

**AWS Organizations Service Control Policies (SCP)** - If you enable all features of AWS organization, then you can apply service control policies (SCPs) to any or all of your accounts. SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU). The SCP limits permissions for entities in member accounts, including each AWS account root user. An explicit deny in any of these policies overrides the allow.

**Access control list (ACL)** - Access control lists (ACLs) are service policies that allow you to control which principals in another account can access a resource. ACLs cannot be used to control access for a principal within the same account. Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs.

**Permissions boundary** - AWS supports permissions boundaries for IAM entities (users or roles). A permissions boundary is an advanced feature for using a managed policy to set the maximum permissions that an identity-based policy can grant to an IAM entity. An entity's permissions boundary allows it to perform only the actions that are allowed by both its identity-based policies and its permissions boundaries.

To calculate the concurrency requirements for the Lambda function simply multiply the number of executions per second (20) by the time it takes to complete the execution (20).

Therefore, for this scenario, the calculation is 20 x 20 = 400.

**KMS**

**KMS stores the CMK, and receives data from the clients, which it encrypts and sends back**

CLOUDWATCH

**ApproximateNumberOfMessagesVisible** - This is a CloudWatch Amazon SQS queue metric. The number of messages in a queue might not change proportionally to the size of the Auto Scaling group that processes messages from the queue. Hence, this metric does not work for target tracking.

The minimum permissions required are as follows:

· “logs:CreateLogGroup” - Creates a log group with the specified name.

· “logs:CreateLogStream” - Creates a log stream for the specified log group.

· “logs:PutLogEvents” - Uploads a batch of log events to the specified log stream.

LAMBDA

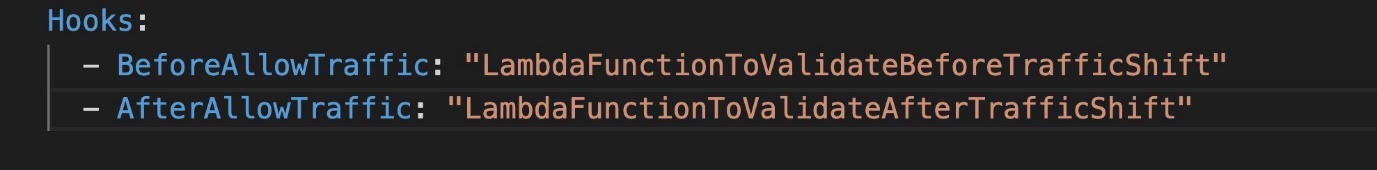
A layer is a ZIP archive that contains libraries, a custom runtime, or other dependencies. Layers will not help in configuring access to RDS instance and hence is an incorrect choice.

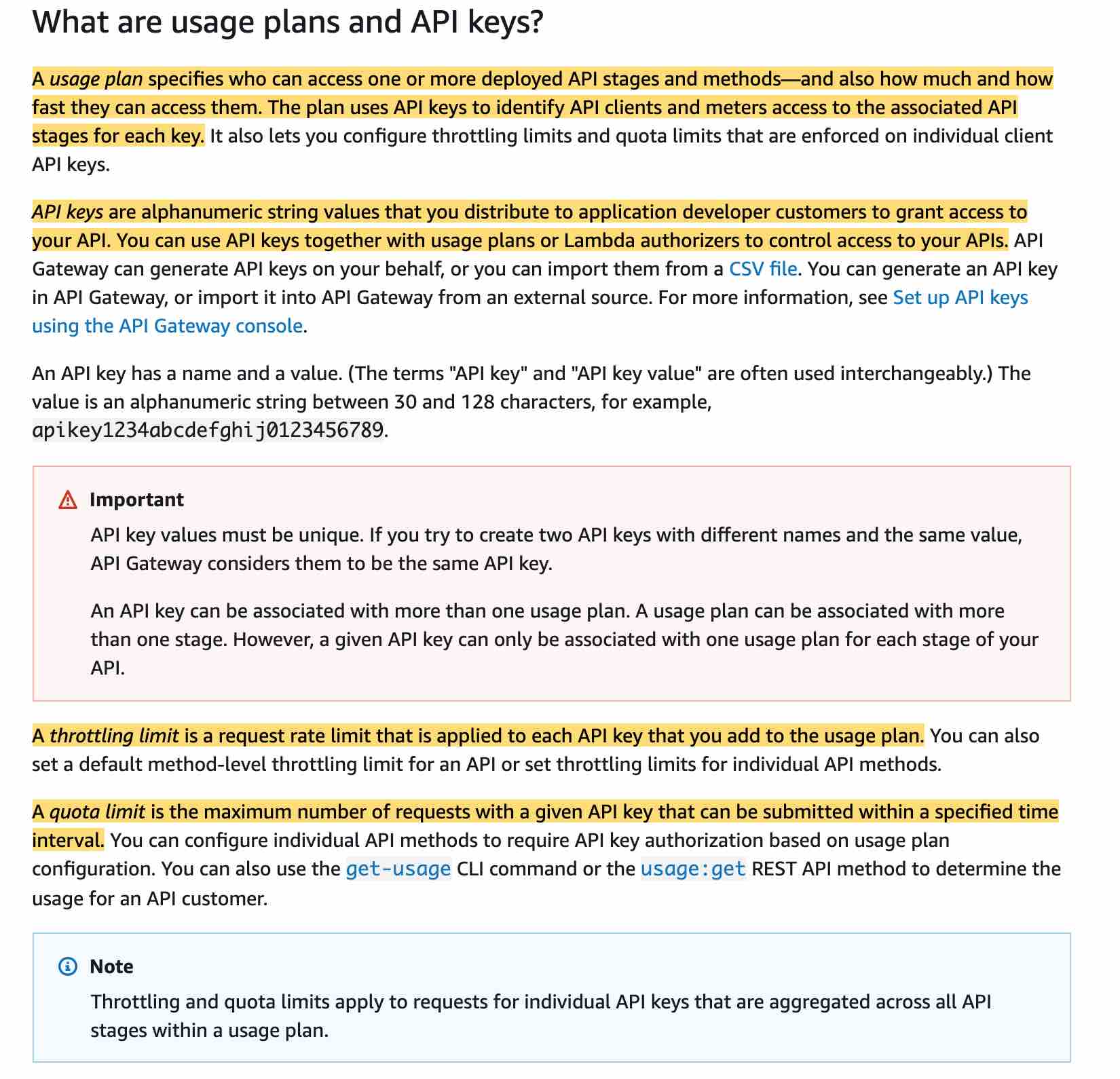
The content in the 'hooks' section of the AppSpec file varies, depending on the compute platform for your deployment. The 'hooks' section for an EC2/On-Premises deployment contains mappings that link deployment lifecycle event hooks to one or more scripts.

The 'hooks' section for a Lambda or an Amazon ECS deployment specifies Lambda validation functions to run during a deployment lifecycle event. If an event hook is not present, no operation is executed for that event. This section is required only if you are running scripts or Lambda validation functions as part of the deployment.

The following code snippet shows a valid example of the structure of hooks for an AWS Lambda deployment:

Therefore, in this scenario a valid structure for the order of hooks that should be specified in the appspec.yml file is: BeforeAllowTraffic > AfterAllowTraffic





**Lambda Versions** - Versions are immutable and cannot be updated over time.

* Stage variables are name-value pairs that you can define as configuration attributes associated with a deployment stage of an API. They act like environment variables and can be used in your API setup and mapping templates
* A Lambda alias is like a pointer to a specific Lambda function version. Users can access the function version using the alias ARN.
* Mapping template overrides provides you with the flexibility to perform many-to-one parameter mappings; override parameters after standard API Gateway mappings have been applied; conditionally map parameters based on body content or other parameter values; programmatically create new parameters on the fly, and override status codes returned by your integration endpoint.
* You can configure a Lambda function to connect to private subnets in a virtual private cloud (VPC) in your account. Use Amazon Virtual Private Cloud (Amazon VPC) to create a private network for resources such as databases, cache instances, or internal services. Connect your lambda function to the VPC to access private resources during execution.
* **Set up reserved concurrency for the Lambda function B so that it throttles if it goes above a certain concurrency limit**
* Concurrency is the number of requests that a Lambda function is serving at any given time. If a Lambda function is invoked again while a request is still being processed, another instance is allocated, which increases the function's concurrency.
* To ensure that a function can always reach a certain level of concurrency, you can configure the function with reserved concurrency. When a function has reserved concurrency, no other function can use that concurrency. More importantly, reserved concurrency also limits the maximum concurrency for the function, and applies to the function as a whole, including versions and aliases.
* Please review this note to understand how reserved concurrency works:
* You should use provisioned concurrency to enable your function to scale without fluctuations in latency. By allocating provisioned concurrency before an increase in invocations, you can ensure that all requests are served by initialized instances with very low latency. Provisioned concurrency is not used to limit the maximum concurrency for a given Lambda function,

KINESIS DATA STREAMS

Amazon Kinesis Data Streams (KDS) is a massively scalable and durable real-time data streaming service. Amazon Kinesis Data Streams is recommended when you need the ability for multiple applications to consume the same stream concurrently.

It provides ordering of records, as well as the ability to read and/or replay records in the same order to multiple Amazon Kinesis Applications. The Amazon Kinesis Client Library (KCL) delivers all records for a given partition key to the same record processor, making it easier to build multiple applications reading from the same Amazon Kinesis data stream (for example, to perform counting, aggregation, and filtering)

**SIMPLE WORKFLOW SERVICE [SWF]**

SWF ensures that a task is assigned only once and is never duplicated. Since Amazon SWF maintains the application’s state durably, workers and deciders don’t have to keep track of execution state. SWF is a workflow solution whereas SQS is a message queue mechanism for decoupling system architecture

The Amazon Simple Workflow Service (Amazon SWF) makes it easy to build applications that coordinate work across distributed components.

**CLOUDFORMATION**

Parameter types enable CloudFormation to validate inputs earlier in the stack creation process.

CloudFormation currently supports the following parameter types:

String – A literal string

Number – An integer or float

List<Number> – An array of integers or floats

CommaDelimitedList – An array of literal strings that are separated by commas

AWS::EC2::KeyPair::KeyName – An Amazon EC2 key pair name

AWS::EC2::SecurityGroup::Id – A security group ID

AWS::EC2::Subnet::Id – A subnet ID

AWS::EC2::VPC::Id – A VPC ID

List<AWS::EC2::VPC::Id> – An array of VPC IDs

List<AWS::EC2::SecurityGroup::Id> – An array of security group IDs

List<AWS::EC2::Subnet::Id> – An array of subnet IDs

Conditions cannot be used within the Parameters section. After you define all your conditions, you can associate them with resources and resource properties only in the Resources and Outputs sections of a template.

**Stack B, then Stack C, then Stack A**

All of the imports must be removed before you can delete the exporting stack or modify the output value. In this case, you must delete either Stack B or Stack C, then you delete Stack A.

**DYNAMODB**

**Consider using Global tables if your application is accessed by globally distributed users** - If you have globally dispersed users, consider using global tables. With global tables, you can specify the AWS Regions where you want the table to be available. This can significantly reduce latency for your users. So, reducing the distance between the client and the DynamoDB endpoint is an important performance fix to be considered.

For this scenario the key requirement is to ensure the data is not lost. Therefore, the data must be stored in a durable data store outside of the EC2 instances. Amazon DynamoDB is a suitable solution for storing session data. DynamoDB has a session handling capability for multiple languages as in the below example for PHP:

“The **DynamoDB Session Handler** is a custom session handler for PHP that allows developers to use Amazon DynamoDB as a session store. Using DynamoDB for session storage alleviates issues that occur with session handling in a distributed web application by moving sessions off of the local file system and into a shared location. DynamoDB is fast, scalable, easy to setup, and handles replication of your data automatically.”

Firstly, the Limit parameter can be used to reduce the page size. The Scan operation provides a Limit parameter that you can use to set the page size for your request. Each Query or Scan request that has a smaller page size uses fewer read operations and creates a "pause" between each request.

Secondly, the Developer can configure parallel scans. With parallel scans the Developer can maximize usage of the available throughput and have the scans distributed across the table’s partitions.

A parallel scan can be the right choice if the following conditions are met:

The table size is 20 GB or larger.

The table's provisioned read throughput is not being fully used.

Sequential Scan operations are too slow.

Therefore, to optimize the scan operation the Developer should use parallel scans while limiting the rate as this will ensure that the scan operation does not affect the performance of production workloads and still have it complete in the minimum time.

**Reduce page size**

Because a Scan operation reads an entire page (by default, 1 MB), you can reduce the impact of the scan operation by setting a smaller page size. The Scan operation provides a Limit parameter that you can use to set the page size for your request. Each Query or Scan request that has a smaller page size uses fewer read operations and creates a "pause" between each request.

**Isolate scan operations**

DynamoDB is designed for easy scalability. As a result, an application can create tables for distinct purposes, possibly even duplicating content across several tables. You want to perform scans on a table that is not taking "mission-critical" traffic. Some applications handle this load by rotating traffic hourly between two tables—one for critical traffic, and one for bookkeeping. Other applications can do this by performing every write on two tables: a "mission-critical" table, and a "shadow" table.

Therefore, the best option to reduce the impact of the scan on the table's provisioned throughput is to set a smaller page size for the scan.

DynamoDB Streams captures a time-ordered sequence of item-level modifications in any DynamoDB table and stores this information in a log for up to 24 hours. Applications can access this log and view the data items as they appeared before and after they were modified, in near-real time.

You can also use the CreateTable or UpdateTable API operations to enable or modify a stream. The StreamSpecification parameter determines how the stream is configured:

**StreamEnabled** — Specifies whether a stream is enabled (true) or disabled (false) for the table.

**StreamViewType**— Specifies the information that will be written to the stream whenever data in the table is modified:

KEYS\_ONLY — Only the key attributes of the modified item.

NEW\_IMAGE — The entire item, as it appears after it was modified.

OLD\_IMAGE — The entire item, as it appeared before it was modified.

NEW\_AND\_OLD\_IMAGES — Both the new and the old images of the item.

In this scenario, we only need to keep a copy of the items before they were modified. Therefore, the solution is to enable DynamoDB streams and set the StreamViewType to OLD\_IMAGES.

**CLOUD TRAIL**

If you have created an organization in AWS Organizations, you can also create a trail that will log all events for all AWS accounts in that organization. This is referred to as an organization trail.

**By default, CloudTrail tracks only bucket-level actions. To track object-level actions, you need to enable Amazon S3 data events.**

AWS CloudTrail supports Amazon S3 Data Events, apart from bucket Events.

Organization trails must be created in the master account, and when specified as applying to an organization, are automatically applied to all member accounts in the organization. Member accounts will be able to see the organization trail, but cannot modify or delete it. By default, member accounts will not have access to the log files for the organization trail in the Amazon S3 bucket.

**SECRET MANAGER**

\*Use Secrets Manager \* - AWS Secrets Manager enables you to easily rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle. Users and applications retrieve secrets with a call to Secrets Manager APIs, eliminating the need to hardcode sensitive information in plain text. Secrets Manager offers secret rotation with built-in integration for Amazon RDS, Amazon Redshift, and Amazon DocumentDB. Secrets Manager cannot be used for encrypting data at rest.

**SQS**

SNS and SQS can be used to create a fanout messaging scenario in which messages are "pushed" to multiple subscribers, which eliminates the need to periodically check or poll for updates and enables parallel asynchronous processing of the message by the subscribers. SQS can allow for later re-processing and dead letter queues. This is called the fan-out pattern.

**SNS**

**SNS + Kinesis** - You can use Amazon Kinesis Data Streams to collect and process large streams of data records in real-time. The only issue is that the retention day period is 7 days, and processing issues would completely block all future messages.

**NETWORKING**

**The network ACLs associated with the subnet must have rules to allow inbound and outbound traffic** - The network access control lists (ACLs) that are associated with the subnet must have rules to allow inbound and outbound traffic on port 80 (for HTTP traffic) and port 443 (for HTTPs traffic). This is a necessary condition for Internet Gateway connectivity

**The route table in the instance’s subnet should have a route to an Internet Gateway** - A route table contains a set of rules, called routes, that are used to determine where network traffic from your subnet or gateway is directed. The route table in the instance’s subnet should have a route defined to the Internet Gateway.

Incorrect options:

**The instance's subnet is not associated with any route table** - This is an incorrect statement. A subnet is implicitly associated with the main route table if it is not explicitly associated with a particular route table. So, a subnet is always associated with some route table.

**The instance's subnet is associated with multiple route tables with conflicting configurations** - This is an incorrect statement. A subnet can only be associated with one route table at a time.

**API GATEWAY**

**Assign a Security Group to your API Gateway** - API Gateway does not use security groups but uses resource policies, which are JSON policy documents that you attach to an API to control whether a specified principal (typically an IAM user or role) can invoke the API. You can restrict IP address using this, the downside being, an IP address can be changed by the accessing user. So, this is not an optimal solution for the current use case.