# APPLICATION INTEGRATION

## AWS APPLICATION INTEGRATION SERVICES

### AWS APPLICATION INTEGRATION SERVICES

Los servicios de integración de aplicaciones de AWS son una familia de servicios que permiten la comunicación desacoplada entre aplicaciones.

Estos servicios proporcionan desacoplamiento para microservicios, sistemas distribuidos y aplicaciones sin servidor.

Los servicios de integración de aplicaciones de AWS le permiten conectar aplicaciones sin necesidad de escribir código personalizado para habilitar la interoperabilidad.

Las aplicaciones desacopladas pueden interoperar mientras son resistentes a la falla o sobrecarga de cualquier componente individual.

Los siguientes servicios están involucrados con la integración de aplicaciones:

|  |  |  |
| --- | --- | --- |
| **Service** | **What it does** | **Example use cases** |
| Simple Queue Service (SQS) | Cola de mensajería; patrones de almacenamiento y reenvío | Creación de aplicaciones distribuidas / desacopladas |
| Simple Notification Service (SNS) | Configurar, operar y enviar notificaciones desde la nube | Enviar una notificación por correo electrónico cuando se activa la alarma de CloudWatch |
| Step Functions | Coordinación lista para usar de los componentes de servicio de AWS con flujo de trabajo visual | Flujo de trabajo de procesamiento de pedidos |
| Simple Workflow Service (SWF) | Necesidad de soportar procesos externos o lógica de ejecución especializada | Flujos de trabajo habilitados por humanos, como un sistema de cumplimiento de pedidos o para solicitudes de procedimiento  ***Nota****: AWS recomienda que, para las nuevas aplicaciones, los clientes consideren Step Functions en lugar de SWF* |
| Amazon MQ | Servicio de agente de mensajes para Apache Active MQ y RabbitMQ | Necesita una cola de mensajes que admita API y protocolos estándar de la industria; migrar colas a AWS |
| Amazon Kinesis | Recopile, procese y analice datos de transmisión. | Recopilar datos de dispositivos IoT para su posterior procesamiento |

#### 

#### AMAZON SNS

Amazon Simple Notification Service (Amazon SNS) es un servicio de mensajería totalmente administrado para la comunicación de aplicación a aplicación (A2A) y de aplicación a persona (A2P).

La funcionalidad pub/sub proporciona mensajería para casos de uso de alto rendimiento, basados en push y de muchos a muchos.

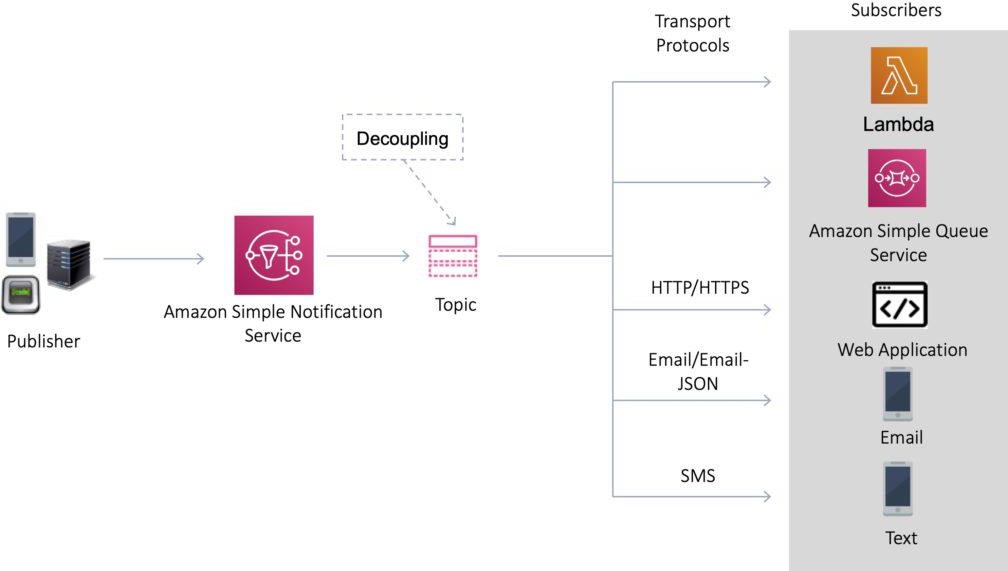
Amazon SNS se utiliza para enviar notificaciones entre sistemas distribuidos, microservicios y aplicaciones sin servidor controladas por eventos.

Las notificaciones push pueden ir a dispositivos Apple, Google, Fire OS y Windows, así como a dispositivos Android en China con Baidu Cloud Push.

Amazon SNS también puede enviar notificaciones a través de mensajes de texto SMS, correo electrónico, colas SQS o a cualquier punto de enlace HTTP.

Las notificaciones de Amazon SNS también pueden activar funciones de Lambda.

Amazon SNS es económico y se basa en un modelo de pago por uso sin costos iniciales. SNS utiliza un modelo pub-sub mediante el cual los usuarios o las aplicaciones se suscriben a temas de SNS.



Amazon SNS proporciona desacoplamiento de las aplicaciones para que los mensajes se puedan procesar de forma asincrónica.

#### SNS Topics

Se pueden agrupar varios destinatarios mediante Temas.

Un tema es un "punto de acceso" para permitir a los destinatarios suscribirse dinámicamente para obtener copias idénticas de

la misma notificación.

Un tema puede admitir entregas a varios tipos de endpoints.

Todos los mensajes se almacenan de forma redundante en varias zonas de disponibilidad. Proporciona entrega instantánea basada en push.

La entrega flexible de mensajes se proporciona a través de múltiples protocolos de transporte.

#### SNS Subscribers and Endpoints

Al suscribirse a un tema de SNS, se admiten los siguientes tipos de extremos:

* HTTP/HTTPS.
* Email/Email-JSON.
* Amazon Kinesis Data Firehose.
* Amazon SQS.
* AWS Lambda.
* Platform application endpoint (mobile push).
* SMS.

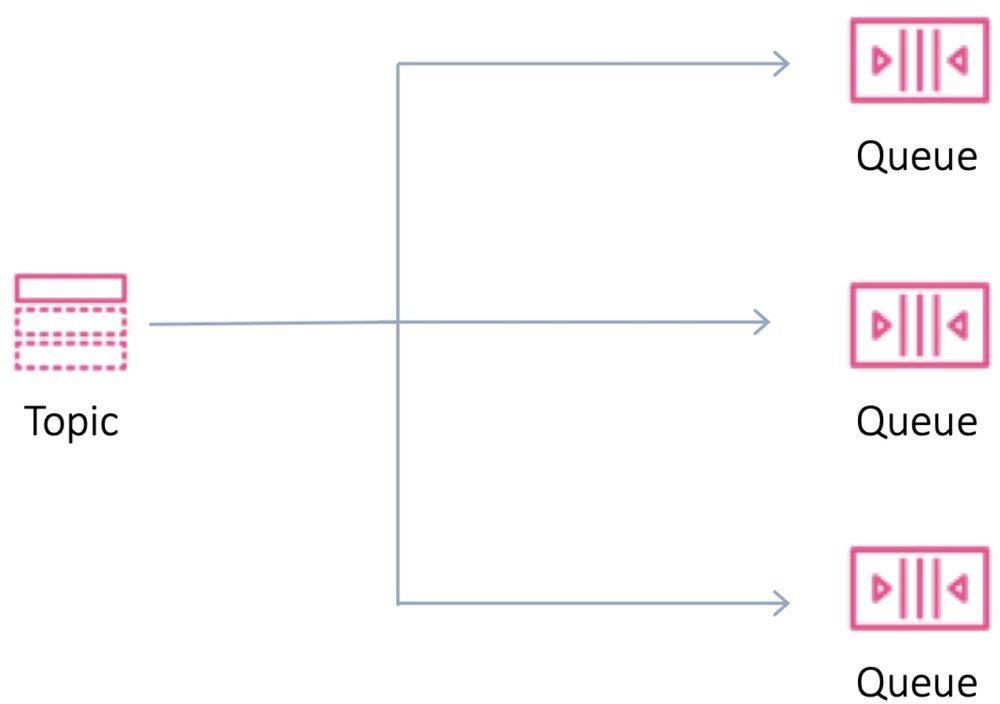
#### SNS Fanout

Los sistemas de los editores pueden distribuir mensajes a muchos sistemas de suscriptores, incluidas las colas de Amazon SQS, las funciones de AWS Lambda y los puntos de enlace HTTPS, para el procesamiento paralelo, y Amazon Kinesis Data Firehose.

Puede suscribir una o más colas de Amazon SQS a un tema de Amazon SNS a partir de una lista de temas disponibles para la cola seleccionada.

Amazon SQS administra la suscripción y los permisos necesarios.

Cuando publica un mensaje en un tema, Amazon SNS envía el mensaje a cada cola suscrita.



Fanout es compatible con la mensajería de aplicación a aplicación (A2A):

* [Fanout to Kinesis Data Firehose delivery streams.](https://docs.aws.amazon.com/sns/latest/dg/sns-firehose-as-subscriber.html)
* [Fanout to Lambda functions.](https://docs.aws.amazon.com/sns/latest/dg/sns-lambda-as-subscriber.html)
* [Fanout to Amazon SQS queues.](https://docs.aws.amazon.com/sns/latest/dg/sns-sqs-as-subscriber.html)
* [Fanout to HTTP/S endpoints.](https://docs.aws.amazon.com/sns/latest/dg/sns-http-https-endpoint-as-subscriber.html)
* [Fanout to AWS Event Fork Pipelines.](https://docs.aws.amazon.com/sns/latest/dg/sns-fork-pipeline-as-subscriber.html)

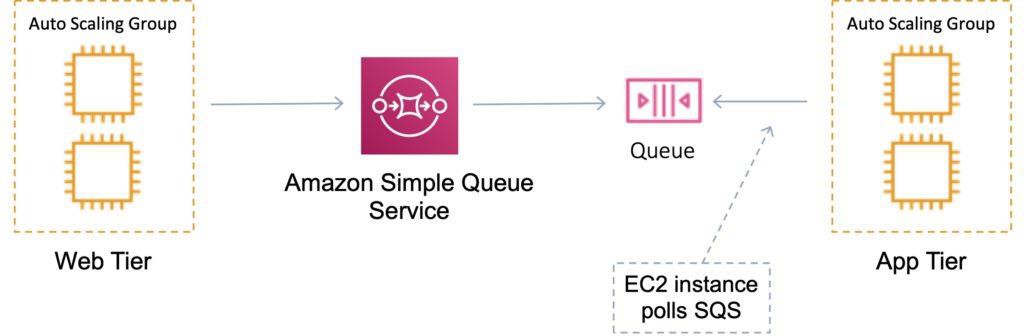
### AMAZON SQS

Amazon Simple Queue Service (SQS) es un sistema de colas distribuidas que permite a las aplicaciones de servicios web poner en cola de forma rápida y fiable los mensajes que genera un componente de la aplicación para ser consumidos por otro componente.

Amazon SQS le permite enviar, almacenar y recibir mensajes entre componentes de software. Una cola de Amazon SQS es un repositorio temporal para los mensajes que están en espera de procesamiento.

La cola SQS actúa como un búfer entre el componente que produce y guarda datos, y el componente que recibe los datos para su procesamiento.

La cola sqs resuelve los problemas que surgen si el productor está produciendo trabajo más rápido de lo que el consumidor puede procesarlo, o si el productor o consumidor solo están conectados intermitentemente a la red.



Esto se conoce como desacoplamiento / acoplamiento suelto y ayuda a habilitar la elasticidad para su aplicación. Amazon SQS se basa en pull, no en push (como Amazon SNS).

Los mensajes tienen un tamaño de hasta 256 KB.

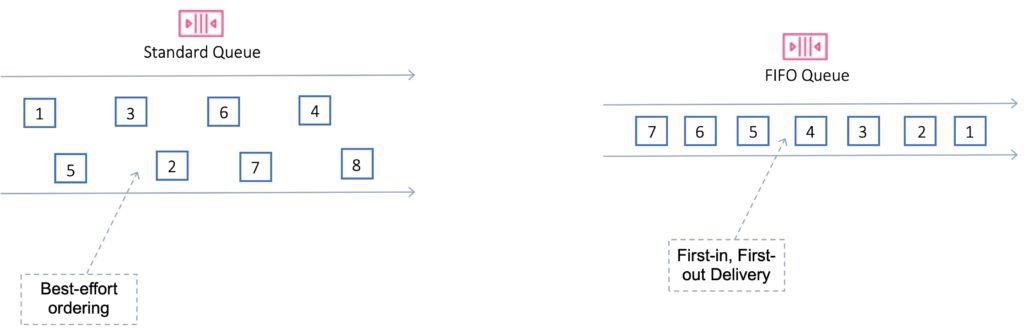
Los mensajes se pueden mantener en la cola de 1 minuto a 14 días. El período de retención predeterminado es de 4 días.

SQS garantiza que sus mensajes serán procesados al menos una vez.

#### SQS Queues

Los nombres de cola deben ser únicos dentro de una región.

Hay dos tipos de colas: colas estándar y colas FIFO.



#### Standard

Tipo de cola predeterminado.

Transacciones casi ilimitadas por segundo. Garantizar que un mensaje se entregue al menos una vez.

Ocasionalmente, más de una copia de un mensaje puede ser entregada fuera de servicio.

Proporciona pedidos de mejor esfuerzo que garantizan que los mensajes generalmente se entreguen en el mismo orden en que se envían.

#### First in First Out (FIFO)

Entrega exactamente una vez el procesamiento.

El orden en que se envían y reciben los mensajes se conserva estrictamente y un mensaje se entrega una vez y permanece disponible hasta que un consumidor lo procesa y elimina.

Los duplicados no se introducen en la cola.

Las colas FIFO también admiten grupos de mensajes que permiten varios grupos de mensajes ordenados dentro de una sola cola.

Limitado a 300 transacciones por segundo (TPS) pero tiene todas las capacidades de las colas estándar. Deduplicación con colas FIFO:

* Proporcionar un MessageDeduplicationId con el mensaje.
* El intervalo de deduplicación es de 5 minutos.
* Duplicación basada en contenido: messageDeduplicationId se genera como SHA-256 con el cuerpo del mensaje.

Secuenciación con colas FIFO:

* Para garantizar un orden estricto entre mensajes, especifique un MessageGroupId.
* Los mensajes con un ID de grupo diferente pueden recibirse fuera de servicio.
* Los mensajes con el mismo ID de grupo se entregan a un consumidor a la vez.

Las colas FIFO requieren que los parámetros ID de grupo de mensajes e ID de deduplicación de mensajes se agreguen a los mensajes.

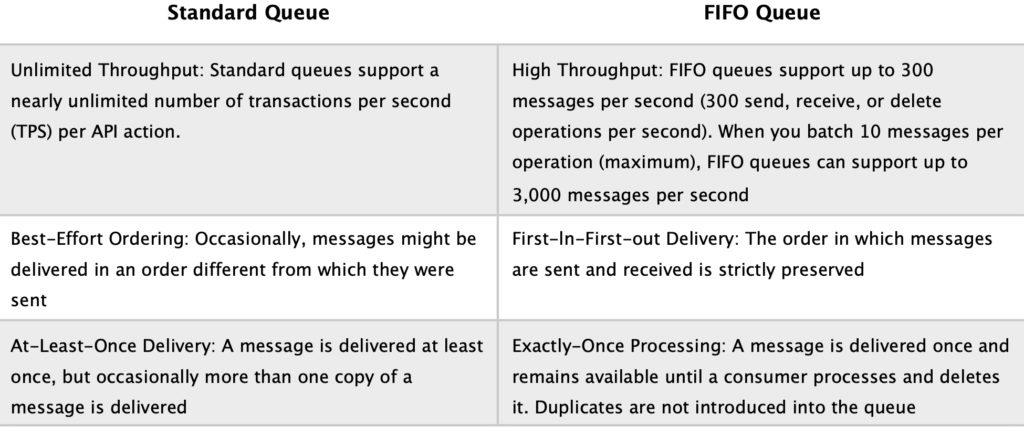
ID de grupo de mensajes:

* Etiqueta que especifica que un mensaje pertenece a un grupo de mensajes específico. Se garantiza que los mensajes que pertenecen al mismo grupo de mensajes se procesarán de manera FIFO.

ID de deduplicación de mensajes:

* El token utilizado para la deduplicación de mensajes dentro del intervalo de deduplicación.

En la tabla siguiente se proporciona una comparación en paralelo de las colas estándar y FIFO:



#### Scalability and Durability

Puede tener varias colas con diferentes prioridades. El escalado se realiza creando más colas.

SQS almacena todas las colas de mensajes y mensajes dentro de una única región de AWS de alta disponibilidad con varias zonas de disponibilidad redundantes.

#### Lambda with Amazon SQS

SQS se puede configurar como un origen de eventos para las funciones de AWS Lambda. Lambda con SQS:

* Puede utilizar una función de AWS Lambda para procesar mensajes en una cola de Amazon Simple Queue Service (Amazon SQS).
* Las asignaciones de orígenes de eventos de Lambda admiten colas estándar y colas de primero en entrar, primero en salir (FIFO).
* Con Amazon SQS, puede descargar tareas de un componente de la aplicación enviándolas a una cola y procesándolas de forma asincrónica.
* Lambda sondea la cola e invoca la función de Lambda de forma sincrónica con un evento que contiene mensajes de cola.
* Lambda lee mensajes en lotes e invoca la función una vez para cada lote.
* Cuando la función procesa correctamente un lote, Lambda elimina sus mensajes de la cola.

#### Monitoring

CloudWatch está integrado con SQS y puede ver y supervisar las métricas de la cola. Las métricas de CloudWatch se recopilan automáticamente cada 5 minutos.

CloudWatch considera que una cola está activa durante un máximo de 6 horas si contiene algún mensaje o si alguna acción de la API accede a ella.

Sin cargo por CloudWatch (sin monitoreo detallado).

CloudTrail captura llamadas a la API desde SQS y registra en un bucket de S3 especificado.

#### Security

Puede utilizar políticas de IAM para controlar quién puede leer y escribir mensajes.

La autenticación se puede utilizar para proteger los mensajes dentro de las colas (quién puede enviar y recibir). SQS cumple con PCI DSS nivel 1 y es elegible para HIPAA.

La seguridad a bordo se proporciona mediante HTTPS.

Puede habilitar el cifrado del lado del servidor (SSE) mediante KMS.

* Puede establecer la CMK que desea usar.
* Puede establecer el período de reutilización de la clave de datos.
* SSE sólo cifra el cuerpo del mensaje, no los atributos del mensaje.

IAM policy must allow usage of SQS.

También puede especificar permisos en una directiva de acceso a colas sqs:

* Proveedores de un control de grano más fino.
* Control sobre las solicitudes que llegan.

### AMAZON SWF

Amazon Simple Workflow Service (SWF) es un servicio web que facilita la coordinación del trabajo entre componentes de aplicaciones distribuidas.

Amazon SWF se utiliza para procesar trabajos en segundo plano que tienen pasos paralelos o secuenciales. Puede pensar en Amazon SWF como un rastreador de estado totalmente administrado y coordinador de tareas.

Use Amazon SWF si los pasos de su aplicación tardan más de 500 milisegundos en completarse, debe

realizar un seguimiento del estado de procesamiento, o debe recuperar o volver a intentarlo si se produce un error en una tarea. Con SWF puede crear sistemas asincrónicos distribuidos como flujos de trabajo. Realiza un seguimiento del estado de su flujo de trabajo que interactúa y actualiza a través de la API.

Más adecuado para flujos de trabajo habilitados por humanos, como un sistema de cumplimiento de pedidos o para solicitudes de procedimiento.

AWS recomienda que, para las nuevas aplicaciones, los clientes consideren AWS Step Functions en lugar de SWF.

SWF habilita aplicaciones para una variedad de casos de uso, incluido el procesamiento de medios, back-ends de aplicaciones web, flujos de trabajo de procesos de negocio y canalizaciones de análisis.

El registro es un paso único que se realiza para cada tipo diferente de flujo de trabajo y actividad. SWF tiene un tiempo de finalización de hasta 1 año para las ejecuciones de flujo de trabajo.

SWF utiliza una API orientada a tareas.

SWF garantiza que una tarea se asigne una vez y nunca se duplique. SWF realiza un seguimiento de todas las tareas y eventos de una aplicación.

Un dominio es un contenedor lógico para recursos de aplicación como flujos de trabajo, actividades y ejecuciones.

Los trabajadores son programas que interactúan con Amazon SWF para obtener tareas, procesar tareas recibidas y devolver los resultados.

El decisor es un programa que controla la coordinación de las tareas, es decir, su ordenación, concurrencia y programación de acuerdo con la lógica de la aplicación.

Las aplicaciones SWF incluyen los siguientes componentes lógicos:

* Domains.
* Workflows.
* Activities.
* Task Lists.
* Workers.
* Workflow Execution.

### AMAZON MQ

Amazon MQ es un servicio de agente de mensajes administrado para ActiveMQ.

Amazon MQ admite API y protocolos estándar del sector para que pueda migrar mensajes y aplicaciones sin necesidad de reescribir el código.

Amazon MQ proporciona una capacidad de mensajería rentable y flexible: usted paga por la instancia del agente y el uso de almacenamiento a medida que avanza.

Amazon MQ administra la administración y el mantenimiento de los agentes ActiveMQ y aprovisiona automáticamente la infraestructura para una alta disponibilidad.

Con Amazon MQ, puede utilizar la consola de administración de AWS, AWS CloudFormation, la interfaz de línea de comandos (CLI) o llamadas simples a la API para lanzar un agente de mensajes listo para producción en cuestión de minutos.

Amazon MQ está totalmente administrado y es de alta disponibilidad dentro de una región.

Amazon MQ almacena sus mensajes de forma redundante en varias zonas de disponibilidad (AZ). Los brokers activos/en espera están diseñados para una alta disponibilidad.

En caso de que se produzca un error del agente, o incluso una interrupción completa de AZ, Amazon MQ conmuta automáticamente por error al agente en espera para que pueda seguir enviando y recibiendo mensajes.

API de ActiveMQ y compatibilidad con JMS, NMS, MQTT y WebSockets.

Está diseñado como un reemplazo directo para los agentes de mensajes locales.

Use SQS si está creando una nueva aplicación desde cero.

Utilice MQ si desea una ruta fácil y sencilla para migrar de los agentes de mensajes existentes a AWS. Amazon MQ proporciona cifrado de sus mensajes en reposo y en tránsito.

Es fácil asegurarse de que sus mensajes se almacenan de forma segura en un formato cifrado. Las conexiones al agente utilizan SSL y el acceso se puede restringir a un punto de enlace privado dentro de su Amazon VPC, lo que le permite aislar a su agente en su propia red virtual.

Puede configurar grupos de seguridad para controlar el acceso de red al intermediario.

Amazon MQ está integrado con Amazon CloudWatch y AWS CloudTrail. Con CloudWatch puede supervisar las métricas de sus agentes, colas y temas.

### AWS STEP FUNCTIONS

AWS Step Functions se puede utilizar para coordinar los componentes de las aplicaciones distribuidas como una serie de pasos en un flujo de trabajo visual.

Puede crear y ejecutar rápidamente máquinas de estado para ejecutar los pasos de su aplicación de una manera confiable y escalable.

Cómo funciona:

1. Defina los pasos de su flujo de trabajo en el lenguaje de Amazon States basado en JSON. La consola visual grafica automáticamente cada paso en el orden de ejecución.
2. Inicie una ejecución para visualizar y verificar que los pasos de la aplicación funcionan según lo previsto. La consola destaca el estado en tiempo real de cada paso y proporciona un historial detallado de cada ejecución.
3. AWS Step Functions opera y escala los pasos de su aplicación y la computación subyacente para que pueda ayudarlo a garantizar que su aplicación se ejecute de manera confiable bajo una demanda cada vez mayor.

Es una plataforma de flujo de trabajo y orquestación administrada.

Es escalable y de alta disponibilidad.

La aplicación se define como un equipo de estado.

Cree tareas, pasos secuenciales, pasos paralelos, rutas de bifurcación o temporizadores.

Utiliza JSON declarativo de lenguaje de estado de Amazon.

Las aplicaciones pueden interactuar y actualizar la transmisión a través de la API de función de paso. Proporciona una interfaz visual que describe el flujo y el estado en tiempo real. Proporciona registros detallados de cada paso de ejecución.

Características de AWS Step Functions:

* **Gestión de errores integrada:** AWS Step Functions realiza un seguimiento del estado de cada paso, por lo que puede volver a intentar automáticamente tareas fallidas o con un tiempo de espera, detectar errores específicos y recuperarse correctamente, ya sea que la tarea tarde segundos o meses en completarse.
* **Escalado automático:** AWS Step Functions escala automáticamente las operaciones y el cómputo subyacente para ejecutar los pasos de su aplicación por usted en respuesta a las cargas de trabajo cambiantes. Step Functions se escala automáticamente para ayudar a garantizar que el rendimiento del flujo de trabajo de la aplicación se mantenga constantemente alto a medida que aumenta la frecuencia de las solicitudes.
* **Pago por uso:** con AWS Step Functions, solo paga por la transición de un paso del flujo de trabajo de la aplicación al siguiente, lo que se denomina transición de estado. La facturación se mide por transición estatal, independientemente de cuánto tiempo persista cada estado (hasta un año).
* **Historial de eventos de ejecución:** AWS Step Functions crea un registro de eventos detallado para cada ejecución, de modo que cuando las cosas salen mal, puede identificar rápidamente no solo dónde, sino también por qué. Todo el historial de ejecución está disponible visual y programáticamente para solucionar y corregir rápidamente los errores.
* **Alta disponibilidad:** AWS Step Functions tiene tolerancia a errores integrada. Step Functions mantiene la capacidad de servicio en varias zonas de disponibilidad de cada región para ayudar a proteger los flujos de trabajo de las aplicaciones contra fallos individuales de las instalaciones de la máquina o del centro de datos. No hay ventanas de mantenimiento ni tiempos de inactividad programados.
* **Seguridad administrativa:** AWS Step Functions está integrado con AWS Identity and Access Management (IAM). Las políticas de IAM se pueden utilizar para controlar el acceso a las API de Step Functions.

## APPLICATION INTEGRATION QUIZ QUESTIONS

Answers and explanations are provided below after the last question in this section.

###### Question 1:

There is expected to be a large increase in write intensive traffic to a website you manage that registers users onto an online learning program. You are concerned about writes to the database being dropped and need to come up with a solution to ensure this does not happen. Which of the solution options below would be the best approach to take?

1. Update the application to write data to an SQS queue and provision additional EC2 instances to process the data and write it to the database
2. Use RDS in a multi-AZ configuration to distribute writes across AZs
3. Update the application to write data to an S3 bucket and provision additional EC2 instances to process the data and write it to the database
4. Use CloudFront to cache the writes and configure the database as a custom origin

###### Question 2:

You are using a series of Spot instances that process messages from an SQS queue and store results in a DynamoDB table. Shortly after picking up a message from the queue AWS terminated the Spot instance. The Spot instance had not finished processing the message. What will happen to the message?

1. The message will be lost as it would have been deleted from the queue when processed
2. The message will remain in the queue and be immediately picked up by another instance
3. The message will become available for processing again after the visibility timeout expires
4. The results may be duplicated in DynamoDB as the message will likely be processed multiple times

###### Question 3:

You are developing a multi-tier application that includes loosely-coupled, distributed application components and need to determine a method of sending notifications instantaneously. Using SNS which transport protocols are supported? (choose 2)

1. FTP
2. Email-JSON
3. HTTPS
4. Amazon SWF
5. AWS Lambda

###### Question 4:

A Solutions Architect is creating the business process workflows associated with an order fulfilment system. What AWS service can assist with coordinating tasks across distributed application components?

1. Amazon STS
2. Amazon SQS
3. Amazon SWF
4. Amazon SNS

###### Question 5:

You are a developer at Digital Cloud Training. An application stack you are building needs a message bus to decouple the application components from each other. The application will generate up to 300 messages per second without using batching. You need to ensure that a

message is only delivered once, and duplicates are not introduced into the queue. It is not necessary to maintain the order of the messages.

Which SQS queue type will you use?

1. Standard queues
2. Long polling queues
3. Auto Scaling queues
4. FIFO queues

###### Question 6:

A client is in the design phase of developing an application that will process orders for their online ticketing system. The application will use a number of front-end EC2 instances that pick-up orders and place them in a queue for processing by another set of back-end EC2 instances. The client will have multiple options for customers to choose the level of service they want to pay for.

The client has asked how he can design the application to process the orders in a prioritized way based on the level of service the customer has chosen?

1. Create multiple SQS queues, configure the front-end application to place orders onto a specific queue based on the level of service requested and configure the back-end instances to sequentially poll the queues in order of priority
2. Create a combination of FIFO queues and Standard queues and configure the applications to place messages into the relevant queue based on priority
3. Create a single SQS queue, configure the front-end application to place orders on the queue in order of priority and configure the back-end instances to poll the queue and pick up messages in the order they are presented
4. Create multiple SQS queues, configure exactly-once processing and set the maximum visibility timeout to 12 hours

### APPLICATION INTEGRATION - ANSWERS

###### Question 1 answer: A

Explanation:

This is a great use case for Amazon Simple Queue Service (Amazon SQS). SQS is a web service that gives you access to message queues that store messages waiting to be processed and offers a reliable, highly-scalable, hosted queue for storing messages in transit between computers.

SQS is used for distributed/decoupled applications. In this circumstance SQS will reduce the risk of writes being dropped and it the best option presented.

RDS in a multi-AZ configuration will not help as writes are only made to the primary database.

Though writing data to an S3 bucket could potentially work, it is not the best option as SQS is recommended for decoupling application components.

The CloudFront option is bogus as you cannot configure a database as a custom origin in CloudFront.

###### Question 2 answer: C

Explanation:

The visibility timeout is the amount of time a message is invisible in the queue after a reader picks up the message. If a job is processed within the visibility timeout the message will be deleted. If a job is not processed within the visibility timeout the message will become visible again (could be delivered twice). The maximum visibility timeout for an Amazon SQS message is 12 hours.

The message will not be lost and will not be immediately picked up by another instance. As mentioned above it will be available for processing in the queue again after the timeout expires.

As the instance had not finished processing the message it should only be fully processed once. Depending on your application process however it is possible some data was written to DynamoDB.

###### Question 3 answer: B,C

Explanation:

Note that the questions asks you which transport protocols are supported, NOT which subscribers - therefore Lambda is not supported

SNS supports notifications over multiple transport protocols:

* HTTP/HTTPS – subscribers specify a URL as part of the subscription registration
* Email/Email-JSON – messages are sent to registered addresses as email (text-based or JSON-object)
* SQS – users can specify an SQS standard queue as the endpoint
* SMS – messages are sent to registered phone numbers as SMS text messages

###### Question 4 answer: C

Explanation:

Amazon Simple Workflow Service (SWF) is a web service that makes it easy to coordinate work across distributed application components. SWF enables applications for a range of use cases, including media processing, web application back-ends, business process workflows, and analytics pipelines, to be designed as a coordination of tasks.

Amazon Security Token Service (STS) is used for requesting temporary credentials.

Amazon Simple Queue Service (SQS) is a message queue used for decoupling application components.

Amazon Simple Notification Service (SNS) is a web service that makes it easy to set up, operate, and send notifications from the cloud.

SNS supports notifications over multiple transports including HTTP/HTTPS, Email/Email-JSON, SQS and SMS.

###### Question 5 answer: D

Explanation:

The key fact you need to consider here is that duplicate messages cannot be introduced into the queue. For this reason alone you must use a FIFO queue. The statement about it not being necessary to maintain the order of the messages is meant to confuse you, as that might lead you to think you can use a standard queue, but standard queues don't guarantee that duplicates are not introduced into the queue.

FIFO (first-in-first-out) queues preserve the exact order in which messages are sent and received

– note that this is not required in the question but exactly once processing is. FIFO queues provide exactly-once processing, which means that each message is delivered once and remains available until a consumer processes it and deletes it.

Standard queues provide a loose-FIFO capability that attempts to preserve the order of messages. Standard queues provide at-least-once delivery, which means that each message is delivered at least once.

Long polling is configuration you can apply to a queue, it is not a queue type. There is no such thing as an Auto Scaling queue.

###### Question 6 answer: A

Explanation:

The best option is to create multiple queues and configure the application to place orders onto a specific queue based on the level of service. You then configure the back-end instances to poll these queues in order or priority, so they pick up the higher priority jobs first.

Creating a combination of FIFO and standard queues is incorrect as creating a mixture of queue types is not the best way to separate the messages, and there is nothing in this option that explains how the messages would be picked up in the right order.

Creating a single queue and configuring the applications to place orders on the queue in order of priority would not work as standard queues offer best-effort ordering so there’s no guarantee that the messages would be picked up in the correct order.

Creating multiple SQS queues and configuring exactly-once processing (only possible with FIFO) would not ensure that the order of the messages is prioritized.