GUÍA DE ESTUDIO TÉCNICA E INTENSIVA



AWS Certified Solutions Architect – Associate (SAA-C03)

Guía de Estudio Técnica Consolidada: 4 Dominios, Matrices de Decisión y Preguntas Trampa.

Dominio 1: Diseño de Arquitecturas Seguras (30%)

Dominio 2: Diseño de Arquitecturas Resilientes (26%)

Dominio 3: Diseño de Arquitecturas de Alto Rendimiento (24%)

Dominio 4: Diseño de Arquitecturas Optimizadas en Costos (20%)

Compilado por: David Nicanor

Versión: Edición SAA-C03 (2026)

Dominio 1 - Diseño de Arquitecturas Seguras (AWS SAA-C03)..... 5
1. Gestión de Identidades y Accesos (IAM) y Control de Seguridad Avanzado..... 5
1.1. Fundamentos e Infraestructura de IAM: Roles, Políticas y la Regla del Involucramiento de Recurso.....5
1.2. Asignación de Permisos en Cargas de Trabajo: ECS Task Roles vs Launch Configurations / Launch Templates..... 7
1.3. Criptografía y Gestión de Claves: AWS KMS, SSE-KMS y Claves Administradas. 8
1.4. Gestión de Configuración y Secretos: Systems Manager Parameter Store vs AWS Secrets Manager..... 10
2. Seguridad en Redes de la VPC e Interconexión Privada..... 11
2.1. Firewalls Virtuales y Control de Tráfico: Security Groups (Stateful) vs NACLs (Stateless).....11
2.2. Acceso Remoto Seguro: Bastion Hosts e Integración con Rangos IP Corporativos..... 13
Pasos para Diseñar un Bastion Host Seguro (Patrón Tradicional de Examen)... 13
2.3. Acceso Privado a Servicios de AWS: VPC Gateway Endpoints vs VPC Interface Endpoints (AWS PrivateLink)..... 14
2.4. Conectividad Segura Híbrida: AWS Site-to-Site VPN..... 15
3. Gobernanza, Protección de Aplicaciones y Gestión Centralizada de Seguridad.....16
3.1. Protección Perimetral y Mitigación de Ataques: AWS WAF y AWS Shield..... 16
3.2. Autenticación y Control de Usuarios: Amazon Cognito User Pools y Gobernanza de Identidad de Aplicación..... 17
"So What?" para el Examen: Cognito User Pools vs Cognito Identity Pools..... 18
3.3. Gobernanza, Auditoría y Remediación Automática: AWS Config Conformance Packs y AWS Organizations.....18
4. Matriz Estratégica para el Examen SAA-C03: Lógica de Decisión, Palabras Clave y Trampas Frecuentes..... 19
4.1. Lógica de Decisión Rápida.....19
4.2. Glosario de Palabras Clave y Desglose de Trampas Habituales del Examen Real..... 20
Catálogo de Palabras Clave de Activación (Trigger Words)..... 20
Las 5 Trampas Más Frecuentes del Dominio 1 en el Examen SAA-C03.....21
Dominio 2 - Diseño de Arquitecturas Resilientes (AWS SAA-C03)..... 22
1. Introducción Estratégica al Dominio 2 (26% del Examen SAA-C03)..... 22
2. Alta Disponibilidad y Escalabilidad Elástica en Cómputo (EC2, ALB y Auto Scaling). 23
2.1 Application Load Balancer (ALB) y Balanceo Multizona..... 23
2.2 Auto Scaling Groups (ASG) y Lógica Interna de Terminación..... 24
2.3 Gestión de Service Quotas y Cuellos de Botella de Aprovisionamiento..... 26
2.4 Tabla Comparativa y Trampas del Examen.....26
3. Arquitecturas Desacopladas y Orientadas a Eventos (Event-Driven Architecture).....27
3.1 Amazon SQS (Standard vs. FIFO) y Manejo de Errores..... 28
3.2 Amazon SNS y Patrones Fanout..... 29
3.3 Ingestión en Tiempo Real: Amazon EventBridge vs. Amazon Kinesis Data Streams..... 30
3.4 Resumen Estructurado de Decisión.....30

4. Bases de Datos Resilientes y Alta Disponibilidad de Datos.....30
4.1 Amazon RDS Multi-AZ vs. Read Replicas..... 31
4.2 Amazon Aurora y Aurora Global Database..... 32
4.3 Amazon DynamoDB y DynamoDB Global Tables..... 33
4.4 Matriz de Decisión de Almacenamiento Persistente..... 33
5. Estrategias Avanzadas de Disaster Recovery (DR) y Continuidad de Negocio..... 34
5.1 Evaluación Comparativa de las 4 Estrategias Tradicionales de DR..... 34
5.2 Route 53 Failover Routing y Resguardo Estático en S3..... 35
5.3 AWS Backup y Replicación Cross-Region..... 36
6. Lógica de Decisión, Palabras Clave y Trampas Típicas del Examen Real.....36
6.1 Mapeo de Patrones de Arquitectura Resiliente (Cheat Sheet de Decisión).....36
6.2 Trampas Frecuentes y Distractores en el Dominio 2..... 37
6.3 Lista de Chequeo de Autoevaluación.....38
Dominio 3 - Diseñar Arquitecturas de Alto Rendimiento (AWS SAA-C03)..... 39
1. Marco Estratégico del Dominio 3 y Lógica General de Resolución..... 39
Contexto Estratégico..... 39
Desglose del Método de Razonamiento.....39
Diferenciación e Integración de Métricas Clave de Rendimiento..... 39
2. Almacenamiento de Alto Rendimiento: EBS, EFS, Instance Store y S3.....40
Contexto Estratégico..... 40
2.1 Amazon EBS: Comparativa de Tipos y Escalado Dinámico vía Elastic Volumes.....40
Mecanismo de Escalado Dinámico: EBS Elastic Volumes y la API ModifyVolume.. 42
2.2 Amazon EFS vs. EC2 Instance Store para Sistemas Masivos..... 42
2.3 Amazon S3 Standard vs. S3 Transfer Acceleration..... 43
3. Cómputo Escalable y Estrategias de Caché: Lambda, ElastiCache, DAX y Athena. 44
Contexto Estratégico..... 44
3.1 Optimización de Cómputo Serverless con AWS Lambda..... 44
3.2 Estrategias de Caché de Base de Datos: Amazon ElastiCache (Lazy Loading vs. Write-Through).....45
3.3 Lecturas de Ultra-Baja Latencia con DynamoDB Accelerator (DAX).....46
3.4 Analítica de Alto Rendimiento en S3: Amazon Athena con Parquet y Particionamiento.....46
Estrategias de Optimización para Examen:..... 47
4. Aceleración Global e Intercepción de Tráfico: AWS Global Accelerator vs. Amazon CloudFront.....47
Contexto Estratégico..... 47
4.1 Comparativa Arquitectónica: Global Accelerator vs. CloudFront..... 47
4.2 Seguridad y Control de Acceso en Contenido Distribuido (CloudFront OAC y Signed URLs/Cookies)..... 48
Asegurar el Origen S3 con Origin Access Control (OAC)..... 48
Control de Acceso Restringido: Signed URLs vs. Signed Cookies.....49
5. Matriz Sintética de Decisiones, Palabras Clave y Trampas Frecuentes del Examen 49

Contexto Estratégico..... 49
5.1 Matriz de Decisión Instantánea por Escenario..... 49
5.2 Trampas Comunes y Distractores en el Examen SAA-C03..... 51
Conclusión y Mentalidad de Arquitecto..... 52
Dominio 4 - Diseño de Arquitecturas Optimizadas en Costos (AWS SAA-C03)..... 52
1. Visión General Estratégica del Dominio 4 (20% del Examen SAA-C03)..... 52
Estructura del Examen y Metodología de Puntuación..... 52
Marco de Razonamiento Arquitectónico..... 53
Walkthrough Práctico del Marco de Razonamiento..... 53
2. Modelos y Estrategias de Cómputo Costo-Efectivos..... 54
Evaluación Comparativa de Modelos de Compra de EC2..... 54
Estrategia de Spot Instances con Auto Scaling Target Tracking..... 55
Cómputo Serverless y Abstracción de Infraestructura..... 56
3. Optimización Avanzada del Almacenamiento (Amazon S3 y EBS)..... 57
Estrategias en Amazon S3 y Análisis Comparativo de Clases..... 57
Configuración Estratégica de S3 Lifecycle..... 58
Análisis de Costos Ocultos y Optimización en Amazon EBS..... 58
La Trampa de Facturación Oculta en Amazon EC2 y EBS..... 58
Migración de EBS io1/io2 a gp3 mediante CloudWatch..... 58
Visión Comparativa de Almacenamiento Complementario..... 59
4. Arquitectura de Bases de Datos Costo-Eficientes..... 59
Amazon Aurora Serverless (v1 vs. v2) para Cargas Intermitentes..... 59
Comparativa y Selección de Modelos en Relacionales y NoSQL..... 60
Amazon DynamoDB: Provisioned Capacity vs. On-Demand..... 60
RDS Multi-AZ, Read Replicas y ElastiCache..... 61
5. Herramientas de Gobernanza, Control y Análisis de Gastos..... 61
Desglose Analítico de Servicios de Control Financiero..... 61
6. Lógica de Decisión, Palabras Clave y Trampas Comunes en el Examen Real..... 62
Tabla de Mapeo: Palabras Clave vs. Solución Arquitectónica Optimizada..... 62
Análisis de Trampas Comunes y Distractores en el Examen..... 63
Resumen Final de Reglas de Oro para el Examen..... 64

Dominio 1 - Diseño de Arquitecturas Seguras (AWS SAA-C03)

1. Gestión de Identidades y Accesos (IAM) y Control de Seguridad Avanzado

1.1. Fundamentos e Infraestructura de IAM: Roles, Políticas y la Regla del Involucramiento de Recurso

La gestión de identidades y accesos dentro de Amazon Web Services (AWS) constituye la piedra angular de cualquier arquitectura en la nube. En el marco de AWS Well-Architected Framework (específicamente dentro del pilar de Seguridad), la identidad actúa como el primer y principal perímetro de defensa operacional. Una configuración rigurosa de control de acceso previene la exposición no autorizada de infraestructura crítica y garantiza que únicamente las entidades autenticadas y explícitamente autorizadas puedan interactuar con los recursos del sistema bajo el principio de menor privilegio (*Least Privilege*).

Para estructurar los permisos adecuadamente, es crucial distinguir las diferencias operativas entre las **políticas basadas en identidad** (*identity-based policies*) y las **políticas basadas en recursos** (*resource-based policies*). Las políticas basadas en identidad se adjuntan directamente a usuarios, grupos o roles de IAM para definir qué acciones puede realizar dicha entidad y sobre qué recursos. Por el contrario, las políticas basadas en recursos se adjuntan directamente al recurso final (tales como buckets de Amazon S3, temas de Amazon SNS, colas de Amazon SQS o claves de AWS KMS). La diferencia fundamental radica en la presencia del elemento Principal: **en las políticas basadas en recursos, el elemento Principal es estrictamente obligatorio** , ya que el recurso debe definir de manera explícita qué cuenta, usuario, rol o servicio tiene permitido o denegado el acceso. En las políticas basadas en identidad, el elemento Principal no existe ni está permitido en la estructura JSON, dado que el sujeto que ejecuta la acción está implícito por la propia entidad a la que la política está adjuntada.

```
// Política Basada en Recursos (S3 Bucket Policy) - 'Principal' es
OBLIGATORIO
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCrossAccountRead",
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::123456789012:root" },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::mi-bucket-seguro/*"
    }
  ]
}
```

En la lógica de evaluación de políticas de IAM, AWS aplica una regla categórica: **una denegación explícita (*explicit deny*) sobreescribe cualquier permiso explícito (*allow*)** . Por defecto, todo acceso en AWS está denegado de forma implícita (*implicit deny*). Para que una petición sea autorizada, la evaluación de políticas debe encontrar al menos

una declaración de permiso explícito (*allow*) y **ninguna** declaración de denegación explícita (*deny*) en ninguna de las políticas aplicables (incluyendo políticas de identidad, de recursos, límites de permisos y Políticas de Control de Servicios u Organizational SCPs). Asimismo, para delegar accesos temporales sin exponer credenciales estáticas de largo plazo, AWS Security Token Service (STS) emite credenciales temporales compuestas por un *Access Key ID* , un *Secret Access Key* y un *Session Token* con un periodo de caducidad programado.

Tipo de Política	Dónde se Adjunta	Presencia del Elemento `Principal`	Caso de Uso Principal en el Examen
Basada en Identidad (*Identity-based*)	Usuarios, Grupos e IAM Roles	**No** (el sujeto está implícito)	Conceder permisos de acceso granulares a usuarios, roles o servicios sobre recursos de AWS.
Basada en Recursos (*Resource-based*)	Recursos (Amazon S3, SNS, SQS, KMS, ECR)	**Sí** (estrictamente obligatorio)	Autorizar acceso cruzado entre cuentas (*cross-account*) o acceso directo en el recurso sin asumir roles.
Límite de Permisos (*Permissions Boundary*)	Usuarios o Roles de IAM	**No**	Establecer el techo máximo de permisos que una entidad de IAM puede recibir, evitando la elevación de privilegios.
Política de Control de Servicios (*SCP*)	Unidades Organizativas (OU) o Cuentas en AWS Organizations	**No**	Restringir de forma centralizada las acciones máximas permitidas para todas las cuentas de una organización.

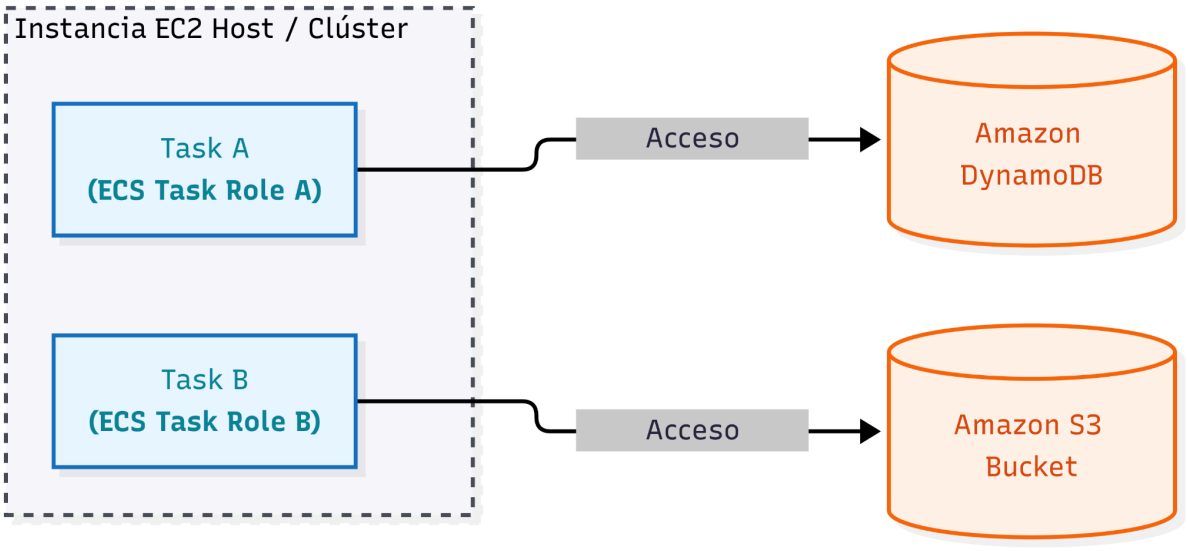
Comprender la separación de permisos a nivel de identidades y recursos sienta las bases para abordar cómo estas credenciales deben inyectarse de forma dinámica y aislada en entornos de ejecución de cómputo y contenedores.

1.2. Asignación de Permisos en Cargas de Trabajo: ECS Task Roles vs Launch Configurations / Launch Templates

El principio de menor privilegio (*Least Privilege*) exige que cada componente de una solución disponga estrictamente de los permisos mínimos requeridos para ejecutar su función explícita. En arquitecturas modernas orientadas a microservicios y contenedores, la asignación de permisos debe realizarse con granularidad individualizada a nivel de proceso o tarea, evitando que el compromiso de un servicio secundario comprometa toda la plataforma.

Al desplegar aplicaciones en Amazon Elastic Container Service (ECS), es un error arquitectónico habitual otorgar permisos de IAM a la instancia host EC2 subyacente a través del perfil de la instancia (*EC2 Instance Profile Role* definido en el *Launch Template* o *Launch Configuration*). Si los permisos de acceso a servicios como DynamoDB o S3 se asignan al nivel de la instancia host, **todos los contenedores y tareas que coexistan en ese nodo heredarán idénticas credenciales de AWS** , eliminando el aislamiento entre aplicaciones. La solución arquitectónica correcta es implementar **ECS Task Roles** .

Los *ECS Task Roles* permiten asociar un rol de IAM específico a cada definición de tarea (*Task Definition*), logrando que cada contenedor reciba sus propias credenciales temporales otorgadas dinámicamente por AWS STS a través de la interfaz de metadatos del contenedor, de manera totalmente independiente del host.



El "So What?" arquitectónico es crítico para la mitigación de riesgos: si un atacante logra explotar una vulnerabilidad de ejecución remota de código en el contenedor que ejecuta la Tarea A, únicamente obtendrá los permisos acotados del *Task Role A* . No podrá acceder a los buckets de S3 asignados a la Tarea B ni a los recursos del nodo EC2 host, garantizando el aislamiento criptográfico y lógico entre inquilinos (*multi-tenant isolation*).

- **Trampa Típica 1:** Respuestas que proponen añadir permisos de lectura/escritura a nivel del rol de la instancia EC2 (*EC2 Instance Profile Role*) para resolver fallos de

acceso en contenedores ECS. **Incorrecto.** La respuesta correcta debe emplear un *ECS Task Role* .

- **Trampa Típica 2:** Confundir el *ECS Task Execution Role* con el *ECS Task Role* . El *Task Execution Role* lo utiliza el agente de ECS para consultar Amazon ECR (descargar imágenes) y enviar logs a Amazon CloudWatch. El *Task Role* lo utiliza el código de la aplicación dentro del contenedor para interactuar con las APIs de AWS.

- **Trampa Típica 3:** Asumir que AWS Fargate requiere gestionar roles de instancia EC2. En Fargate la infraestructura es serverless; por lo tanto, no existen instancias EC2 gestionadas por el usuario y el uso de *Task Roles* es el único mecanismo para dotar de identidad IAM a los contenedores.

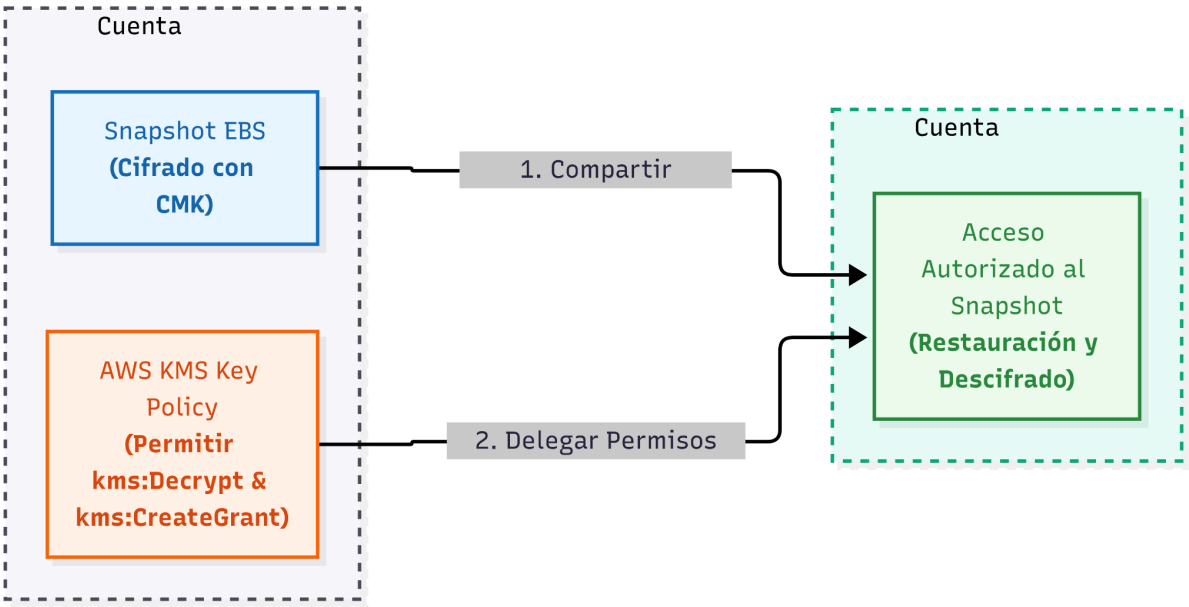
Garantizar la asignación rigurosa de permisos en el plano de cómputo protege las operaciones activas, pero requiere complementarse con la protección criptográfica de los datos en reposo procesados por dichas cargas de trabajo.

1.3. Criptografía y Gestión de Claves: AWS KMS, SSE-KMS y Claves Administradas

El cifrado en reposo y en tránsito es un requerimiento imprescindible para cumplir con los estándares normativos de la industria (tales como PCI-DSS, HIPAA y SOC 2) y garantizar la confidencialidad de los datos. AWS Key Management Service (KMS) proporciona una infraestructura totalmente gestionada basada en módulos de seguridad de hardware (HSM) para la creación y control de claves criptográficas.
Existen dos categorías principales de Claves KMS (KMS Keys) dentro del servicio:

1. **Claves Administradas por AWS** (*AWS-managed keys*): Creadas automáticamente por los servicios de AWS (como S3, EBS o RDS) al activar el cifrado. Poseen el formato de nombre aws/servicio (por ejemplo, aws/ebs). Son gratuitas en términos de mantenimiento de clave, pero su política de clave (*Key Policy*) no puede ser modificada por el usuario ni se pueden compartir entre distintas cuentas de AWS. Su rotación se realiza automáticamente cada año por parte de AWS.
2. **Claves Administradas por el Cliente** (*Customer-managed keys* - CMK): Creadas y gestionadas explícitamente por el cliente. Permiten el control total sobre la política de clave (*Key Policy*), la activación de rotación automática anual configurable, la definición de alias y la **delegación de acceso cruzado entre cuentas de AWS (*cross-account sharing*)** .

Conforme a las reglas técnicas del examen, **cuando se comparte un recurso cifrado (como un *snapshot* de EBS o una imagen AMI) con otra cuenta de AWS, es estrictamente obligatorio utilizar una Clave Administrada por el Cliente (CMK)** . Las claves administradas por AWS (aws/ebs) no pueden ser compartidas entre cuentas bajo ninguna circunstancia. Para compartir el *snapshot* , se debe autorizar a la cuenta destino dentro de la política de la clave CMK y posteriormente compartir el *snapshot* .



El Cifrado del Lado del Servidor con KMS (SSE-KMS) aplica el patrón de **Cifrado de Sobre** (*Envelope Encryption*). KMS utiliza la clave KMS (*KMS Key*) para generar una Clave de Cifrado de Datos (*Data Encryption Key* - DEK) en texto plano y una versión cifrada de la misma. El servicio de destino (ej. Amazon S3 o Amazon EBS) utiliza la DEK en texto plano para cifrar el volumen o datos localmente y luego descarta la DEK no cifrada de la memoria, conservando únicamente la DEK cifrada junto con los datos. Esto evita enviar volúmenes masivos de datos a través de la API de KMS, reduciendo la latencia y maximizando el rendimiento.
En el almacenamiento de bloques, los volúmenes de Amazon EBS ofrecen diferentes tipos optimizados por rendimiento que deben combinarse con cifrado KMS persistiéndose en todo el ciclo de vida:

- **SSD General Purpose (gp2 / gp3)** : Diseñados para cargas transaccionales, arranque de SO y entornos de prueba/desarrollo.
- **Provisioned IOPS SSD (io1 / io2)** : Diseñados para bases de datos de ultra baja latencia y E/S crítica (NoSQL, MySQL, RDS).
- **Throughput Optimized HDD (st1)** : Almacenamiento magnético de bajo costo para cargas de lectura secuencial e intensiva (Big Data, Data Warehousing).
- **Cold HDD (sc1)** : El menor costo en disco HDD para datos de acceso poco frecuente.

Regla de Persistencia de Cifrado EBS : Cuando se crea un volumen EBS cifrado, todos los *snapshots* derivados de él, así como las AMIs creadas a partir de dichos *snapshots* y los nuevos volúmenes EBS restaurados a partir de esas AMIs, **permanecen cifrados por defecto de manera automática** .

Mecanismo de Cifrado	Control de Rotación	Costo / Complejidad	Impacto en el Examen SAA-C03
SSE-S3	Gestionado íntegramente por AWS con claves S3.	Sin costo de clave. Complejidad nula.	Opción por defecto cuando se requiere cifrado simple sin control de acceso granular a claves ni auditoría KMS.
SSE-KMS (AWS-managed)	Automático (Gestionado por AWS cada año).	Clave gratuita; se pagan peticiones API a KMS.	Adecuado si se requiere cifrado gestionado por KMS en una sola cuenta, sin personalizar Key Policies.
SSE-KMS (Customer-managed CMK)	Configurable (Rotación automática opcional).	Costo mensual por clave ($1/mes) + peticiones API.	**Obligatorio para acceso cruzado entre cuentas (cross-account)** y cumplimiento de control estricto de claves.
Cifrado en Cliente (*Client-Side*)	Gestionado fuera de los servicios de AWS por la aplicación.	Alta complejidad de código y gestión de claves.	Requerido cuando los datos deben cifrarse antes de transitar por la red hacia la infraestructura de AWS.

1.4. Gestión de Configuración y Secretos: Systems Manager Parameter Store vs AWS Secrets Manager

El desarrollo de software seguro prohíbe incluir credenciales, claves de API o cadenas de conexión a bases de datos directamente en el código fuente de las aplicaciones. AWS proporciona dos servicios especializados para desacoplar estos datos sensibles: AWS Systems Manager Parameter Store y AWS Secrets Manager.

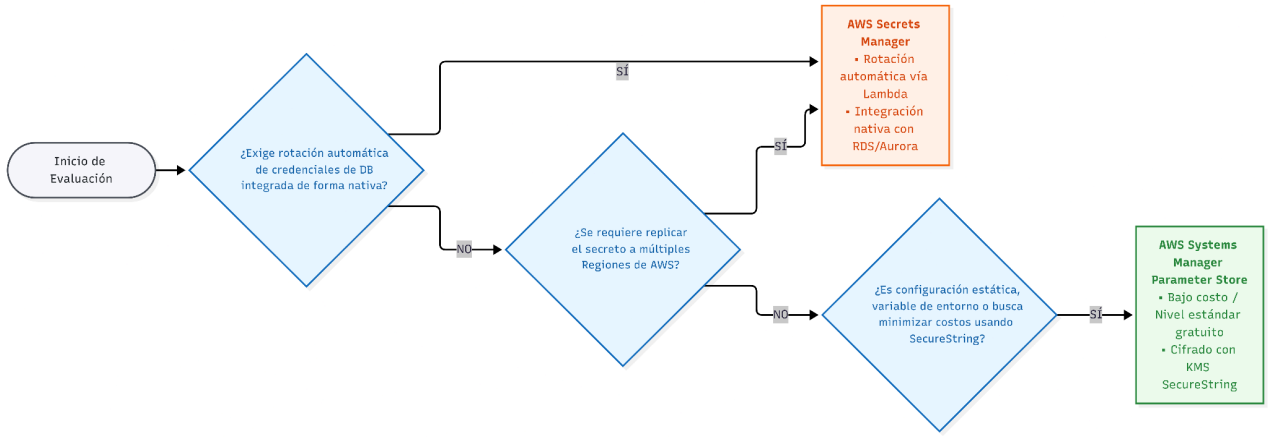
AWS Systems Manager Parameter Store permite almacenar datos de configuración como valores clave-texto. Admite parámetros en texto plano o cifrados mediante el tipo SecureString. Los parámetros SecureString utilizan AWS KMS de forma transparente para cifrar y descifrar los valores. Parameter Store es un servicio de bajo costo y alta eficacia

para configuraciones estáticas, variables de entorno o secretos que no requieren esquemas automatizados de rotación.

AWS Secrets Manager está diseñado específicamente para la gestión de credenciales confidenciales avanzadas (como contraseñas de bases de datos Amazon RDS, credenciales de Amazon DocumentDB o tokens de API de terceros). Sus diferenciadores clave frente a Parameter Store determinan su elección en el examen:

1. **Rotación Automática de Secretos** : Integración nativa con funciones AWS Lambda prefabricadas para rotar periódicamente y sin intervención humana las credenciales en bases de datos (RDS MySQL, PostgreSQL, Aurora, etc.), actualizando la contraseña en el motor de base de datos y en el secreto simultáneamente.

2. **Replicación entre Regiones** : Capacidad de replicar secretos automáticamente a múltiples regiones de AWS para dar soporte a arquitecturas de recuperación ante desastres (*Disaster Recovery*) o aplicaciones multirregión.



La correcta segregación de identidades, claves criptográficas y secretos asegura la capa lógica y de datos. Sin embargo, esta protección se vuelve vulnerable si la red por la que transita el tráfico no se encuentra debidamente aislada a nivel de VPC.

2. Seguridad en Redes de la VPC e Interconexión Privada

2.1. Firewalls Virtuales y Control de Tráfico: Security Groups (Stateful) vs NACLs (Stateless)

La estrategia de defensa en profundidad dentro de Amazon Virtual Private Cloud (VPC) requiere implementar controles de seguridad multicapa para mitigar intrusiones de red. Las dos herramientas fundamentales para el filtrado de tráfico son los Grupos de Seguridad (*Security Groups*) y las Listas de Control de Acceso a la Red (*Network Access Control Lists* - NACLs).

Los **Security Groups** operan como firewalls virtuales a nivel de la interfaz de red elástica (ENI) o instancia de EC2. Son **con estado** (*stateful*) y **solo admiten reglas de permisión** (*allow rules*). Que un Security Group sea *stateful* significa que si se permite una petición de entrada (*inbound*), el tráfico de respuesta o salida (*outbound*) asociado a

esa conexión se autoriza automáticamente, independientemente de las reglas de salida configuradas.

Por otro lado, las **NACLs** operan a nivel del límite de la subred. Son componentes **sin estado** (*stateless*) y evalúan sus reglas en orden numérico ascendente estricto (por ejemplo, la regla 100 se procesa antes que la regla 200). Al ser *stateless* , si se autoriza una conexión de entrada en el puerto 443, la NACL **no** autorizará el tráfico de retorno de forma automática; es estrictamente necesario configurar una regla explícita en la tabla de salida para los puertos efímeros (rango 1024-65535). Asimismo, las NACLs admiten reglas de permisión (*allow*) y de denegación explícita (*deny*), resultando idóneas para bloquear direcciones IP maliciosas específicas a nivel de subred.

Un patrón de diseño esencial para el examen es la **autorreferenciación de Security Groups** (*Security Group Chaining*). En lugar de definir reglas que autoricen el tráfico entre capas mediante bloques CIDR rígidos (ej. 10.0.1.0/24), se configura la regla del Security Group del destino especificando el ID del Security Group del origen (ej. sg-web-tier). Esto garantiza que la capa de base de datos solo acepte tráfico procedente de instancias que pertenezcan explícitamente al grupo sg-web-tier, independientemente de cuáles sean sus direcciones IP privadas.

Característica	Security Groups	Network ACLs (NACLs)	Comportamiento en Tráfico de Retorno	Patrón de Uso en Examen
Nivel de Operación	Interfaz de Red (ENI) / Instancia EC2	Límite de la Subred	**SG**: *Stateful* (Tráfico de retorno permitido automáticamente)	**SG**: Control de acceso granular n-capas mediante autorreferenciació n (`sg-id`).
Tipo de Reglas	Solo permisión (*Allow*)	Permisión (*Allow*) y Denegación (*Deny*)	**NACL**: *Stateless* (Tráfico de retorno requiere abrir puertos efímeros)	**NACL**: Bloqueo explícito de direcciones IP maliciosas o rangos CIDR a nivel subred.
Procesamiento	Evaluadas en su totalidad conjuntamente	Orden numérico ascendente estricto	**SG**: Sin estado de puerto efímero / **NACL**: Requiere regla saliente `1024-65535`	**SG**: Aislar capas de app / **NACL**: Protección perimetral de subred.

El aislamiento estricto de las subredes privadas plantea el desafío de cómo administrar operacionalmente las instancias privadas sin exponerlas a la Internet pública.

2.2. Acceso Remoto Seguro: Bastion Hosts e Integración con Rangos IP Corporativos

Exponer puertos administrativos de gestión como SSH (puerto 22) o RDP (puerto 3389) a la Internet pública representa un riesgo severo de vectores de ataque de fuerza bruta. Las arquitecturas seguras en AWS exigen restringir el acceso operativo estrictamente a canales autorizados.

El modelo tradicional utiliza un **Bastion Host** (o *Jump Box*). Un Bastion Host es una instancia EC2 endurecida (*hardened*) situada en una **subred pública** de la VPC. Las instancias alojadas en las subredes privadas no poseen direcciones IP públicas y solo aceptan tráfico de administración proveniente exclusivamente del Security Group del Bastion Host. Para cumplir con el menor privilegio, el Security Group del Bastion Host debe restringir el tráfico de entrada en el puerto 22/3389 **únicamente al rango CIDR correspondiente a la IP pública corporativa** (x.x.x.x/32 o bloque de la empresa).

Sin embargo, las mejores prácticas actuales evaluadas en el SAA-C03 priorizan el uso de **AWS Systems Manager Session Manager** sobre los Bastion Hosts tradicionales. Session Manager permite conectarse a instancias privadas mediante una consola en navegador o AWS CLI sin necesidad de infraestructura intermedia, aportando ventajas arquitectónicas clave:

- **Sin puertos de entrada abiertos** : Elimina la necesidad de abrir el puerto 22 o 3389 en los Security Groups.
- **Sin IP pública ni Bastion Host** : Las instancias privadas no requieren IP pública ni gateways de salto expuestos.
- **Auditoría y Registro Centralizado** : Cada comando ejecutado se audita y registra en Amazon CloudWatch Logs o en buckets de Amazon S3.

Pasos para Diseñar un Bastion Host Seguro (Patrón Tradicional de Examen)

1. Desplegar una instancia EC2 con una AMI mínima y endurecida en una **subred pública** de la VPC.
2. Asignar una dirección Elastic IP (EIP) al Bastion Host.
3. Configurar el Security Group del Bastion Host permitiendo entrada SSH (22) o RDP (3389) **únicamente desde la IP pública corporativa (x.x.x.x/32)** .
4. Ubicar las instancias de la aplicación en **subredes privadas** sin direcciones IP públicas.
5. Configurar el Security Group de las instancias privadas permitiendo tráfico en el puerto 22/3389 **únicamente desde el ID del Security Group del Bastion Host (sg-bastion)** .

Garantizar el acceso remoto seguro a las instancias privadas es el primer paso; el siguiente es permitir que dichas cargas de trabajo consuman servicios gestionados de AWS sin exponer su tráfico a la Internet pública.

2.3. Acceso Privado a Servicios de AWS: VPC Gateway Endpoints vs VPC Interface Endpoints (AWS PrivateLink)

Cuando las instancias situadas en subredes privadas consumen servicios gestionados de AWS (como Amazon S3, DynamoDB, SNS o Kinesis), dirigir dicho tráfico hacia Internet a través de un dispositivo NAT Gateway genera costos por procesamiento de datos, introduce latencia y viola los principios de aislamiento estricto.

Los **VPC Endpoints** permiten conectar de forma privada una VPC a los servicios respaldados por AWS sin requerir un Internet Gateway, NAT Gateway, conexión VPN o AWS Direct Connect.

AWS divide los VPC Endpoints en dos tecnologías con características operativas y de costo marcadamente distintas:

Gateway Endpoints :

- **Servicios Soportados** : Disponibles **exclusivamente para Amazon S3 y Amazon DynamoDB** .
- **Mecanismo** : Funcionan mediante la adición de una entrada de ruta en la **tabla de enrutamiento** (*Route Table*) de la subred privada, dirigiendo el tráfico de la prefijo-lista del servicio directamente al endpoint.
- **Estructura de Costos** : **Sin costo adicional** (gratuitos). No consumen interfaces de red en la subred ni aplican tarifas por hora o transferencia de datos.

Interface Endpoints (AWS PrivateLink) :

- **Servicios Soportados** : Disponibles para la gran mayoría de servicios de AWS (Amazon SNS, SQS, Kinesis, CloudWatch, Systems Manager, etc.) y servicios alojados por terceros.
- **Mecanismo** : Despliegan una interfaz de red elástica (*Elastic Network Interface* - ENI) con una dirección IP privada dentro de la subred seleccionada. La resolución DNS privada dirige las peticiones hacia la IP privada de la ENI.
- **Estructura de Costos** : Aplica una tarifa por hora de provisión del endpoint más un costo por Gigabyte (GB) de datos procesados.

En cuanto al almacenamiento de archivos compartidos dentro de la VPC, el examen requiere evaluar **Amazon EFS (Elastic File System)** . EFS es un sistema de archivos de archivos POSIX totalmente gestionado que permite el acceso concurrente desde múltiples instancias EC2 a través de subredes y Zonas de Disponibilidad. Para optimizar costos, EFS dispone de políticas de ciclo de vida que mueven automáticamente los archivos poco accedidos a la clase de almacenamiento **EFS Infrequent Access (EFS IA)** . Además, ofrece dos modos de rendimiento (*General Purpose* para baja latencia frente a *Max I/O* para altísima concurrencia) y dos modos de tasa de transferencia (*Bursting* frente a *Provisioned*).

Criterio	Gateway Endpoints	Interface Endpoints (AWS PrivateLink)

Servicios Soportados	Exclusivamente **Amazon S3** y **Amazon DynamoDB**.	Amazon SNS, SQS, Kinesis, CloudWatch, SSM, Athena y servicios de terceros.
Componente de Red Utilizado	Entrada de destino en la **Tabla de Enrutamiento** (*Route Table*) de la VPC.	Interfaz de Red Elástica (**ENI**) provista con IP privada en la subred.
Estructura de Costos	**Gratis** (sin costo por hora ni por gigabyte procesado).	Tarifa por hora por ENI provista + costo por GB procesado.
Modificación de Tablas de Ruta	**Obligatoria** (se actualiza la tabla de ruta asignada a la subred).	**No requerida** (el tráfico se resuelve mediante DNS local apuntando a la ENI).

El uso de VPC Endpoints asegura que el tráfico no transite por el NAT Gateway ni por Internet. Sin embargo, cuando se necesita extender la red corporativa local (*on-premises*) hacia la VPC de forma privada y cifrada, se deben emplear soluciones de conectividad híbrida.

2.4. Conectividad Segura Híbrida: AWS Site-to-Site VPN

Las arquitecturas de nube híbrida requieren interconectar los centros de datos corporativos locales (*on-premises*) con las redes de AWS garantizando la privacidad y el cifrado de la información en tránsito.

Una conexión **AWS Site-to-Site VPN** crea un túnel cifrado mediante el protocolo IPsec entre la red corporativa y la VPC. Los componentes esenciales de esta arquitectura son:

- **Customer Gateway (CGW)** : El dispositivo físico de red o la aplicación software ubicada en la infraestructura *on-premises* del cliente. Debe disponer de una IP pública fija y enrutable.
- **Virtual Private Gateway (VGW)** : El concentrador de VPN en el extremo de AWS adjuntado a una VPC específica.
- **Transit Gateway (TGW)** : Un conmutador de red centralizado que actúa como *hub* para interconectar cientos de VPCs y redes *on-premises* mediante un punto único de terminación VPN.

Para asegurar una alta disponibilidad, una conexión AWS Site-to-Site VPN establece automáticamente **dos túneles IPsec redundantes en paralelo** orientados a dos Zonas de Disponibilidad distintas dentro de AWS. En las preguntas del examen SAA-C03, la opción VPN es la respuesta correcta cuando se cumplen los siguientes criterios:

1. Necesidad de cifrado IPsec en tránsito de extremo a extremo a través de Internet.
2. Requerimiento de despliegue rápido e inmediato con bajo costo inicial en comparación con la instalación física de enlaces dedicados (*AWS Direct Connect*).

La infraestructura de red privada y segura proporciona la base sobre la cual se deben superponer los servicios de protección perimetral, la gobernanza global y los sistemas de autenticación de aplicaciones.

3. Gobernanza, Protección de Aplicaciones y Gestión Centralizada de Seguridad

3.1. Protección Perimetral y Mitigación de Ataques: AWS WAF y AWS Shield

Las aplicaciones web expuestas hacia Internet requieren protección contra amenazas de la capa de aplicación (Capa 7 del modelo OSI) y ataques volumétricos de Denegación de Servicio Distribuido (DDoS en Capas 3 y 4).

AWS WAF (Web Application Firewall) es un firewall de la **Capa de Aplicación (Capa 7)** que permite inspeccionar las peticiones HTTP/HTTPS entrantes. AWS WAF se integra de forma nativa con los siguientes puntos de entrada:

- Application Load Balancer (ALB)
- Amazon API Gateway
- Amazon CloudFront (Edge Locations)

Mediante el uso de Listas de Control de Acceso Web (*Web ACLs*), AWS WAF permite desplegar **reglas administradas por AWS** , **reglas personalizadas** y **reglas basadas en tasa de peticiones** (*rate-based rules*). Las reglas basadas en tasa de peticiones son cruciales en el examen para mitigar ataques por fuerza bruta o inundaciones HTTP al bloquear temporalmente direcciones IP que superen un umbral definido de peticiones en un intervalo de 5 minutos. AWS WAF inspecciona los encabezados, la dirección IP de origen, las cadenas de consulta de la URI y el cuerpo de las solicitudes para bloquear ataques de Inyección SQL (SQLi) y Scripting entre Sitios (XSS).

Para la protección contra ataques DDoS a nivel de red y transporte, AWS ofrece la suite **AWS Shield** :

- **AWS Shield Standard** : Protección automática y gratuita integrada por defecto en toda la infraestructura de AWS. Mitiga ataques comunes de Capas 3 y 4 (tales como *SYN floods* o ataques de reflexión UDP).
- **AWS Shield Advanced** : Servicio gestionado de pago que proporciona mitigación mejorada contra ataques DDoS complejos. Incluye acceso 24/7 al equipo de respuesta ante DDoS (*DDoS Response Team* - DRT), protección financiera contra picos de costos derivados de picos de tráfico por ataques, y soporte para CloudFront, Route 53, ALB y Network Load Balancers (NLB).

Servicio	Capa OSI que Protege	Puntos de Integración	Escenario Típico de Examen
AWS WAF	**Capa 7** (Aplicación)	ALB, API Gateway, Amazon CloudFront	Bloquear ataques SQLi, XSS, aplicar filtrado por geolocalización o *rate-limiting* por IP.
AWS Shield Standard	**Capas 3 y 4** (Red y Transporte)	Todos los servicios de AWS expuestos a Internet	Protección DDoS básica automática y sin costo adicional para la infraestructura.
AWS Shield Advanced	**Capas 3, 4 y 7** (Protección avanzada)	CloudFront, Route 53, ALB, NLB	Necesidad de mitigación experta asistida 24/7 (DRT) y protección financiera contra sobrecostos por DDoS.

Complementando la defensa del perímetro público, es indispensable asegurar el acceso de los usuarios finales a la aplicación mediante sistemas de gestión de identidades delegadas.

3.2. Autenticación y Control de Usuarios: Amazon Cognito User Pools y Gobernanza de Identidad de Aplicación

Es vital diferenciar la autorización de infraestructura (controlada mediante IAM para otorgar permisos sobre recursos de AWS a desarrolladores o servicios) de la autenticación de usuarios finales que interactúan con una aplicación móvil o web. **Amazon Cognito** es el servicio gestionado para controlar la identidad y el acceso de los usuarios finales.

El componente central para la autenticación es **Amazon Cognito User Pools** . Un *User Pool* es un directorio de usuarios totalmente gestionado que proporciona funciones de registro, inicio de sesión, recuperación de credenciales y administración de perfiles. Características principales de Cognito User Pools:

- **Autenticación de Múltiples Factores (MFA)** : Implementación de MFA basada en SMS o aplicaciones de contraseñas temporales basadas en tiempo (TOTP).
- **Federación de Identidades** : Integración con proveedores de identidad corporativos mediante SAML o OAuth 2.0, e inicio de sesión social (Google, Facebook, Apple, Amazon).
- **Generación de Tokens** : Emisión de JSON Web Tokens (JWT) firmados para autorizar solicitudes en el backend.

"So What?" para el Examen: Cognito User Pools vs Cognito Identity Pools

- **Cognito User Pools (Directorio y Autenticación)** : Se utiliza cuando la pregunta solicita **gestionar el directorio de usuarios, manejar registros, inicios de sesión y autenticación de la aplicación** .

- **Cognito Identity Pools (Autorización de Recursos AWS)** : Se utiliza cuando el requerimiento es **intercambiar la autenticación de un usuario (proveniente de un User Pool o proveedor federado) por credenciales temporales de IAM** , permitiendo que el cliente acceda directamente a recursos subyacentes de AWS (por ejemplo, subir un objeto directamente a un bucket de Amazon S3).

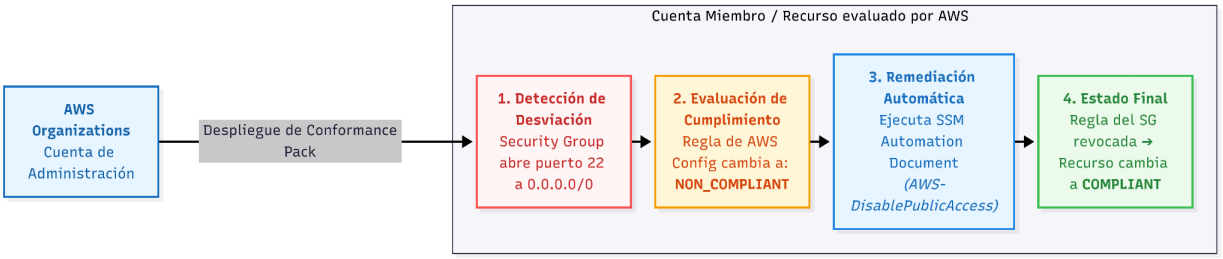
La seguridad a nivel de aplicación e identidad debe respaldarse con la capacidad de auditar y hacer cumplir de manera continua las políticas de configuración en entornos empresariales multicuenta.

3.3. Gobernanza, Auditoría y Remediación Automática: AWS Config Conformance Packs y AWS Organizations

En entornos empresariales multicuenta, garantizar el cumplimiento normativo requiere soluciones automatizadas que auditen continuamente la configuración de los recursos y corrijan desviaciones de seguridad respecto a marcos internacionales (tales como PCI-DSS, NIST SP 800-53, CIS Benchmarks y HIPAA).

AWS Config es el servicio encargado de rastrear, auditar y evaluar las configuraciones de los recursos de AWS respecto a las políticas deseadas. Para aplicar esta gobernanza en arquitecturas multicuenta, se utilizan los **AWS Config Conformance Packs** . Un *Conformance Pack* es una colección empaquetada de reglas de AWS Config y acciones de remediación compiladas dentro de una sola plantilla de AWS CloudFormation. Mediante la integración con **AWS Organizations** , un equipo de seguridad centralizado puede desplegar estos paquetes de conformidad en todas las cuentas de la organización desde la cuenta pagadora o cuenta delegada de administración.

Para completar el ciclo de gobernanza, AWS Config ofrece **remediación automática** cuando un recurso es evaluado como no conforme (*NON_COMPLIANT*). Esto se logra asociando la regla de AWS Config con documentos de automatización de AWS Systems Manager (**SSM Automation Documents**).



Este mecanismo automatizado garantiza la remediación inmediata de brechas de seguridad sin intervención manual, manteniendo la postura de cumplimiento de toda la organización en tiempo real.

4. Matriz Estratégica para el Examen SAA-C03: Lógica de Decisión, Palabras Clave y Trampas Frecuentes

4.1. Lógica de Decisión Rápida

Para resolver eficientemente las preguntas del examen SAA-C03, el estudiante debe aplicar el patrón metodológico: **Requisito del Negocio → Problema Técnico → Servicio Clave de AWS → Patrón de Arquitectura Recomendado**.

Requisito del Negocio / Examen	Problema o Desafío Técnico	Servicio / Característica Clave	Patrón de Arquitectura Recomendado
Aislamiento de permisos en contenedores	Múltiples contenedores en la misma EC2 host no deben compartir credenciales IAM.	**ECS Task Roles**	Definir un *Task Role* específico en la *Task Definition* en lugar de otorgar permisos al perfil de la instancia host EC2.
Compartir Snapshots cifrados entre cuentas	No se pueden compartir *snapshots* cifrados con claves administradas por AWS (`aws/ebs`).	**AWS KMS (Customer-managed CMK)**	Volver a cifrar el *snapshot* usando una CMK, actualizar la *Key Policy* para dar acceso a la cuenta destino y compartir el *snapshot*.
Acceso privado a S3 / DynamoDB	Conectar con S3/DynamoDB desde subredes privadas sin costo adicional ni salir de la VPC.	**VPC Gateway Endpoint**	Crear un Gateway Endpoint en la VPC y asociarlo a la tabla de enrutamiento (*Route Table*) de la subred privada.
Conexión privada a servicios de AWS (SNS, SQS)	Consumir servicios gestionados sin exponer tráfico a Internet ni usar NAT Gateways.	**VPC Interface Endpoint (PrivateLink)**	Desplegar una ENI con IP privada en la subred privada y asociarle un Security Group.

Administración remota sin puertos abiertos	Conectarse por SSH/RDP a instancias privadas sin IP pública ni abrir el puerto 22/3389.	**AWS Systems Manager Session Manager**	Instalar el agente SSM en las instancias privadas y acceder mediante IAM sin abrir puertos de entrada en el SG.
Rotación automática de credenciales de DB	Cumplir normativas rotando automáticamente la contraseña de Amazon RDS.	**AWS Secrets Manager**	Almacenar credenciales en Secrets Manager y activar la rotación automática mediante la función Lambda integrada.
Protección contra Inyección SQL y XSS	Filtrar peticiones maliciosas en la Capa 7 expuestas en un ALB o CloudFront.	**AWS WAF**	Configurar Web ACLs con reglas administradas contra SQLi/XSS y aplicar *rate-based rules*.
Gobernanza multicuenta automatizada	Desplegar y auditar reglas de cumplimiento (PCI-DSS, NIST) en toda la empresa.	**AWS Config Conformance Packs + Organizations**	Desplegar paquetes de conformidad de AWS Config centralizadamente desde AWS Organizations y vincular remediación con SSM.

4.2. Glosario de Palabras Clave y Desglose de Trampas Habituales del Examen Real

Catálogo de Palabras Clave de Activación (Trigger Words)

- **"Acceso a S3/DynamoDB sin costo de transferencia y mediante tablas de ruta"** → **VPC Gateway Endpoint**.
- **"Rotación automática de credenciales de base de datos con Lambda"** → **AWS Secrets Manager** (descartar Parameter Store).
- **"Reglas con estado (Stateful) para filtrar por ID de Security Group"** → **Security Group (Autorreferenciación / sg-id)**.

- **"Denegación explícita de IP maliciosa a nivel de subred"** → **Network ACL (NACL)**.

- **"Acceso de administración sin IP pública ni puertos de entrada abiertos"** → **AWS Systems Manager Session Manager**.

- **"Cifrado de snapshots para compartir entre múltiples cuentas de AWS"** → **AWS KMS Customer-Managed Key (CMK)**.

- **"Autenticación de usuarios finales e inicio de sesión social / SAML"** → **Amazon Cognito User Pools**.

- **"Obtención de credenciales temporales de IAM para usuarios de aplicación"** → **Amazon Cognito Identity Pools**.

- **"Desacoplamiento de microservicios asíncronos garantizando estricto ordenamiento de mensajes"** → **Amazon SQS FIFO**.

- **"Almacenamiento de bloques de ultra baja latencia físicamente adjunto al host (efímero)"** → **EC2 Instance Store**.

- **"Gobernanza centralizada de copias de seguridad en entornos multicuenta"** → **AWS Backup**.

Las 5 Trampas Más Frecuentes del Dominio 1 en el Examen SAA-C03

1. **La Trampa del NAT Gateway para Amazon S3** : El enunciado propondrá utilizar un NAT Gateway en una subred pública para permitir que instancias privadas suban datos a Amazon S3. Aunque técnicamente funciona, **es la respuesta incorrecta** si existe la opción de un *VPC Gateway Endpoint* , ya que este último no transita por Internet, ofrece mayor rendimiento y no tiene costo adicional.

2. **La Trampa del Rol de Instancia EC2 para Contenedores ECS** : En preguntas sobre clústeres de Amazon ECS en EC2, las respuestas distractoras sugerirán añadir permisos al rol de la instancia host EC2. Esta opción violará el principio de menor privilegio. La respuesta correcta siempre exigirá el uso de **ECS Task Roles** .

3. **La Trampa de Compartir Snapshots Cifrados con Claves por Defecto (aws/ebs)** : El examen presentará un escenario donde falla la compartición de una AMI o *snapshot* cifrado con otra cuenta. Las opciones distractoras sugerirán editar la política de la clave aws/ebs. **La trampa es que las claves administradas por AWS no se pueden compartir.** La solución correcta requiere copiar el *snapshot* cifrándolo con una *Customer-managed CMK* y compartir tanto el *snapshot* como la clave CMK.

4. **La Trampa de Parameter Store para Rotación Automatizada** : Se solicitará guardar credenciales de base de datos garantizando rotación automática cada 30 días. Parameter Store con SecureString aparecerá como la opción económica. **Es una trampa.** Parameter Store no posee motor nativo de rotación de secretos; la respuesta correcta que resuelve la rotación automática con Lambda es **AWS Secrets Manager** .

5. **La Trampa del Bastion Host Abierto a 0.0.0.0/0** : Al evaluar el diseño de un Bastion Host, las opciones erróneas permitirán tráfico SSH desde 0.0.0.0/0 para facilitar el acceso remoto. **Esta opción viola las buenas prácticas de seguridad.** La regla del Security Group del Bastion Host debe restringirse **únicamente al bloque CIDR/IP pública corporativa explícita (x.x.x.x/32)** , o preferir **AWS Systems Manager Session Manager** para eliminar por completo el uso de puertos de entrada.

Dominio 2 - Diseño de Arquitecturas Resilientes (AWS SAA-C03)

1. Introducción Estratégica al Dominio 2 (26% del Examen SAA-C03)

En el examen de certificación **AWS Certified Solutions Architect - Associate (SAA-C03)** , el Dominio 2, denominado **"Diseñar Arquitecturas Resilientes"** , representa de manera exacta el **26% del contenido puntuado** . Dentro del esquema de puntuación compensatoria implementado por AWS —donde los puntajes escalados se sitúan en un rango de 100 a 1.000 puntos con un umbral mínimo de aprobación fijado en 720—, este dominio se consolida como el eje técnico fundamental de la evaluación. De las 65 preguntas totales que componen el examen (50 evaluables y 15 no puntuadas destinadas a calibración estadística), más de una cuarta parte analiza de forma directa la competencia del aspirante para diseñar entornos distribuidos capaces de tolerar fallos de infraestructura, adaptarse elásticamente a picos imprevistos de carga y recuperarse de eventos catastróficos sin interrupción del servicio ni intervención manual.

La filosofía del pilar de **Resiliencia** dentro del *AWS Well-Architected Framework* exige superar el modelo tradicional de mitigación reactiva y asumir una premisa de diseño fundamental: todos los componentes de un sistema fallarán en algún momento. Por lo tanto, la arquitectura debe proyectarse desde su origen para eliminar los puntos únicos de fallo (**SPOF** , por sus siglas en inglés *Single Points of Failure*) y automatizar completamente los procesos de recuperación. Para cuantificar y planificar la tolerancia a fallos, las decisiones de diseño deben alinearse rigurosamente con dos métricas operativas de recuperación que condicionan los acuerdos de nivel de servicio (**SLA**) de la organización:

- **Recovery Point Objective (RPO):** El objetivo de punto de recuperación establece el umbral máximo de pérdida de datos tolerable ante un fallo, medido como la brecha temporal entre el último estado de datos persistido correctamente y el momento de la interrupción.
- **Recovery Time Objective (RTO):** El objetivo de tiempo de recuperación representa la cantidad máxima de tiempo de inactividad que el negocio puede soportar antes de que la interrupción del sistema cause un impacto inaceptable.

Para resolver con éxito los escenarios complejos presentados en el examen SAA-C03, el arquitecto de soluciones debe aplicar de forma metódica una secuencia de deducción técnica basada en la siguiente fórmula:

Requisito de Negocio ➔ Problema de Arquitectura ➔ Servicio AWS ➔ Patrón de Diseño ➔ Opción Correcta

Dominar este flujo de razonamiento resulta indispensable para identificar las soluciones óptimas y descartar los distractores plausibles pero ineficientes desde la perspectiva de la resiliencia. La alta disponibilidad y la elasticidad a nivel de la capa de cómputo constituyen la primera línea de defensa de toda infraestructura distribuida ante fluctuaciones de carga e interrupciones de zona.

2. Alta Disponibilidad y Escalabilidad Elástica en Cómputo (EC2, ALB y Auto Scaling)

La capa de cómputo actúa como el motor de procesamiento directo de la aplicación. Para absorber incrementos imprevistos de demanda y garantizar que la pérdida de un servidor físico o la degradación completa de una Zona de Disponibilidad (**AZ** , *Availability Zone*) no afecten la continuidad operativa, es imprescindible integrar sistemas de balanceo de carga inteligentes con grupos de cómputo auto-escalables.

2.1 Application Load Balancer (ALB) y Balanceo Multizona

El **Application Load Balancer (ALB)** opera en la capa 7 (capa de aplicación) del modelo OSI y distribuye el tráfico HTTP/HTTPS entrante hacia múltiples objetivos, como instancias Amazon EC2, contenedores de Amazon ECS y direcciones IP privadas repartidas en múltiples AZs.

- **Algoritmos y Reglas de Enrutamiento Avanzativo:** ALB evalúa el tráfico entrante mediante reglas con prioridades numéricas que permiten enrutar peticiones en función de la ruta URL (*path-based routing*), el nombre de host (*host-based routing*), encabezados HTTP, métodos HTTP o parámetros de consulta. Cada regla redirige el tráfico hacia un **Target Group** específico.

- **Target Groups y Comprobaciones de Estado (Active vs. Passive Health Checks):**
    - *Active Health Checks:* El ALB envía peticiones periódicas (HTTP, HTTPS o TCP) a los objetivos registrados según un intervalo, umbral de éxitos y umbral de fallos prefijados. Si un objetivo no responde adecuadamente, el ALB interrumpe de inmediato el envío de tráfico a dicho recurso.
    - *Passive Health Checks:* El ALB monitorea en tiempo real el tráfico transaccional real. Si un objetivo comienza a retornar errores a nivel de transporte o tiempos de espera agotados en peticiones reales, el balanceador puede detectar la anomalía y marcar el recurso como no saludable antes de que el siguiente *active health check* sea ejecutado.

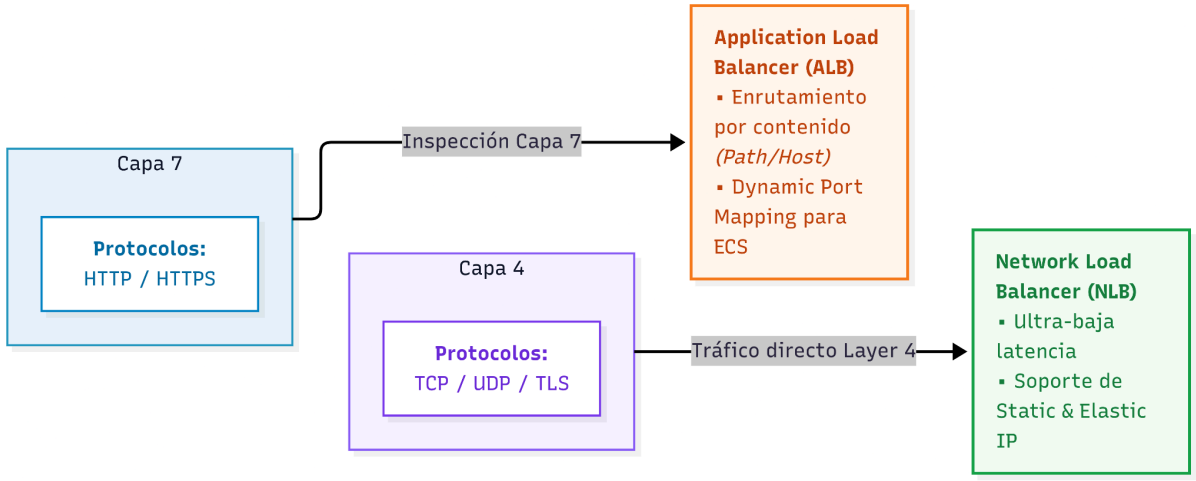
- **Afinidad de Sesión (Sticky Sessions) y Connection Draining:** ALB admite la afinidad de sesión (*sticky sessions*) utilizando galletas (*cookies*) emitidas por el

balanceador o por la aplicación para dirigir las peticiones repetidas de un cliente al mismo objetivo. Para garantizar actualizaciones sin caídas, el mecanismo de **Connection Draining** (*Deregistration Delay*) otorga un tiempo de gracia configurable para que las peticiones activas sobre un objetivo en proceso de desvinculación o terminación concluyan antes de cerrar la conexión.

- **Mapeo Dinámico de Puertos para Contenedores:** ALB se integra nativamente con Amazon ECS permitiendo el *dynamic host port mapping* . Esto posibilita que múltiples tareas o contenedores pertenecientes al mismo servicio se ejecuten simultáneamente en la misma instancia EC2 sobre puertos efímeros asignados dinámicamente.

- **Balanceo Multizona (Cross-Zone Load Balancing):** Al activar *Cross-Zone Load Balancing* , el ALB distribuye uniformemente el tráfico entrante entre el 100% de los objetivos registrados en todas las AZs habilitadas, previniendo la saturación de nodos en AZs con menor número relativo de instancias.

Diferenciación Técnica de Balanceadores:



Mientras que ALB está optimizado para aplicaciones HTTP/HTTPS, Network Load Balancer (NLB) opera en la Capa 4 (Transporte), siendo capaz de gestionar millones de peticiones por segundo con una latencia ultrabaja, ofreciendo compatibilidad con direcciones IP estáticas y reenvío directo (*pass-through*) para tráfico TCP/UDP.

2.2 Auto Scaling Groups (ASG) y Lógica Interna de Terminación

Los **Amazon EC2 Auto Scaling Groups (ASG)** automatizan la provisión y el desaprovisionamiento de instancias EC2 para mantener la capacidad de cómputo deseada ante variaciones de carga.

Launch Templates vs. Launch Configurations:

- *Launch Configurations:* Mecanismo clásico de definición de parámetros de instancia (AMI, tipo de instancia, pares de claves, security groups). Representa un artefacto inmutable y obsoleto que no admite múltiples versiones ni características avanzadas.

- *Launch Templates:* La opción recomendada por AWS. Permite la gestión de versiones, herencia de parámetros y es un requisito obligatorio para habilitar características modernas como el despliegue combinado de instancias On-Demand e **instancias Spot** , el uso de **Dedicated Hosts** (para cumplimiento de licencias de software de terceros) y la especificación de múltiples tipos de instancia dentro del mismo ASG.

Políticas de Escalado:

- *Target Tracking Scaling:* Mantiene una métrica objetivo constante (por ejemplo, mantener la utilización media de CPU del grupo al 60%). Es la política preferida en el examen por su ajuste dinámico bidireccional automático.

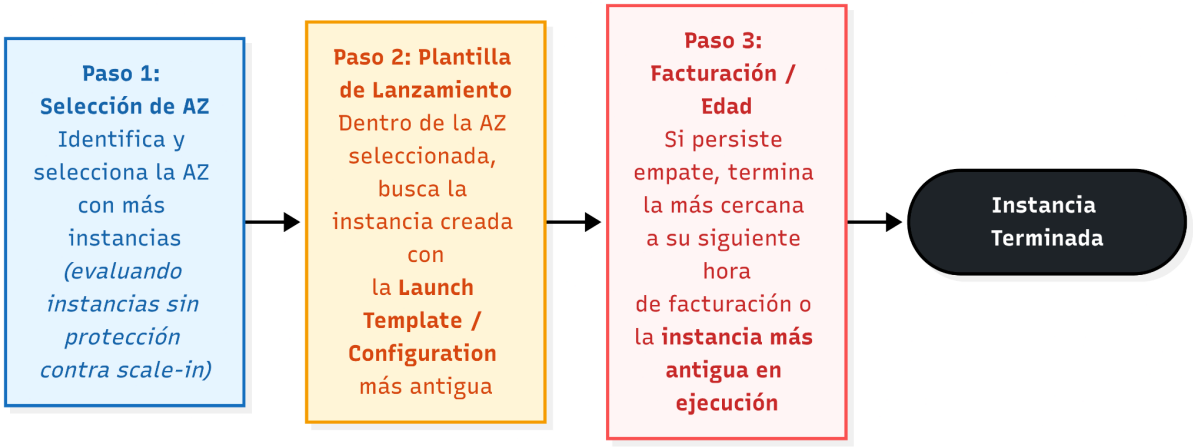
- *Step Scaling:* Escala la capacidad en tramos o "pasos" basados en la magnitud del umbral rebasado en las alarmas de Amazon CloudWatch.

- *Simple Scaling:* Ajusta la capacidad mediante una única alarma de CloudWatch y exige esperar un periodo de enfriamiento (*cooldown*) antes de permitir nuevas acciones de escalado.

- *Scheduled Scaling:* Escala la capacidad de acuerdo con un calendario o patrón de tráfico predecible prefijado.

Ganchos de Ciclo de Vida (Lifecycle Hooks): Permiten pausar el estado de lanzamiento o terminación de una instancia EC2 dentro del ASG para ejecutar scripts de inicialización, extracción de logs o análisis forense antes de completar la transición de estado.

Política Predeterminada de Terminación de Instancias en ASG (Default Termination Policy): Cuando un evento de *scale-in* o reducción de capacidad es disparado, el ASG debe seleccionar qué instancia terminar. Para prevenir el desbalanceo entre zonas y conservar la configuración más reciente, el ASG ejecuta de forma estricta la siguiente secuencia de evaluación de 3 pasos:



1. **Selección de la Zona de Disponibilidad:** Identifica la AZ que contiene el mayor número de instancias EC2 asignadas a este ASG. Si todas las AZs tienen el mismo

número de instancias, selecciona la AZ con la instancia no protegida contra el *scale-in* .

2. **Evaluación de la Plantilla de Lanzamiento:** Dentro de la AZ elegida, identifica cuál de las instancias fue creada utilizando la versión más antigua de la *Launch Template* (o *Launch Configuration*).

3. **Evaluación de la Hora de Facturación y Edad:** Si existen múltiples instancias ejecutando la misma versión de la plantilla, selecciona la instancia más cercana a su siguiente hora de facturación (criterio histórico para optimización de costo) o, en su defecto, la instancia más antigua en ejecución (*oldest instance*).

2.3 Gestión de Service Quotas y Cuellos de Botella de Aprovisionamiento

En escenarios de escalado masivo e imprevisto (como eventos de ventas flash o ataques de denegación de servicio), la capacidad de un Auto Scaling Group puede verse severamente bloqueada no por falta de recursos en AWS, sino por los límites operativos por defecto asignados a la cuenta de AWS.

AWS impone un límite predeterminado sobre el número total de *vCPUs* de EC2 que pueden ejecutarse simultáneamente por región y por familia de instancias. Si un ASG intenta escalar para absorber un pico masivo de tráfico pero la cuenta ha alcanzado el límite de *vCPUs* de la región, la llamada a la API de EC2 para lanzar nuevas instancias fallará silenciosamente con un error de cuota superada (*QuotaExceededException*).

Para evitar fallos en cascada y estrangulamiento de la provisión:

- Las arquitecturas deben monitorear y gestionar proactivamente los límites de cómputo mediante la **Service Quotas API** o la integración con **Amazon CloudWatch Alarms** .

- Se deben configurar alertas cuando el uso de *vCPUs* alcance el 80% de la cuota regional, y solicitar incrementos de cuota (*Quota Increase Requests*) de forma automatizada o anticipada antes de eventos de alta demanda previsibles.

2.4 Tabla Comparativa y Trampas del Examen

Componente de Cómputo/Escalado	Caso de Uso Ideal en Examen	Trampas/Distractores Comunes en la Pregunta

Application Load Balancer (ALB)	Aplicaciones web basadas en HTTP/HTTPS que requieren enrutamiento por ruta URL, nombre de host o mapeo dinámico de puertos en contenedores ECS.	Proponer ALB para balancear tráfico de bajo nivel a nivel de red (TCP/UDP de baja latencia) o donde se requieran IPs estáticas (el servicio correcto es NLB).
Launch Templates	Definición flexible de instancias EC2 con gestión de versiones, soporte para instancias Spot mixtas, On-Demand y Dedicated Hosts.	Seleccionar *Launch Configurations* en opciones que exigen actualizar la configuración de lanzamiento o utilizar funciones avanzadas de EC2.
Target Tracking Scaling	Mantenimiento automático de una métrica de rendimiento constante (ej. utilización media de CPU al 50% o recuento de peticiones por objetivo).	Configurar *Scheduled Scaling* para cargas de trabajo con picos impredecibles o de comportamiento dinámico.
Service Quotas API	Prevención proactiva de fallos de provisionamiento en Auto Scaling Groups durante eventos masivos de escalado elástico.	Ignorar las cuotas de *vCPUs* a nivel de cuenta/región asumiendo que un ASG puede escalar indefinidamente sin ajustes administrativos previas.

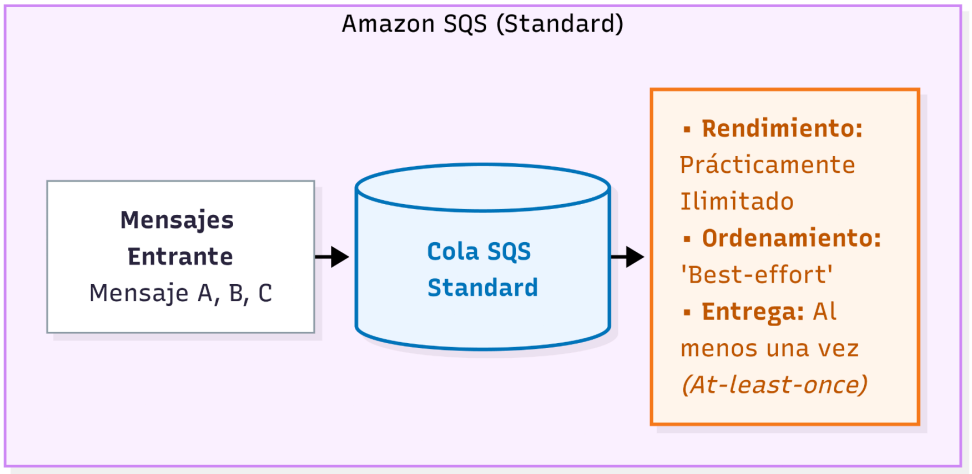
Una capa de cómputo elástica e invulnerable a fallos locales pierde efectividad si sus componentes están acoplados de forma síncrona; desacoplar los servicios mediante arquitecturas orientadas a eventos es indispensable para mitigar la propagación de fallos en cadena.

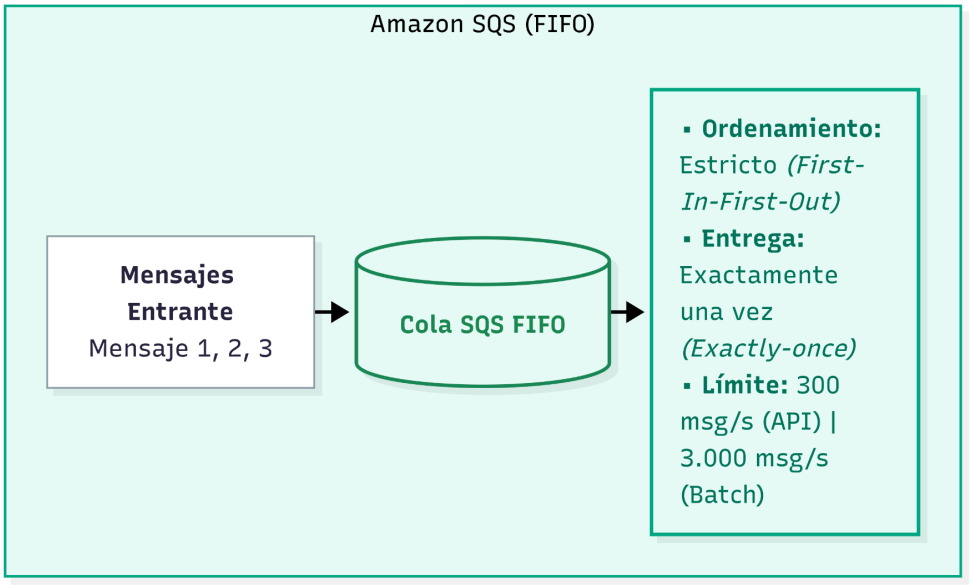
3. Arquitecturas Desacopladas y Orientadas a Eventos (Event-Driven Architecture)

El desacoplamiento asíncrono transforma las dependencias directas en interacciones basadas en mensajes almacenados en búferes intermedios. Este patrón evita que un pico repentino de peticiones en la capa de entrada colapse los servicios del backend, garantizando que cada componente procese la carga a su propio ritmo sin perder transacciones.

3.1 Amazon SQS (Standard vs. FIFO) y Manejo de Errores

Amazon SQS (Simple Queue Service) es un servicio completamente administrado de colas de mensajes que habilita la comunicación asíncrona entre microservicios y componentes *serverless* .





SQS Standard vs. SQS FIFO:

- *SQS Standard:* Ofrece un rendimiento de mensajes prácticamente ilimitado. No obstante, no garantiza un ordenamiento estricto (utiliza un ordenamiento de "mejor esfuerzo" o *best-effort*) y garantiza la entrega del mensaje *al menos una vez* (*at-least-once*), lo que exige que los consumidores sean idempotentes frente a posibles duplicados.

- *SQS FIFO (First-In-First-Out):* Garantiza que los mensajes sean procesados exactamente en el orden de envío y con una entrega *exactamente una vez* (*exactly-once*). Elimina duplicados mediante deduplicación automática basada en contenido o en un *Deduplication ID* .

- *Límites de Rendimiento en SQS FIFO:* SQS FIFO tiene una capacidad nominal limitada por diseño a **300 mensajes por segundo** (300 operaciones de envío, lectura o borrado por segundo usando llamadas API individuales). Al implementar el agrupamiento de transacciones (*batching*) de hasta 10 mensajes por llamada API, SQS FIFO alcanza un límite máximo de **3.000 mensajes por segundo** .

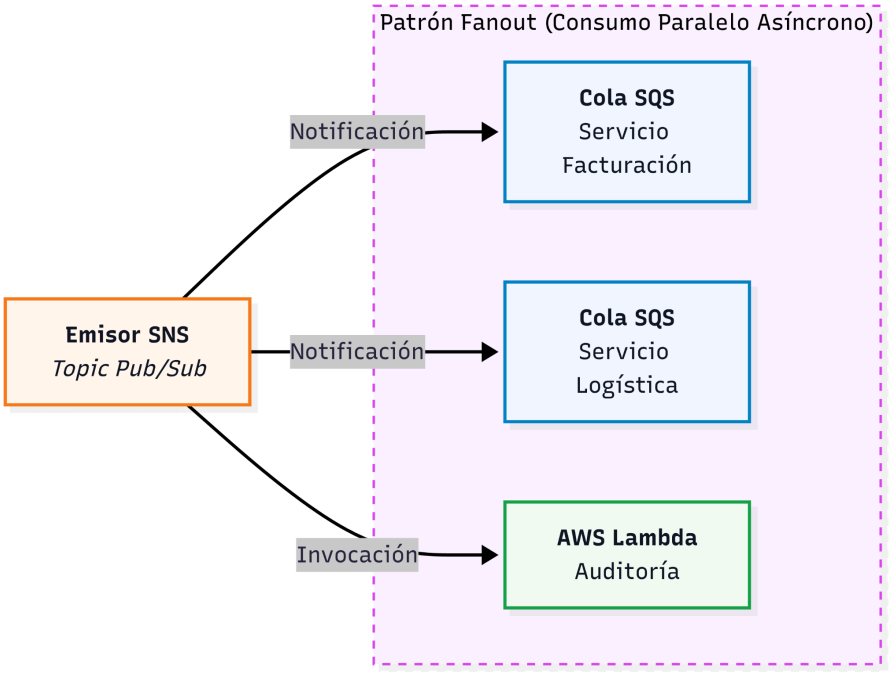
Visibility Timeout: Es el periodo durante el cual SQS oculta un mensaje del resto de los consumidores mientras un trabajador activo lo está procesando. Si el *Visibility Timeout* expira antes de que el trabajador elimine explícitamente el mensaje de la cola, el mensaje vuelve a estar visible, provocando un procesamiento duplicado.

Dead-Letter Queue (DLQ) y Redrive Policy: Una DLQ es una cola SQS secundaria utilizada para aislar mensajes que fallan repetidamente durante su procesamiento. Mediante la política de reenvío (*redrive policy*), el arquitecto configura el parámetro maxReceiveCount. Cuando un mensaje es leído sin éxito dicho número de veces, SQS desplaza automáticamente el mensaje a la DLQ para su inspección posterior.

Daemon sqsd en Elastic Beanstalk Worker Environment: En un entorno de tipo *Worker Environment* de AWS Elastic Beanstalk, un proceso *daemon* nativo denominado sqsd se ejecuta en cada instancia EC2. El sqsd realiza *long-polling* continuo contra la cola SQS configurada y reenvía el contenido del mensaje como una petición HTTP POST local (http://localhost/) a la aplicación web del trabajador. Si la aplicación devuelve un código HTTP 200 OK, el sqsd elimina el mensaje de la cola.

3.2 Amazon SNS y Patrones Fanout

Amazon SNS (Simple Notification Service) es un servicio de mensajería del tipo Publicador/Suscriptor (Pub/Sub). Un único tema de SNS (*SNS Topic*) puede diseminar notificaciones de forma simultánea hacia múltiples suscriptores (colas SQS, funciones Lambda, endpoints HTTP/S, correos electrónicos).



El patrón de arquitectura **Fanout** se construye publicando un mensaje único en un tema de SNS que se replica automáticamente hacia múltiples colas SQS paralelas. Esto permite que diferentes sistemas procesen la misma información de forma totalmente aislada, asíncrona y paralela.

3.3 Ingestión en Tiempo Real: Amazon EventBridge vs. Amazon Kinesis Data Streams

- **Amazon EventBridge:** Bus de eventos *serverless* para la integración de aplicaciones mediante eventos procedentes de servicios AWS, aplicaciones SaaS (como Salesforce o Zendesk) o fuentes personalizadas. Utiliza reglas con patrones JSON para filtrar y enrutar eventos hacia múltiples objetivos (Lambda, SQS, Step Functions).

- **Amazon Kinesis Data Streams:** Servicio diseñado para la ingestión e interconexión masiva de flujos de datos continuos en tiempo real (logs telemáticos, métricas financieras, streams de datos de IoT). Los datos se dividen en *shards* de capacidad fija (1 MB/s de entrada, 2 MB/s de salida) y permiten la retención de datos configurable (de 24 horas a 365 días) con acceso concurrente para múltiples consumidores independientes.

3.4 Resumen Estructurado de Decisión

- **Requisito: "Garantizar el procesamiento ordenado de transacciones financieras sin duplicados"**
  Solución de Arquitectura: **Amazon SQS FIFO** (considerando el límite de $300 msg/s$ o $3.000 msg/s$ en batch).

- **Requisito: "Notificar un mismo evento de compra a múltiples microservicios independientes de forma asíncrona"**
  Solución de Arquitectura: **Patrón SNS Fanout conectado a múltiples colas SQS Standard** .

- **Requisito: "Ingestar y retener peticiones masivas de telemetría de IoT para análisis concurrente por varios motores"**
  Solución de Arquitectura: **Amazon Kinesis Data Streams** .

- **Requisito: "Reaccionar automáticamente a cambios de estado en recursos de AWS o eventos de aplicaciones SaaS"**
  Solución de Arquitectura: **Amazon EventBridge** .

Garantizar la resiliencia en el transporte e intercambio de eventos debe ir acompañado de una estrategia equivalente a nivel de persistencia de datos relacionales y no relacionales.

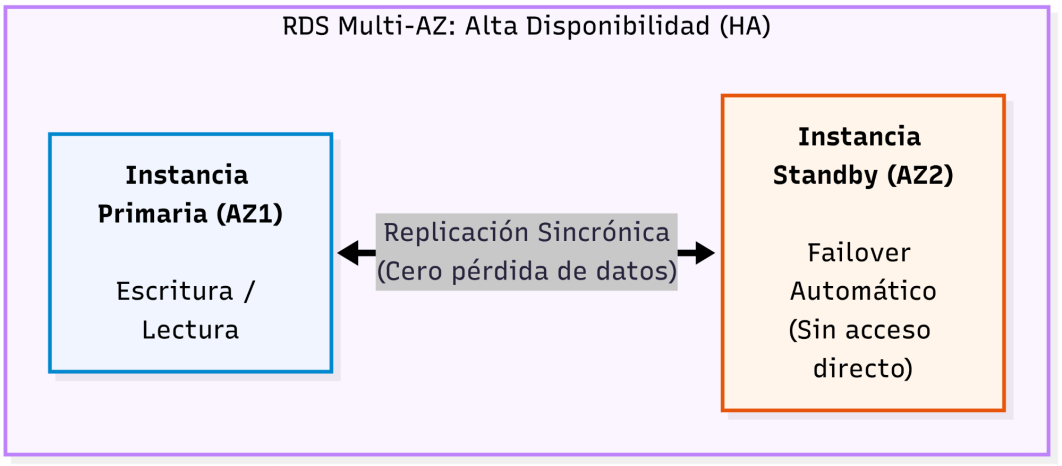
4. Bases de Datos Resilientes y Alta Disponibilidad de Datos

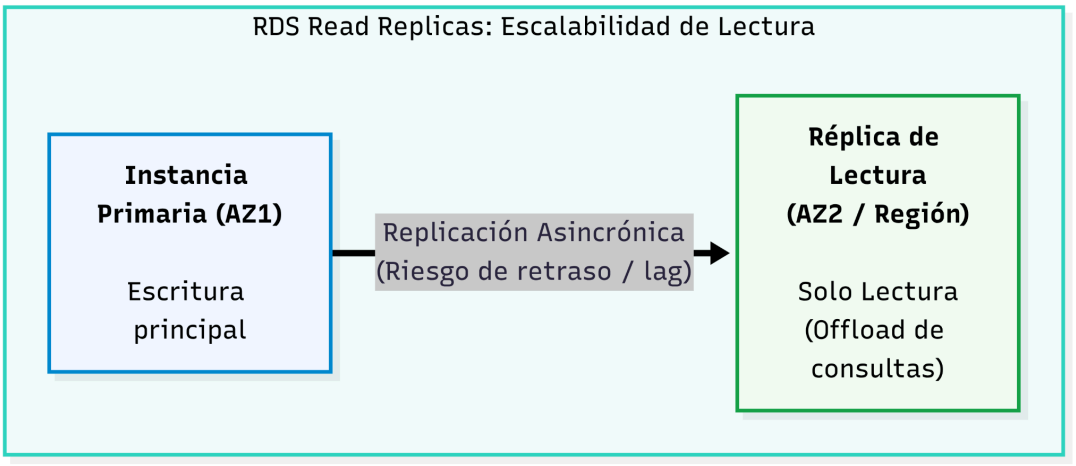
El diseño de una capa de almacenamiento de datos resiliente requiere equilibrar las necesidades de alta disponibilidad local (conmutación transparente dentro de la misma

región) frente a las estrategias de continuidad de negocio y recuperación ante desastres a escala global.

4.1 Amazon RDS Multi-AZ vs. Read Replicas

En **Amazon RDS** , la alta disponibilidad y la capacidad de lectura se abordan mediante dos arquitecturas conceptualmente independientes:





Amazon RDS Multi-AZ:

- *Propósito:* **Alta Disponibilidad (HA)** y tolerancia a fallos a nivel de infraestructura o AZ.

- *Mecanismo:* Mantiene una instancia secundaria pasiva (*Standby*) en una AZ distinta mediante **replicación sincrónica** a nivel de almacenamiento.

- *Comportamiento de Failover:* Ante una caída de la primaria, RDS conmuta de forma automática y transparente hacia la instancia Standby. El endpoint DNS del servidor de base de datos permanece idéntico, evitando cambios de configuración en la aplicación.

- *Punto Clave de Examen:* RDS Multi-AZ **no** escala el rendimiento de lectura; la instancia *Standby* no puede procesar consultas directas de lectura.

Amazon RDS Read Replicas (Réplicas de Lectura):

- *Propósito:* **Escalabilidad de rendimiento para cargas con alta demanda de lectura** (*read-heavy workloads*).

- *Mecanismo:* Utiliza **replicación asincrónica** desde la instancia primaria.

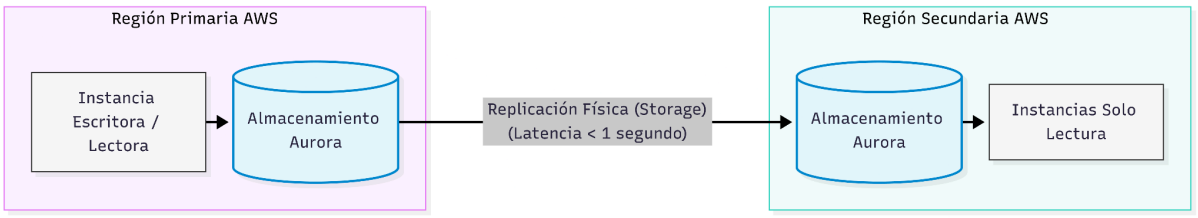
- *Capacidades:* Se pueden crear múltiples réplicas en la misma AZ, en AZs distintas o en **diferentes regiones de AWS**. Una réplica de lectura se puede promover a base de datos independiente. Dado el carácter asíncrono de la replicación, existe un retraso (*replication lag*) que implica un $RPO > 0$.

4.2 Amazon Aurora y Aurora Global Database

Amazon Aurora rearchitecta el almacenamiento relacional dividiendo el nivel de cómputo del almacenamiento subyacente.

- **Almacenamiento Distribuido Intra-Región:** Un clúster de Aurora distribuye 6 copias de los datos a través de 3 Zonas de Disponibilidad dentro de una misma región, soportando la pérdida de hasta 2 copias sin afectar la capacidad de escritura, y de hasta 3 copias sin afectar la lectura. Admite hasta 15 *Aurora Read Replicas* con latencia de replicación de un solo dígito de milisegundo.

- **Amazon Aurora Global Database:** Para aplicaciones con alcance global o requisitos de recuperación ante desastres críticos, **Aurora Global Database** extiende el clúster a múltiples regiones de AWS.



La replicación se realiza directamente en la capa de almacenamiento físico sin consumir ciclos de CPU en los nodos de cómputo de la base de datos primaria. Esta arquitectura ofrece:

- **RPO típico:** Menos de 1 segundo ($RPO < 1 segundo$).
- **RTO típico:** Menos de 1 minuto ($RTO < 1 minuto$), permitiendo la promoción de la región secundaria a capacidad total de escritura en caso de desastre regional sin pérdida de rendimiento en la región primaria.

4.3 Amazon DynamoDB y DynamoDB Global Tables

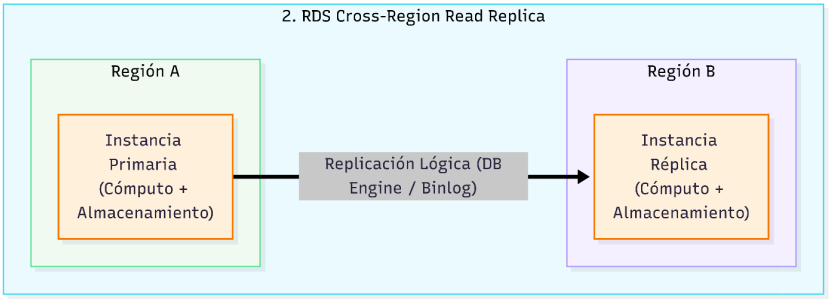
Amazon DynamoDB es una base de datos NoSQL de clave-valor completamente administrada que ofrece latencias de milisegundos de un solo dígito a cualquier escala.

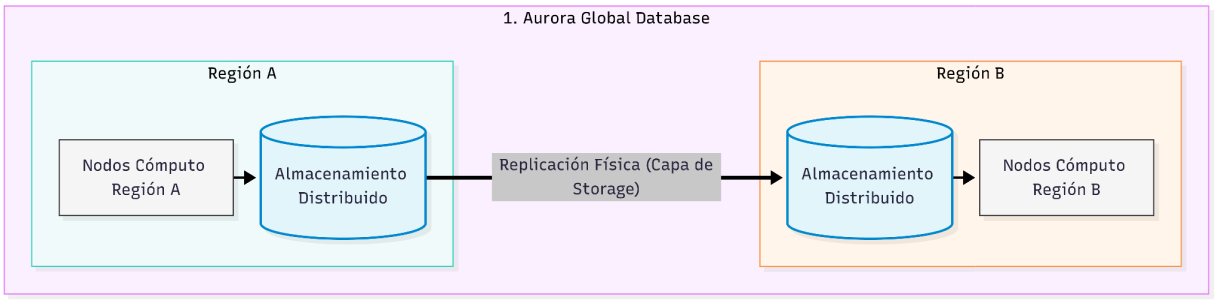
Resiliencia Intra-Región: DynamoDB almacena y replica automáticamente los datos de las tablas en 3 Zonas de Disponibilidad de la región.
Amazon DynamoDB Global Tables: Es una solución completamente administrada de base de datos **activa-activa multirregional** . Permite que aplicaciones distribuidas globalmente realicen operaciones de lectura y escritura locales sobre tablas réplica desplegadas en múltiples regiones de AWS.

Mecanismo: DynamoDB utiliza **DynamoDB Streams** de forma subyacente para propagar automáticamente las modificaciones entre las réplicas regionales con una consistencia eventual que se alcanza habitualmente en segundos.

Ventaja Operativa: Ofrece latencias de lectura y escritura ultrabajas a nivel global, eliminando la necesidad de canalizaciones complejas de replicación personalizada.

4.4 Matriz de Decisión de Almacenamiento Persistente





Motor de BBDD / Servicio	Mecanismo de Replicación / Alta Disponibilidad	Objetivo de Recuperación (RPO / RTO)	Caso de Uso Principal en Examen
Amazon RDS Multi-AZ	Replicación sincrónica a nivel de almacenamiento en 2 AZs de la misma región.	$RPO \approx 0$ / RTO de minutos (Failover automático).	Alta Disponibilidad y tolerancia a fallos local en bases de datos relacionales empresariales.

Amazon RDS Read Replicas	Replicación asincrónica intra-región o cross-region.	$RPO > 0$ / RTO variable según promoción.	Descargar tráfico de lectura de la base de datos primaria y posibilitar DR de costo moderado.
Amazon Aurora Global Database	Replicación física directa a nivel de almacenamiento distribuido entre regiones.	$RPO < 1 segundo$ / $RTO < 1 minuto$.	Aplicaciones relacionales globales de misión crítica con requerimientos de DR ultrarrápidos sin impacto en escrituras.
Amazon DynamoDB Global Tables	Replicación activa-activa multirregional completamente administrada vía DynamoDB Streams.	$RPO \approx 0$ (Consistencia eventual) / $RTO \approx 0$.	Aplicaciones NoSQL globales que requieren lecturas y escrituras de baja latencia con conmutación multi-maestro.

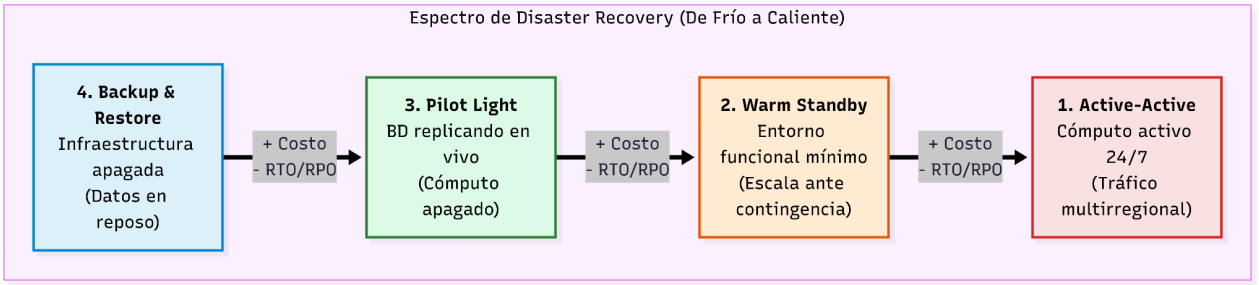
Las estrategias de resiliencia en la capa de datos deben estar articuladas dentro del plan general de Recuperación ante Desastres de la organización.

5. Estrategias Avanzadas de Disaster Recovery (DR) y Continuidad de Negocio

La planificación de Recuperación ante Desastres (**DR**) consiste en preparar la infraestructura para sobrevivir a fallos regionales catastróficos. El diseño de una solución de DR exige evaluar el balance entre el costo operativo de mantener recursos standby en una región secundaria y el costo financiero resultante de la pérdida de datos (RPO) y del tiempo de inactividad (RTO).

5.1 Evaluación Comparativa de las 4 Estrategias Tradicionales de DR

AWS clasifica las estrategias de DR en 4 patrones principales:



Backup and Restore (Copia de Seguridad y Restauración):

- *Características:* Mínimo costo operativo; máximos valores de RTO y RPO.
- *Mecanismo:* Se generan respaldos periódicos de datos (snapshots de EBS, copias de seguridad de RDS, objetos en S3) que se replican a una región secundaria mediante *Cross-Region Replication* (CRR) o **AWS Backup** . Ante un desastre, la

infraestructura se despliega desde cero mediante plantillas de **AWS CloudFormation** .

Pilot Light (Luz Piloto):

- *Características:* RTO y RPO reducidos con un costo financiero moderado.
- *Mecanismo:* Los datos de la aplicación se mantienen sincronizados en tiempo real en la región secundaria (mediante réplicas de lectura de RDS cross-region o Aurora Global Database). Sin embargo, la capa de cómputo (instancias EC2, servicios de aplicación) permane apagada o desaprovisionada; únicamente existen las AMIs y *Launch Templates* listas para ser lanzadas en caso de contingencia.

Warm Standby (Espera Activa):

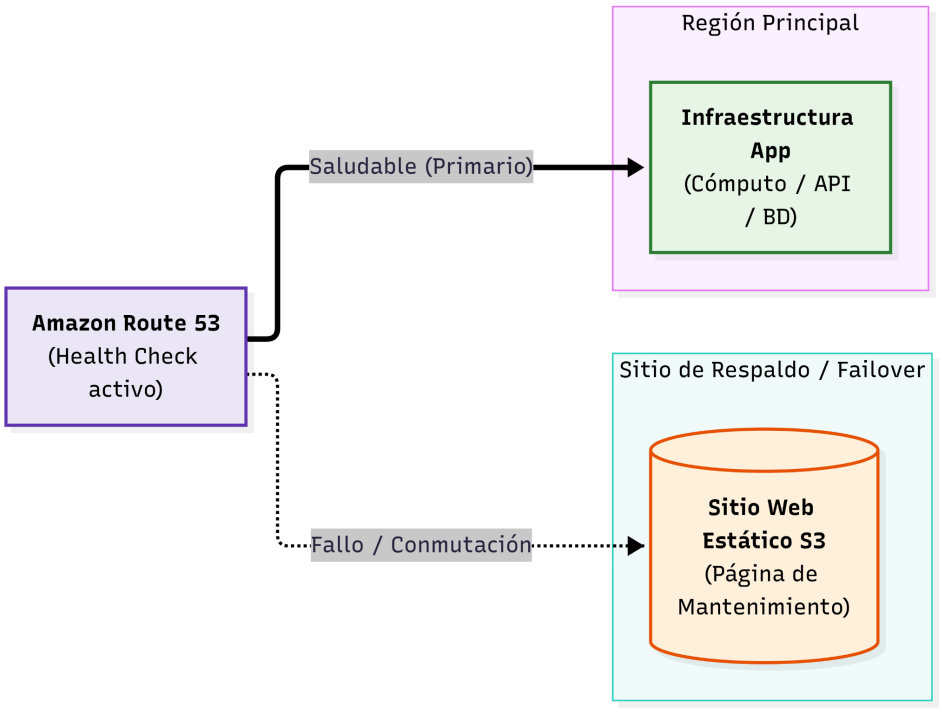
- *Características:* Bajos valores de RTO y RPO.
- *Mecanismo:* Una versión completamente funcional pero escalada al mínimo (*scaled-down*) de la infraestructura se ejecuta de forma continua en la región secundaria. Durante un evento de contingencia, la región secundaria asume el tráfico de producción simplemente incrementando la capacidad del Auto Scaling Group.

Multi-Region Active-Active / Hot Standby:

- *Características:* RTO y RPO de prácticamente cero; máximo costo operativo.
- *Mecanismo:* La infraestructura completa está desplegada y operativa a plena capacidad en dos o más regiones de AWS de forma simultánea, procesando tráfico balanceado continuamente.

5.2 Route 53 Failover Routing y Resguardo Estático en S3

Amazon Route 53 gestiona la conmutación de tráfico ante desastres mediante políticas de enrutamiento por conmutación por error (**Failover Routing Policy**).



- **Mecanismo de Evaluación:** Route 53 enlaza *Health Checks* continuos al endpoint de la aplicación principal.

- **Patrón de Resguardo Estático (Static Website Backup):** Si las comprobaciones de estado determinan que la aplicación principal no responde, Route 53 redirige automáticamente las peticiones DNS hacia un sitio web estático alojado en un **bucket de Amazon S3** (configurado con una interfaz degradada de solo lectura o una página de mantenimiento), preservando la experiencia de usuario durante la resolución del incidente.

5.3 AWS Backup y Replicación Cross-Region

AWS Backup centraliza la protección de datos mediante políticas automatizadas.

- **Gestión Centralizada:** Permite definir planes de respaldo (*backup plans*) automatizados para volúmenes EBS, instancias RDS, tablas DynamoDB, sistemas de archivos EFS y máquinas EC2.

- **Replicación Cross-Region y Cross-Account:** AWS Backup permite copiar automáticamente los puntos de recuperación hacia otras regiones de AWS (*Cross-Region Backup*) y hacia cuentas secundarias aisladas (*Cross-Account Backup*), protegiendo la organización contra la pérdida regional o el compromiso de credenciales maestras.

Comprender la teoría de la resiliencia es el primer paso; el segundo consiste en aplicar la lógica de decisión precisa necesaria para abordar las preguntas del examen real.

6. Lógica de Decisión, Palabras Clave y Trampas Típicas del Examen Real

El examen SAA-C03 evalúa la capacidad del aspirante para balancear compensaciones arquitectónicas (*trade-offs*). Mapear palabras clave en el enunciado de la pregunta permite identificar la solución idónea y descartar distractores.

6.1 Mapeo de Patrones de Arquitectura Resiliente (Cheat Sheet de Decisión)

- **Requisito en Enunciado:** *"Desacoplar componentes garantizando un procesamiento estrictamente ordenado de mensajes sin duplicidad."*

  → **Solución de Arquitectura:** Cola **Amazon SQS FIFO** (considerando el límite de throughput de $300 msg/s$ o $3.000 msg/s$ con batching) + procesamiento mediante instancias EC2 o AWS Lambda.

- **Requisito en Enunciado:** *"Base de datos relacional con failover automático transparente y cero pérdida de datos en la misma región."*

  → **Solución de Arquitectura: Amazon RDS en despliegue Multi-AZ**.

- **Requisito en Enunciado:** *"Base de datos relacional con problemas de rendimiento ocasionados exclusivamente por un volumen masivo de consultas de lectura."*

→ **Solución de Arquitectura:** Crear **Amazon RDS Read Replicas** para desviar el tráfico de lectura.

- **Requisito en Enunciado:** *"Base de datos relacional multirregional con RPO en segundos, RTO en minutos y sin degradación de escrituras primarias."*

  → **Solución de Arquitectura: Amazon Aurora Global Database**.

- **Requisito en Enunciado:** *"Estrategia de Disaster Recovery multirregional con el menor costo operativo y tolerancia a tiempos de recuperación prolongados."*

  → **Solución de Arquitectura:** Estrategia **Backup and Restore** utilizando **AWS Backup** con replicación entre regiones.

- **Requisito en Enunciado:** *"Redirigir a los usuarios a un mensaje de mantenimiento degradado si la aplicación principal cae por completo."*

  → **Solución de Arquitectura: Amazon Route 53 Failover Routing** con destino a un sitio web estático en **Amazon S3**.

6.2 Trampas Frecuentes y Distractores en el Dominio 2

Confundir RDS Multi-AZ con un mecanismo para escalar el rendimiento de lectura:

- *La trampa:* La pregunta describe lentitud en la base de datos por exceso de consultas SELECT y ofrece como opción "Convertir la base de datos RDS a Multi-AZ".

- *Por qué es un distractor:* RDS Multi-AZ es un mecanismo pasivo para Alta Disponibilidad (HA) y conmutación por error. La instancia *Standby* no permite lecturas ni escrituras. La solución correcta para escalar lecturas es implementar **Read Replicas** .

Asumir que SQS Standard garantiza el orden secuencial de los mensajes:

- *La trampa:* El enunciado exige que las transacciones financieras se procesen estrictamente en la secuencia en que fueron recibidas y la opción propone "Usar SQS Standard".

- *Por qué es un distractor:* SQS Standard realiza un ordenamiento de "mejor esfuerzo" y puede entregar mensajes fuera de orden o duplicados. La respuesta correcta exige **SQS FIFO** .

Ignorar el bloqueo de Auto Scaling por Cuotas de vCPUs sin monitoreo de Service Quotas:

- *La trampa:* La opción propone configurar un Auto Scaling Group extremadamente grande para absorber picos masivos de tráfico, pero omite la gestión previa de límites operativos.

- *Por qué es un distractor:* Sin la gestión proactiva de la **Service Quotas API** , el ASG intentará lanzar instancias y fallará al alcanzar el límite predeterminado de *vCPUs* de EC2 de la cuenta en esa región.

Seleccionar estrategias Active-Active cuando el enunciado requiere costo mínimo:

- *La trampa:* La pregunta solicita un plan de Disaster Recovery añadiendo explícitamente la restricción *"de la manera más económica"* (*most cost-effective*).

- *Por qué es un distractor:* Las soluciones Active-Active o Hot Standby eliminan el RTO/RPO pero duplican el costo de infraestructura. Bajo la restricción de costo mínimo, la opción correcta será **Backup and Restore** o **Pilot Light** .

6.3 Lista de Chequeo de Autoevaluación

Antes de rendir el examen SAA-C03, verifique el dominio completo de las siguientes capacidades clave del Dominio 2:

- [ ] ¿Comprendo la diferencia conceptual entre las métricas RPO (pérdida de datos) y RTO (tiempo de inactividad)?
- [ ] ¿Distingo el comportamiento de comprobaciones de estado activas (*Active Health Checks*) y pasivas (*Passive Health Checks*) en un ALB?
- [ ] ¿Conozco en detalle la secuencia de 3 pasos de la política predeterminada de terminación de instancias en un Auto Scaling Group?
- [ ] ¿Sé cómo utilizar la **Service Quotas API** para evitar el estrangulamiento por límites de *vCPUs* de EC2 en escalados masivos?
- [ ] ¿Conozco los límites de rendimiento exactos de SQS FIFO ($300 msg/s$ API / $3.000 msg/s$ en batch) y su diferencia frente a SQS Standard?
- [ ] ¿Comprendo el patrón SNS Fanout y el funcionamiento del *daemon* `sqsd` en el *Worker Environment* de Elastic Beanstalk?
- [ ] ¿Distingo técnicamente RDS Multi-AZ (sincrónico, HA) de RDS Read Replicas (asincrónico, escalabilidad de lectura)?
- [ ] ¿Identifico los valores de $RPO < 1 segundo$ y $RTO < 1 minuto$ de **Aurora Global Database** frente a réplicas asincrónicas tradicionales?
- [ ] ¿Entiendo el funcionamiento activa-activa multirregional de **DynamoDB Global Tables** apoyado en DynamoDB Streams?
- [ ] ¿Puedo clasificar las 4 estrategias de DR (Backup & Restore, Pilot Light, Warm Standby, Active-Active) en función de su costo, RTO y RPO?
- [ ] ¿Sé configurar la política **Failover Routing** de Amazon Route 53 redirigiendo hacia un bucket de **Amazon S3** estático?

Dominio 3 - Diseñar Arquitecturas de Alto Rendimiento (AWS SAA-C03)

1. Marco Estratégico del Dominio 3 y Lógica General de Resolución

Contexto Estratégico

El Dominio 3, **Diseñar Arquitecturas de Alto Rendimiento** , representa el **24% del contenido evaluado** en el examen AWS Certified Solutions Architect - Associate (SAA-C03). Dentro de la metodología del *AWS Well-Architected Framework* , el pilar de Eficiencia del Rendimiento no evalúa la memorización aislada de catálogos de servicios o parámetros de consola, sino la capacidad del arquitecto para traducir restricciones del negocio en patrones de diseño desacoplados y optimizados. Superar este dominio requiere erradicar la memorización pasiva y adoptar una mentalidad sistemática basada en escenarios de carga: ante cualquier problema técnico, se debe aislar el cuello de botella (I/O, cómputo, red o base de datos) y seleccionar la combinación de servicios AWS que entregue el máximo rendimiento con la menor complejidad operativa y al menor costo.

Desglose del Método de Razonamiento

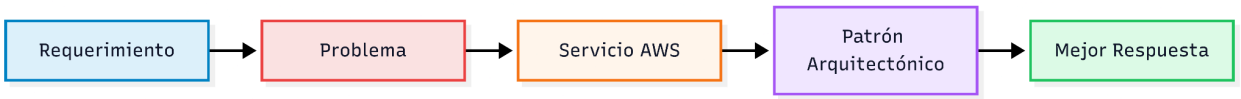
Para deconstruir con precisión analítica las preguntas de escenario en el examen real, aplique rigurosamente la metodología de resolución en cuatro pasos:

1. **Requerimiento (Identificación de Métricas SLA/SLO):** Extraiga del enunciado el objetivo cuantitativo explícito de rendimiento (ej. latencia inferior a un milisegundo, rendimiento de transferencia en GB/s, acceso concurrente multinodo o distribución global acelerada).

2. **Problema (Aislamiento del Cuello de Botella):** Determine qué componente de la infraestructura limita el sistema (saturación de I/O en disco de bloque, contención en la capa NoSQL/relacional, latencia RTT por distancia geográfica o sobrecosto por asignación ineficiente de recursos de cómputo).

3. **Servicio AWS y Patrón Arquitectónico:** Asocie el problema con el patrón de diseño idóneo de AWS (ej. *caching* transparente en memoria, aceleración por red privada de borde, almacenamiento efímero físicamente adjunto o desacoplamiento asíncrono).

4. **Mejor Respuesta (Evaluación de Restricciones y Distractores):** Discrimine la opción correcta evaluando las restricciones implícitas de costo, persistencia y esfuerzo de ingeniería frente a los distractores técnicamente viables pero subóptimos.



Diferenciación e Integración de Métricas Clave de Rendimiento

En el examen SAA-C03, los requerimientos técnicos se traducen en tres límites métricos fundamentales que dictan la selección de la arquitectura:

Latencia (Tiempo de Respuesta en μs o ms): Retardo total para completar una operación de I/O o red.

- *Límite de Microsegundos (μs):* Amazon DynamoDB Accelerator (DAX) para NoSQL, EC2 Instance Store para bloque local.

- *Límite de Milisegundos de un solo dígito (1-9 ms):* Amazon ElastiCache (Redis/Memcached), volúmenes EBS SSD `gp3` / `io2`.

Throughput (Ancho de Banda en MB/s o GB/s): Volumetría de datos transferidos o procesados por unidad de tiempo.

- *Límite de Alto Rendimiento Secuencial:* EBS `st1`, Amazon EFS (*Max I/O* / *Provisioned Throughput*), Amazon Athena con archivos Apache Parquet particionados.

IOPS (Operaciones de Entrada/Salida por Segundo): Cantidad de lecturas/escrituras individuales por segundo (bloques pequeños de 4 KB - 256 KB).

- *Límite Transaccional Crítico:* EBS `io2` / `io2 Block Express`, DynamoDB *Provisioned Capacity*.

Comprender la interacción de estas métricas sobre la capa física y lógica es el requisito previo para evaluar el impacto del almacenamiento en bloque, de archivos y de objetos dentro de AWS.

2. Almacenamiento de Alto Rendimiento: EBS, EFS, Instance Store y S3

Contexto Estratégico

El almacenamiento es el punto de origen más común de degradación de rendimiento en sistemas distribuidos. Una elección desacertada —como emplear almacenamiento en bloque remoto saturado de I/O para procesamiento temporal de imágenes o un sistema de archivos en red para transacciones relacionales críticas— genera cuellos de botella severos de latencia e incurre en sobrecostos masivos por provisionamiento incorrecto.

2.1 Amazon EBS: Comparativa de Tipos y Escalado Dinámico vía Elastic Volumes

Amazon Elastic Block Store (EBS) proporciona almacenamiento de bloque persistente vinculado a una Zona de Disponibilidad (*AZ*) específica. Para tomar decisiones costo-eficientes en el examen, es vital diferenciar la generación gp3 de su predecesora gp2, así como de los volúmenes de IOPS Aprovisionadas io2.

Tipo de Volumen EBS	Rendimiento / IOPS Máximo	Latencia Típica	Caso de Uso Óptimo en Negocio	Criterio de Decisión / Keyword en SAA-C03

General Purpose SSD (gp3)	**3,000 IOPS y 125 MB/s base gratis**. Escalable a 16,000 IOPS y 1,000 MB/s independientemente del tamaño.	Single-digit ms	Módulos de aplicación, bases de datos transaccionales pequeñas/medianas, entornos dev/test.	**Opción por defecto costo-eficiente.** Permite escalar IOPS o MB/s sin pagar por almacenamiento en disco no utilizado.
General Purpose SSD (gp2)	3 IOPS por GiB (Mínimo 100, Máximo 16,000 IOPS). Ráfaga de 3,000 IOPS con *burst credits*.	Single-digit ms	Cargas heredadas, volúmenes de arranque del sistema operativo.	**Subóptimo en costo.** Vincula el rendimiento directamente al tamaño asignado en GB.
Provisioned IOPS SSD (io2 / io2 Block Express)	Hasta 64,000 IOPS por volumen (256,000 IOPS en Block Express) y 4,000 MB/s. Ratio de 500:1 IOPS/GiB.	**Sub-milisegundo a single-digit ms** con 99.999% de durabilidad.	Bases de datos relacionales de misión crítica (Oracle, PostgreSQL, SQL Server, RDS Custom) y NoSQL con transacciones masivas.	**SLA de rendimiento garantizado.** Elija esta opción cuando la pregunta exija rendimiento sostenido sin variaciones de latencia.
Throughput Optimized HDD (st1)	Hasta 500 MB/s y 500 IOPS por volumen.	Latencia HDD (decenas de ms)	*Big Data*, procesamiento de *logs*, data warehouses no estructurados, mapas MapReduce.	**Secuencial masivo económico.** No puede utilizarse como volumen de arranque (*boot volume*).
Cold HDD (sc1)	Hasta 250 MB/s y 250 IOPS por volumen.	Latencia HDD alta	Almacenamiento frío de bloques, datos accedidos infrecuentemente que requieren pocos escaneos al día.	**Costo mínimo absoluto en bloque.** No apto para cargas interactivas ni arranque.

Mecanismo de Escalado Dinámico: EBS Elastic Volumes y la API ModifyVolume

En escenarios donde una aplicación experimenta picos imprevistos de I/O o requiere reestructurar su capa de almacenamiento sin afectar los SLAs de disponibilidad, AWS dispone de la funcionalidad **EBS Elastic Volumes** .

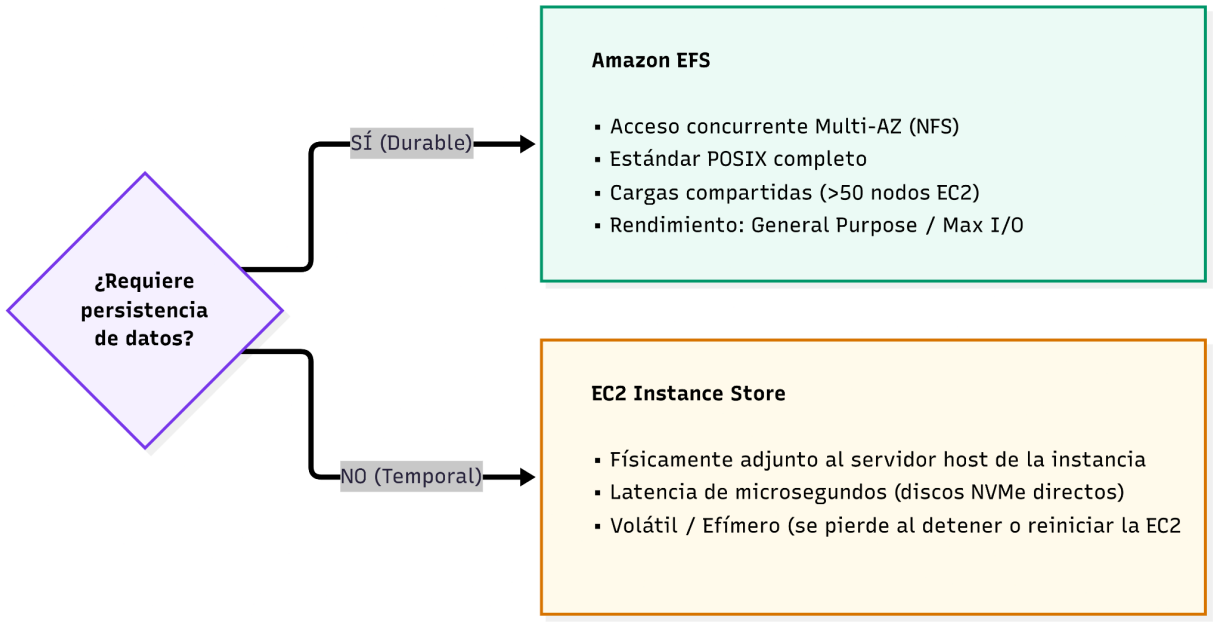
Mediante la llamada a la API o comando CLI ModifyVolume, el arquitecto puede realizar las siguientes modificaciones **en vivo y sin tiempo de inactividad (*zero-downtime*)** :

1. Incrementar la capacidad de almacenamiento del volumen.
2. Ajustar los IOPS aprovisionados o el rendimiento (MB/s) (ej. escalar un gp3 de 3,000 a 10,000 IOPS).
3. Cambiar el tipo de volumen dinámicamente (ej. migrar de gp2 a gp3, o de gp3 a io2 durante un evento de alta demanda transaccional).

Regla del examen: No requiere detener la instancia EC2 ni desmontar el volumen. Una vez ejecutada la modificación, el volumen entra en estado de optimización sin degradar la disponibilidad del sistema.

2.2 Amazon EFS vs. EC2 Instance Store para Sistemas Masivos

Frente a requerimientos de alto rendimiento, el examen SAA-C03 contrasta la persistencia compartida en red de Amazon EFS frente a la velocidad física directa de EC2 Instance Store.



Amazon Elastic File System (EFS): Sistema de archivos de red NFSv4/POSIX totalmente administrado, diseñado para acceso concurrente desde decenas a miles de instancias EC2 simultáneamente a través de múltiples Zonas de Disponibilidad.

- *Modos de Rendimiento:* **General Purpose** (menor latencia por operación de I/O, ideal para servidores web contenidos y CMS) vs **Max I/O** (permite niveles masivos de rendimiento y operaciones concurrentes a costa de latencias ligeramente superiores por operación; ideal para analítica paralela y *Big Data*).

- *Modos de Throughput:* **Bursting Throughput** (el ancho de banda escala en función del tamaño acumulado del sistema de archivos) vs **Provisioned Throughput** (asigna un ancho de banda fijo en MB/s independientemente del almacenamiento utilizado, clave para archivos de poco tamaño que exigen alto rendimiento).

EC2 Instance Store (Almacenamiento Efímero): Unidades de disco SSD/NVMe integradas físicamente en el servidor hardware donde reside la instancia EC2.

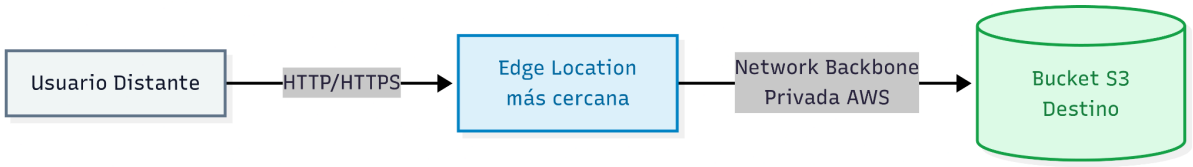
- *Rendimiento:* Ofrece la **más baja latencia de I/O posible (**μs**)** y cientos de miles de IOPS, al eliminar la sobrecarga de la red de almacenamiento.

- *Riesgo y Persistencia:* Es almacenamiento **volátil**. Los datos persisten si el sistema operativo interno se reinicia (*reboot*), pero **se pierden de forma irreversible** si la instancia es detenida (*stop*), terminada (*terminate*) o si el hardware subyacente falla.

- *Lógica de Decisión:* Seleccione *Instance Store* únicamente para cachés temporales, *buffers* de procesamiento en memoria, archivos swap o datos replicados a nivel de aplicación en clústeres distribuídos con tolerancia a fallos (ej. clústeres Elasticsearch/Hadoop).

2.3 Amazon S3 Standard vs. S3 Transfer Acceleration

Amazon S3 es la plataforma de almacenamiento de objetos con durabilidad de 11 nueves (99.999999999%). Sin embargo, para la ingesta de datos a gran escala, la ubicación geográfica del cliente respecto a la Región de AWS introduce cuellos de botella por latencia de red pública.



- **Amazon S3 Standard:** Ingesta directa a través de la red pública de Internet hacia el endpoint regional del bucket (bucketname.s3.amazonaws.com). Expuesto a fluctuaciones de enrutamiento BGP, congestión pública y latencias variables según la distancia.

- **Amazon S3 Transfer Acceleration:** Optimiza las cargas de datos a larga distancia mediante la red de ubicaciones de borde (*Edge Locations*) de Amazon CloudFront. El cliente realiza el *upload* hacia un endpoint mutado optimizado por DNS (bucketname.s3-accelerate.amazonaws.com). Los datos ingresan inmediatamente a la red backbone privada y optimizada de AWS en la ubicación geográfica más

cercana al usuario, viajando hacia el bucket S3 final con latencia reducida y mayor rendimiento constante.

- *Criterio exacto del examen:* Aceleración de cargas hacia un bucket centralizado desde clientes o centros de datos **geográficamente dispersos a nivel global** . *Nota técnica:* Transfer Acceleration no almacena en caché el objeto subido en la Edge Location; actúa estrictamente como un puente de red acelerado de Capa 4/7.

Una vez optimizada la capa de almacenamiento persistente y de objetos, la arquitectura debe acelerar la capa de cómputo y descargar las bases de datos mediante cachés en memoria.

3. Cómputo Escalable y Estrategias de Caché: Lambda, ElastiCache, DAX y Athena

Contexto Estratégico

Aumentar la capacidad de cómputo en la capa de aplicación sin desacoplar la capa de base de datos genera colapsos por contención de I/O. Para soportar millones de solicitudes concurrentes, las arquitecturas modernas combinan la escalabilidad del cómputo *serverless* con capas intermadias de almacenamiento en caché en memoria, eliminando consultas redundantes antes de que impacten a los motores relacionales o NoSQL.

3.1 Optimización de Cómputo Serverless con AWS Lambda

AWS Lambda ejecuta código sin gestión de servidores, escalando automáticamente en función del volumen de eventos entrantes.

Relación Proporcional Recursos-Memoria: En Lambda, la RAM es la **única palanca de configuración de rendimiento** (asignable desde 128 MB hasta 10,240 MB). AWS no permite configurar la CPU de forma independiente: al incrementar la memoria RAM, AWS asigna de manera estrictamente proporcional mayor potencia de CPU, ancho de banda de red y priorización de I/O.

Demostración Matemática del "Efecto Paradoja" (Costo vs. Rendimiento): Dado que la facturación de Lambda se calcula mediante la fórmula $Costo = MemoriaAsignada(GB) \times TiempodeEjecución(segundos)$, incrementar la RAM asignada a una función con carga intensiva de CPU puede reducir drásticamente el tiempo de ejecución, resultando en un costo total inferior por invocación.

Demostración Práctica:

Escenario A (128 MB RAM = 0.125 GB): Tarea de procesamiento matemático tarda **1,000 ms (1.0 s)**.

$$GB - Segundos = 0.125GB \times 1.0s = 0.1250GB - s$$

Escenario B (1,024 MB RAM = 1.0 GB): Se asigna 8 veces más memoria (y por ende 8 veces más potencia de CPU). La misma tarea se completa en **100 ms (0.1 s)**.

$$GB - Segundos = 1.000GB \times 0.1s = 0.1000GB - s$$

Resultado Arquitectónico: El Escenario B procesa la solicitud **10 veces más rápido** y reduce el consumo de GB-Segundos en un **20%**, generando una invocación más rápida y económica.

3.2 Estrategias de Caché de Base de Datos: Amazon ElastiCache (Lazy Loading vs. Write-Through)

Amazon ElastiCache proporciona entornos en memoria compatibles con **Redis** y **Memcached** para descargar bases de datos relacionales (RDS) o NoSQL.

Estrategia Lazy Loading (Cache-Aside)

1. App consulta a ElastiCache.
2. Si es Miss, lee de la BD.
3. App escribe el dato en Caché.

- **Riesgo:** Datos obsoletos (Stale).

Estrategia Write-Through

1. App escribe en la Base de Datos.
2. App escribe de inmediato en Caché.

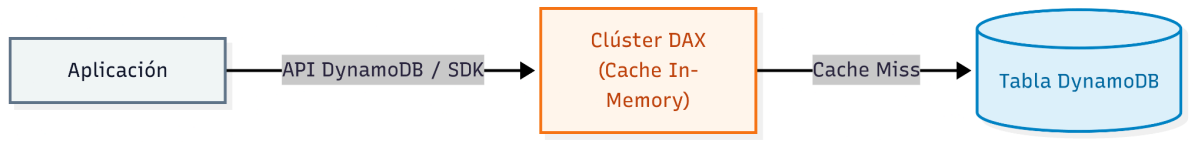
- **Ventaja:** Datos siempre frescos.
- **Desventaja:** Penalización en escritura.

Estrategia de Caché	Flujo de Operación	Ventajas Arquitectónicas	Desventajas y Riesgos	Palabras Clave / Triggers en SAA-C03
Lazy Loading (Cache-Aside)	La aplicación consulta primero a ElastiCache. Si el dato no existe (*Cache Miss*), lo lee de la base de datos, lo escribe en ElastiCache y lo retorna.	Solo se almacenan los datos que realmente se solicitan. Eficiencia de memoria en la caché. Tolera fallos en el nodo de caché.	**Riesgo de datos obsoletos (stale data)** si la BD se actualiza sin invalidar la caché. Penalización de latencia (*penalty*) durante el *Cache Miss*.	*"Read-heavy workloads"*, *"tolerancia a inconsistencia temporal"*, *"minimizar espacio de memoria en caché"*.

Write-Through	La aplicación escribe o actualiza el dato en la base de datos e inmediatamente actualiza/escribe el mismo dato en la caché en memoria.	**Garantía de consistencia de datos.** La caché nunca contiene datos obsoletos respecto a la BD.	Penalización en la latencia de escritura (doble escritura por cada operación). Riesgo de llenar la caché con datos poco solicitados.	*"Datos siempre actualizados", "evitar stale data", "lecturas críticas inmediatamente posteriores a la escritura".*

3.3 Lecturas de Ultra-Baja Latencia con DynamoDB Accelerator (DAX)

Para arquitecturas basadas en Amazon DynamoDB que exigen tiempos de respuesta inferiores al milisegundo, Amazon ElastiCache introduce complejidad innecesaria al requerir lógica explícita en el código de la aplicación. La respuesta arquitectónica especializada es **Amazon DynamoDB Accelerator (DAX)** .



- **Arquitectura de DAX:** Clúster de caché en memoria totalmente administrado, con alta disponibilidad y distribuido, diseñado específicamente para Amazon DynamoDB.
- **Reducción de Latencia:** Reduce el tiempo de respuesta de milisegundos de un solo dígito a **microsegundos (**μs**)** para cargas intensivas de lectura repetitiva e intensiva (*read-heavy, eventual consistency*).
- **Diferenciador Arquitectónico Clave (100% Compatibilidad con la API de DynamoDB):** DAX es **in-line y transparente**. A diferencia de ElastiCache (que requiere reescribir la aplicación para gestionar las llamadas a la caché vía Redis/Memcached SDK), DAX mantiene **estricta compatibilidad con la API de DynamoDB**. Solo se requiere cambiar el *endpoint* de la llamada en el SDK de DynamoDB para apuntar al clúster DAX; la aplicación sigue ejecutando operaciones `GetItem` o `BatchGetItem` idénticas.

3.4 Analítica de Alto Rendimiento en S3: Amazon Athena con Parquet y Particionamiento

Amazon Athena es un servicio de consultas interactivo *serverless* que analiza datos directamente en Amazon S3 utilizando SQL estándar. Athena cobra en función de los **terabytes de datos escaneados** por consulta.

Estrategias de Optimización para Examen:

1. **Formatos de Almacenamiento Columnar (Apache Parquet / ORC):** Convertir archivos de texto plano (CSV, JSON) a formatos columnares como Parquet. Parquet comprime los datos y permite a Athena leer estrictamente las columnas requeridas en la sintaxis SQL `SELECT columna_a`, eliminando la lectura del resto del archivo. Esto reduce el volumen de datos escaneados hasta en un 90%, acelerando la ejecución y disminuyendo drásticamente el costo.

2. **Particionamiento de Datos:** Organizar los datos en S3 mediante estructuras de carpetas/prefijos basados en dimensiones de consulta frecuentes (ej. `s3://mi-bucket/logs/year=2026/month=03/day=15/`). Al ejecutar una consulta con filtros `WHERE year='2026' AND month='03'`, Athena realiza un *pruning* de particiones, escaneando únicamente el prefijo correspondiente e ignorando el resto del bucket.

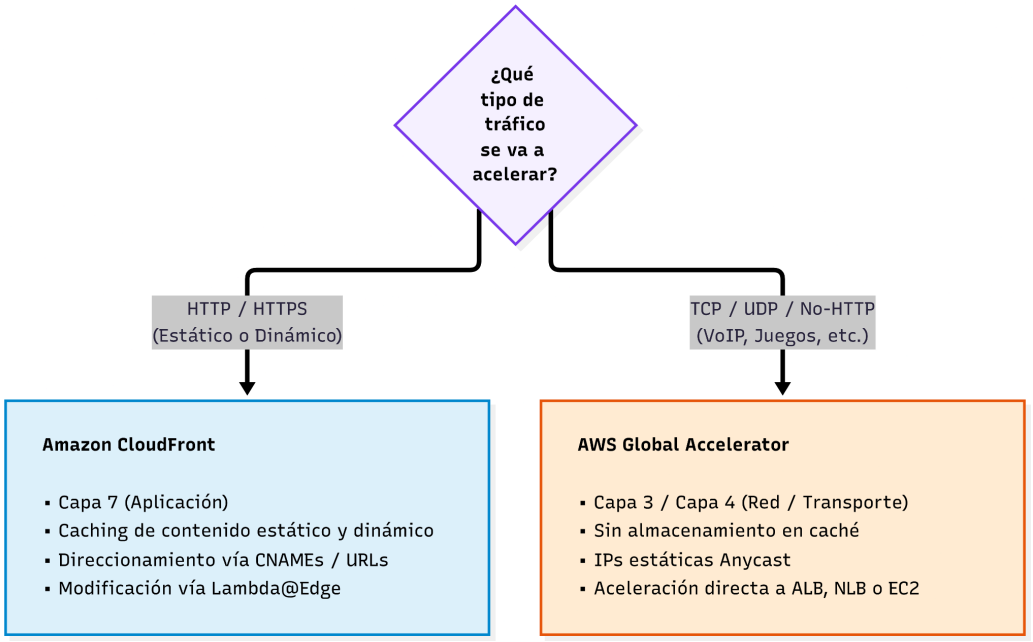
Una vez optimizada la velocidad de respuesta en cómputo y bases de datos, es necesario acelerar la entrega de estas respuestas a escala global.

4. Aceleración Global e Intercepción de Tráfico: AWS Global Accelerator vs. Amazon CloudFront

Contexto Estratégico

Cuando una aplicación atiende a usuarios distribuidos globalmente, el tráfico orientado a Internet sufre degradación por pérdida de paquetes, *jitter* y latencias de enrutamiento WAN. Mitigar estos problemas requiere interceptar el tráfico en el punto de presencia (*Edge Location*) más cercano e ingresarlo inmediatamente a la red privada backbone de alta velocidad de AWS.

4.1 Comparativa Arquitectónica: Global Accelerator vs. CloudFront



Criterio Arquitectónico	Amazon CloudFront	AWS Global Accelerator
Capa del Modelo OSI	**Capa 7** (Capa de Aplicación).	**Capa 3 y Capa 4** (Capas de Red y Transporte).
Tipos de Tráfico Soportados	Tráfico **HTTP / HTTPS** (contenido web estático y dinámico, vídeo HLS/DASH).	Tráfico **TCP y UDP**, telefonía VoIP, juegos en tiempo real, IoT, HTTP/HTTPS no-web.
Mecanismo de Caché	**Edge Caching con TTL:** Almacena copias de contenido estático/dinámico en puntos de presencia global.	**Sin almacenamiento en caché.** Enruta los paquetes de red directamente al origen por la ruta de menor latencia.
Mecanismo de Direccionamiento	Direccionamiento basado en DNS (nombres de dominio CNAME distribuidos).	Proporciona **dos direcciones IP estáticas Anycast** fijas que no cambian.
Capacidad de Cómputo en el Borde	Ejecuta lógica en el borde mediante **CloudFront Functions** o **Lambda@Edge**.	No ejecuta código en el borde; actúa puramente como un acelerador de red y failover entre Regiones.
Orígenes Soportados	Buckets S3, Application Load Balancers (ALB), instancias EC2, servidores HTTP externos.	Network Load Balancers (NLB), Application Load Balancers (ALB), IPs Elásticas, EC2.

4.2 Seguridad y Control de Acceso en Contenido Distribuido (CloudFront OAC y Signed URLs/Cookies)

Asegurar el Origen S3 con Origin Access Control (OAC)

Para garantizar que los usuarios no eludan CloudFront accediendo directamente a las URLs del bucket S3 de origen, se debe implementar **Origin Access Control (OAC)** . OAC firma las solicitudes que pasan desde CloudFront hacia S3 mediante AWS Signature Version 4

(SigV4). El bucket S3 se configura con una política de bucket que autoriza el acceso **únicamente al ARN del Service Principal de CloudFront** , bloqueando cualquier acceso público directo.

Control de Acceso Restringido: Signed URLs vs. Signed Cookies

Cuando la aplicación requiere entregar contenido privado o de pago (ej. videos bajo suscripción o descargas de software) a usuarios autenticados:

Signed URLs (URLs Firmadas):

- Otorga acceso a **un único archivo o recurso individual** .
- Adecuado para clientes que no soportan cookies HTTP (ej. aplicaciones cliente móviles nativas o reproductores multimedia integrados).
- *Ejemplo de uso:* Generar una URL con tiempo de expiración para descargar un archivo `.pdf` específico.

Signed Cookies (Cookies Firmadas):

- Otorga acceso a **múltiples archivos o catálogos completos** con una sola validación.
- No requiere modificar la estructura de las URLs en el código HTML de la aplicación.
- *Ejemplo de uso:* Permitir el acceso a todo el contenido de video dentro del directorio privado `s3://mi-bucket/cursos-premium/*` para usuarios suscritos.

Con estos patrones de aceleración global y seguridad de borde consolidados, revise la matriz de decisión y las trampas del examen para afianzar el criterio de selección.

5. Matriz Sintética de Decisiones, Palabras Clave y Trampas Frecuentes del Examen

Contexto Estratégico

El examen SAA-C03 presenta múltiples opciones plausiblemente correctas. La capacidad de descarte rápido radica en identificar las "palabras clave de activación" (*triggers*) que eliminan distractores sobreingenierizados o costosos.

5.1 Matriz de Decisión Instantánea por Escenario

Requerimiento del Cliente / Caso de Uso	Servicio / Configuración Ganadora	Razón Arquitectónica Clave
Base de datos relacional con latencia sensible de I/O y garantía estricta de rendimiento.	**EBS io2 / io2 Block Express**	Latencia constante de milisegundo de un solo dígito / sub-milisegundo con IOPS garantizados independientes del tamaño.

Aplicación transaccional con presupuesto ajustado que requiere escalar IOPS sin comprar disco extra.	**EBS gp3**	Asigna 3,000 IOPS y 125 MB/s gratis base. Permite escalar IOPS/MB/s de forma independiente sin agrandar el volumen.
Escalado dinámico de almacenamiento en disco en producción sin tiempo de inactividad.	**EBS Elastic Volumes (ModifyVolume API)**	Permite cambiar el tipo de volumen, capacidad o IOPS en vivo con *zero-downtime*.
Almacenamiento compartido NFSv4 accesible por más de 50 instancias EC2 en múltiples AZs.	**Amazon EFS (General Purpose / Max I/O)**	Sistema de archivos compartido distribuido, compatible con POSIX y acceso concurrente Multi-AZ.
Caché temporal o procesamiento de buffers con la menor latencia de I/O posible (μs).	**EC2 Instance Store**	Bloque físico directamente adjunto al hardware de la instancia EC2. *Atención: Volátil / Efímero*.
Aceleración de subidas masivas de datos hacia S3 desde clientes dispersos globalmente.	**S3 Transfer Acceleration**	Encauza el tráfico por las Edge Locations de CloudFront hacia la red backbone privada de AWS sin hacer caché de objeto.
Optimización costo-rendimiento en función serverless con carga intensiva de CPU.	**Incrementar RAM en AWS Lambda**	La asignación proporcional de CPU acorta el tiempo de ejecución, reduciendo los GB-Segundos facturados.
Descargar lecturas repetitivas en BD relacional garantizando datos sin obsolescencia.	**ElastiCache con estrategia Write-Through**	Escribe simultáneamente en BD y en la caché, garantizando consistencia y evitando *stale data*.
Reducir latencia de DynamoDB a microsegundos sin modificar el código de la aplicación.	**Amazon DynamoDB Accelerator (DAX)**	Caché en memoria transparente e *in-line* que mantiene **100% de compatibilidad con la API/SDK de DynamoDB**.
Consultas SQL analíticas sobre archivos masivos en S3 minimizando costo por TB escaneado.	**Amazon Athena + Parquet + Particionamiento**	Formato columnar y *pruning* de prefijos que reducen drásticamente los GB/TB escaneados durante el `SELECT`.

Aceleración de tráfico de juegos en tiempo real basado en UDP con IPs estáticas en el cliente.	**AWS Global Accelerator**	Enruta tráfico de Capa 3/4 por la red privada de AWS utilizando dos IPs estáticas Anycast sin hacer caché.
Distribución de sitio web estático/dinámico sobre HTTPS protegiendo el origen S3 de accesos públicos.	**Amazon CloudFront + S3 con Origin Access Control (OAC)**	Caché en el borde (Capa 7) firmando las peticiones al origen S3 vía OAC para bloquear accesos directos.
Acceso temporal restringido a un catálogo entero de archivos privados para usuarios autenticados.	**CloudFront con Signed Cookies**	Permite dar acceso a múltiples recursos bajo una ruta privada sin alterar las URLs en la aplicación.

5.2 Trampas Comunes y Distractores en el Examen SAA-C03

- ***Confundir la persistencia de EC2 Instance Store:*** *Distractor habitual que propone Instance Store para bases de datos principales o datos que deben sobrevivir al reinicio o detención de la instancia EC2. Instance Store es efímero; si la instancia se detiene (Stop), los datos se pierden permanentemente.*

- ***Proponer CloudFront para protocolos no-HTTP (UDP/TCP en tiempo real):*** *Seleccionar CloudFront para acelerar tráfico de juegos basado en UDP o telefonía VoIP. CloudFront opera exclusivamente en Capa 7 (HTTP/HTTPS); para tráfico UDP/TCP de baja latencia o IPs estáticas Anycast se debe seleccionar Global Accelerator.*

- ***Seleccionar ElastiCache en lugar de DAX para DynamoDB:*** *Proponer ElastiCache cuando el enunciado menciona explícitamente "minimizar el esfuerzo de desarrollo" o "mantener compatibilidad transparente sin reescribir la lógica de la aplicación". ElastiCache requiere implementar lógica de caché en el código; DAX es 100% compatible con la API de DynamoDB sin cambiar código.*

- ***Sobreaprovisionar con volúmenes EBS io2 :*** *Elegir volúmenes `io2` cuando un volumen SSD `gp3` satisface el requisito a una fracción del costo. Recuerde que gp3 entrega 3,000 IOPS base gratis e independientes del tamaño.*

- ***S3 Transfer Acceleration vs. CloudFront Edge Caching para cargas:*** *Confundir la función de Transfer Acceleration asumiendo que almacena los objetos subidos en caché. Transfer Acceleration usa la Edge Location como punto de entrada a la red privada de AWS, no como almacenamiento persistente o caché de lectura.*

- ***Uso incorrecto de Signed URLs frente a Signed Cookies:*** *Seleccionar Signed URLs para proteger cientos de archivos multimedia dentro de un área privada de miembros. Obligar a la aplicación a generar URLs firmadas individuales para cada recurso*

introduce una sobrecarga masiva de procesamiento; la respuesta correcta para múltiples archivos o directorios es Signed Cookies.

- ***Confundir Réplicas de Lectura (Read Replicas) con Multi-AZ en RDS:*** *Proponer un despliegue RDS Multi-AZ para resolver problemas de rendimiento en lecturas. RDS Multi-AZ es una solución de Alta Disponibilidad y Recuperación ante Desastres (escritura síncrona pasiva); las Read Replicas son la única solución para escalar cargas de trabajo intensivas de lectura.*

Conclusión y Mentalidad de Arquitecto

Superar el Dominio 3 del examen SAA-C03 exige asumir el rol de un *Principal Solutions Architect*: **priorizar invariablemente la arquitectura que cumpla con los SLAs de rendimiento requeridos al menor costo y con la menor complejidad operativa**. Al enfrentarse a las preguntas de escenario, aísle primero dónde reside el cuello de botella (disco, red, base de datos o cómputo), descarte los distractores que violan las restricciones de persistencia o protocolo, y elija la combinación de servicios desacoplados que resuelva el problema de forma elegante, escalable y costo-eficiente.

Dominio 4 - Diseño de Arquitecturas Optimizadas en Costos (AWS SAA-C03)

1. Visión General Estratégica del Dominio 4 (20% del Examen SAA-C03)

En la evaluación para la certificación **AWS Certified Solutions Architect - Associate (SAA-C03)** , el **Dominio 4: Diseño de Arquitecturas Optimizadas en Costos** representa de forma estricta el **20% del contenido evaluado** . En el marco de **AWS Well-Architected Framework** (específicamente en el Pilar de Optimización de Costos), la reducción de gastos no es una meta aislada que deba lograrse a expensas del rendimiento o de la resiliencia. Por el contrario, este pilar exige tomar decisiones de arquitectura ponderadas que alineen los costos operativos con los objetivos del negocio. La optimización de costos en AWS implica seleccionar de manera continua las clases de recursos más costo-efectivas, ajustar la oferta a la demanda en tiempo real (*Matching Supply and Demand*), eliminar el sobredimensionamiento sin comprometer la alta disponibilidad y mantener una visibilidad total sobre los componentes del gasto (*Expenditure and Usage Awareness*).

Estructura del Examen y Metodología de Puntuación

Para garantizar una preparación rigurosa, el aspirante debe entender con precisión la mecánica de evaluación establecida en la guía oficial de AWS:

Composición y Unscored Content: El examen consta de **65 preguntas en total** . De ellas, **50 son preguntas puntuadas** y **15 son preguntas no puntuadas** (*unscored*). Las 15 preguntas no puntuadas se distribuyen aleatoriamente sin identificación explícita para recopilar métricas estadístico-operativas antes de su aprobación formal.

Tipos de Preguntas:

- *Opción Múltiple:* Presenta una respuesta correcta y tres opciones incorrectas (*distractors*).

- *Respuesta Múltiple:* Presenta dos o más respuestas correctas entre cinco o más opciones de respuesta.

Escala de Puntuación Compensatoria: Los resultados se calculan mediante un modelo de puntuación escalada que oscila entre **100 y 1.000 puntos** , siendo el **mínimo para aprobar 720 puntos** . Dado que la metodología es compensatoria, no se exige alcanzar un puntaje mínimo individual en cada dominio, sino acumular el puntaje ponderado global. No existe penalización por respuestas incorrectas (*no penalty for guessing*), por lo que responder la totalidad de los reactivos es tácticamente obligatorio.

Análisis Teórico de Distractores (*Distractors*): Los distractores en el Dominio 4 son funcionalmente viables, pero técnicamente ineficientes o financieramente subóptimos. El examinador suele atraer al candidato mediante opciones que resuelven el requisito técnico utilizando servicios sobredimensionados, seleccionando modelos de compra sin compromiso para cargas constantes o ignorando los costos ocultos por transacciones y tarifas de recuperación de datos.

Marco de Razonamiento Arquitectónico

Para resolver los escenarios del examen con precisión de arquitecto principal, aplique la siguiente metodología sistemática de decisión:

Requerimiento ➔ Problema ➔ Servicio de AWS ➔ Patrón de Arquitectura ➔ Mejor Respuesta

1. **Requerimiento:** Identificar los parámetros operativos del enunciado (ej. picos impredecibles de tráfico, retención a largo plazo o trabajos por lotes tolerantes a fallos).

2. **Problema:** Aislar la ineficiencia financiera (ej. recursos ociosos en estado estable, sobredimensionamiento de almacenamiento de bloques o costos fijos de base de datos).

3. **Servicio de AWS:** Seleccionar los componentes de cómputo, almacenamiento o base de datos capaces de cubrir la necesidad técnica primaria.

4. **Patrón de Arquitectura:** Integrar los mecanismos de escalado automático, clases de almacenamiento, modelos de reserva o capas de caché.

5. **Mejor Respuesta:** Filtrar la opción que proporcione el menor costo total de operación (*Total Cost of Ownership*) cumpliendo con las restricciones de disponibilidad y rendimiento.

Walkthrough Práctico del Marco de Razonamiento

Considere un escenario de examen donde una aplicación de comercio electrónico experimenta picos imprevistos de lectura en su base de datos relacional PostgreSQL en

Amazon RDS durante promociones relámpago, elevando la utilización de CPU del nodo principal al 95% y generando latencias elevadas:

- **Requerimiento:** Eliminar los cuellos de botella de lectura y reducir la CPU de la base de datos sin incurrir en caídas de servicio.
- **Problema:** Aumentar la clase de instancia de RDS de forma permanente para absorber picos temporales genera un gasto fijo excesivo durante el 90% del tiempo en que el tráfico es normal.
- **Servicio de AWS:** Amazon ElastiCache (Redis / Memcached) o RDS Read Replicas.
- **Patrón de Arquitectura:** Configurar una capa de almacenamiento en caché en memoria con Amazon ElastiCache frente a Amazon RDS para responder las consultas frecuentes de lectura.
- **Mejor Respuesta:** Implementar Amazon ElastiCache para absorber hasta el 90% de las lecturas repetitivas, permitiendo mantener la instancia principal de Amazon RDS en un tamaño reducido y costo-eficiente, en lugar de escalar la instancia primaria de base de datos a un nivel superior.

Este marco mental de decisión establece el criterio analítico necesario para evaluar los modelos de adquisición de cómputo en AWS.

2. Modelos y Estrategias de Cómputo Costo-Efectivos

El aprovisionamiento de capacidad de cómputo representa el rubro operativo de mayor volumen financiero en la infraestructura cloud. Alinear de forma precisa el modelo de adquisición de Amazon EC2 o de las plataformas *serverless* con la variabilidad del tráfico es la regla fundamental para evitar el sobrepago por capacidad ociosa.

Evaluación Comparativa de Modelos de Compra de EC2

Modelo de Compra	Mecánica de Cobro y Compromiso	Flexibilidad (Región, Familia, SO, Tenencia)	Porcentaje de Ahorro	Justificación Arquitectónica (Uso Idóneo)
On-Demand	Pago por segundo o por hora sin compromiso contractual ni pagos por adelantado.	**Máxima:** Modificable en cualquier momento sin restricciones de región o tipo.	**0%** (Precio base).	Cargas de trabajo de corta duración, impredecibles, o fases iniciales de desarrollo donde no se han establecido patrones de uso.

Reserved Instances (RIs)	Compromiso formal de 1 o 3 años para una capacidad de cómputo específica.	**Baja:** Vinculado a parámetros de instancia, región y plataforma (con variaciones en RIs convertibles).	**Hasta el 72%** en compromisos a 3 años.	Cargas de trabajo de estado estable (*steady-state*), predecibles e ininterrumpidas a largo plazo.
Compute Savings Plans	Compromiso monetario ($/hora) por 1 o 3 años en cómputo global.	**Extrema:** Aplica automáticamente entre familias de EC2, regiones, SO, tenencias, **Fargate** y **AWS Lambda**.	**Hasta el 66%** según el compromiso.	Arquitecturas dinámicas y cambiantes que evolucionan entre servidores tradicionales, contenedores Fargate y funciones Lambda.
Instance Savings Plans	Compromiso monetario ($/hora) por 1 o 3 años asignado a una familia en una región.	**Media:** Limitado a una familia de instancias específica dentro de una región (ej. `m5` en `us-east-1`), pero flexible en tamaño, SO y tenencia.	**Hasta el 72%** (descuento superior a Compute SP).	Aplicaciones consolidadas cuya familia de instancias y región de despliegue no cambiarán durante el ciclo del compromiso.
Spot Instances	Subasta de capacidad sobrante de EC2 en la infraestructura global de AWS.	**Variable:** Sujeto a la disponibilidad de capacidad en las Zonas de Disponibilidad.	**Hasta el 90%** en comparación con On-Demand.	Cargas tolerantes a fallos, procesamiento por lotes (*AWS Batch*), análisis masivo de datos o contenedores interrumplibles con aviso de 2 minutos.

Estrategia de Spot Instances con Auto Scaling Target Tracking

Para reducir drásticamente el costo de cómputo manteniendo la disponibilidad de la aplicación, el patrón arquitectónico óptimo consiste en configurar grupos de **Amazon EC2**

Auto Scaling que combinen instancias Spot y On-Demand, guiados por políticas de escalado por seguimiento de objetivos (*Target Tracking*).

- **Procesamiento por Lotes con AWS Batch:** **AWS Batch** administra de forma nativa la ejecución de trabajos sobre entornos de cómputo compuestos por instancias Spot. AWS Batch gestiona las colas de trabajo y reintenta automáticamente los trabajos en caso de que una instancia Spot sea recuperada por AWS mediante la notificación de interrupción de 2 minutos.

- **Orquestación de Contenedores en ECS y EKS:** En clústeres de **Amazon ECS** o **Amazon EKS** , implemente la estrategia *Capacity Providers* para combinar un núcleo de capacidad base en instancias On-Demand con un escalado horizontal elástico apoyado en Spot Instances. Los microservicios sin estado (*stateless*) o con soporte para terminación elegante absorben las interrupciones redirigiendo el tráfico sin causar caídas de servicio.

Cómputo Serverless y Abstracción de Infraestructura

Las arquitecturas *serverless* trasladan la responsabilidad operativa del aprovisionamiento a AWS, eliminando los pagos por servidores ociosos.

AWS Lambda: Opera bajo el principio estricto de pago por uso. No existen cobros cuando la función no está en ejecución. La facturación se calcula combinando el número de peticiones con la duración de ejecución. Para fines de facturación en el examen, la duración se mide en **incrementos de 100 milisegundos** .

- *Optimización con Lambda Layers:* Permite empaquetar bibliotecas, dependencias o motores de ejecución personalizados en capas inmutables (hasta **5 capas por función**), reduciendo el tamaño del paquete de despliegue y mejorando el tiempo de inicio.

- *Capa Gratuita:* Incluye **1 millón de peticiones gratuitas al mes** y **400.000 GB-segundos** de tiempo de cómputo mensual.

AWS Fargate: Proporciona un motor de ejecución *serverless* para contenedores en **Amazon ECS** y **Amazon EKS** . Elimina la necesidad de administrar instancias EC2 subyacentes. Se cobra estrictamente por los recursos de **vCPU y memoria consumidos** desde que se inicia la descarga de la imagen del contenedor hasta que el *pod* o tarea se detiene.

- *Fargate Savings Plans:* Permitir adquirir compromisos financieros a 1 o 3 años para reducir los costos de Fargate en **hasta un 50%** .

AWS Elastic Beanstalk: Servicio administrado de Plataforma como Servicio (PaaS) que simplifica el despliegue de aplicaciones en entornos *Web Tier* y *Worker Tier* (impulsados por **Amazon SQS**). **Elastic Beanstalk no tiene costo adicional por el servicio en sí** ; la facturación deriva exclusivamente de los recursos subyacentes aprovisionados (instancias EC2, balanceadores ELB o bases de datos RDS).

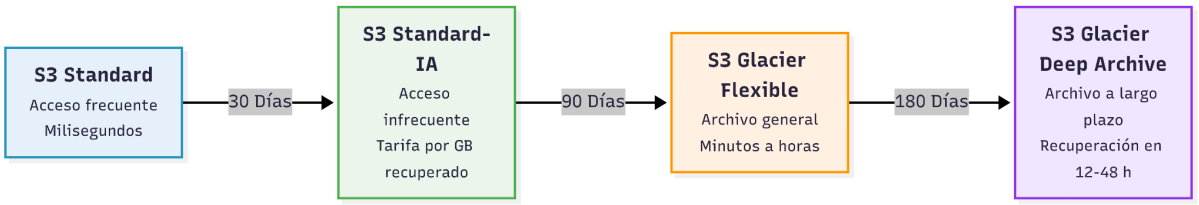
La optimización del cómputo debe complementarse con una gestión rigurosa del almacenamiento persistente de bloques y objetos.

3. Optimización Avanzada del Almacenamiento (Amazon S3 y EBS)

A diferencia del cómputo —donde apagar un servidor detiene el cobro de procesamiento—, el almacenamiento mal gestionado genera costos acumulativos silenciosos. Diseñar políticas adecuadas para el ciclo de vida de los datos es vital para garantizar la sostenibilidad económica en la nube.

Estrategias en Amazon S3 y Análisis Comparativo de Clases

Amazon S3 ofrece 7 clases de almacenamiento diseñadas para adaptarse a la frecuencia de acceso y la durabilidad requerida:



1. **S3 Standard:** Diseñado para datos de acceso frecuente. Ofrece alta disponibilidad (99,99%) y durabilidad (99,999999999% a lo largo de ≥ 3 Zonas de Disponibilidad). No exige permanencia mínima ni cobros por recuperación.

2. **S3 Standard-IA (Infrequent Access):** Orientado a datos con menor frecuencia de consulta que requieren acceso inmediato en milisegundos. Ofrece costos de almacenamiento sustancialmente menores por GB, pero exige un mínimo de **30 días de permanencia** y aplica **tarifas por recuperación de datos** por GB.

3. **S3 One Zone-IA:** Almacena los datos en una sola Zona de Disponibilidad (disponibilidad del 99,50%). Es un **20% más económico que Standard-IA**; idóneo para copias de seguridad secundarias o datos fácilmente reproducibles. Exige **30 días de permanencia mínima** y aplica tarifas por recuperación.

4. **S3 Intelligent-Tiering:** La respuesta idónea en el examen para **patrones de acceso desconocidos, cambiantes e impredecibles**. Monitoriza y mueve objetos automáticamente entre niveles de acceso Frecuente e Infrecuente. **No aplica tarifas por recuperación de datos** ni cobros por transiciones operativas, únicamente una tarifa mínima de monitorización y un período mínimo de retención de **30 días**.

5. **S3 Glacier Instant Retrieval:** Diseñado para datos de archivo a los que se accede una vez por trimestre pero que **requieren un tiempo de respuesta de milisegundos**. Ofrece costos de almacenamiento muy bajos, exige un mínimo de **90 días de permanencia** y aplica tarifas de recuperación por GB superiores a Standard-IA.

6. **S3 Glacier Flexible Retrieval (anteriormente S3 Glacier):** Orientado al archivo de datos donde el acceso en tiempo real no es necesario. Ofrece tres opciones de recuperación: *Expedited* (1 a 5 minutos), *Standard* (3 a 5 horas) y *Bulk* (5 a 12 horas). Exige un mínimo de **90 días de permanencia**.

7. **S3 Glacier Deep Archive:** La clase de menor costo en AWS, diseñada para el cumplimiento normativo y la retención a largo plazo (7 a 10 años). Los tiempos de recuperación oscilan entre 12 horas (*Standard*) y 48 horas (*Bulk*). Exige una **permanencia mínima de 180 días**.

Configuración Estratégica de S3 Lifecycle

Mediante las reglas de **S3 Lifecycle** , es posible automatizar la transición entre clases y programar la eliminación definitiva de objetos. Como regla técnica del examen: *un objeto debe cumplir el tiempo mínimo de permanencia en una clase antes de transicionar a la siguiente* (ejemplo: 30 días en Standard antes de pasar a Standard-IA, 90 días para Glacier y 180 días para Deep Archive).

Análisis de Costos Ocultos y Optimización en Amazon EBS

La Trampa de Facturación Oculta en Amazon EC2 y EBS

La trampa conceptual más común del examinador consiste en asumir que apagar (*Stop*) una instancia EC2 durante fuera de horario detiene por completo su costo. La realidad técnica es: **la instancia no cobra por vCPU ni memoria mientras está detenida, pero el volumen EBS adjunto continúa cobrando el 100% de la capacidad provisionada (GB/mes)** , además de las IOPS adicionales provisionadas y el almacenamiento de *snapshots* en S3.

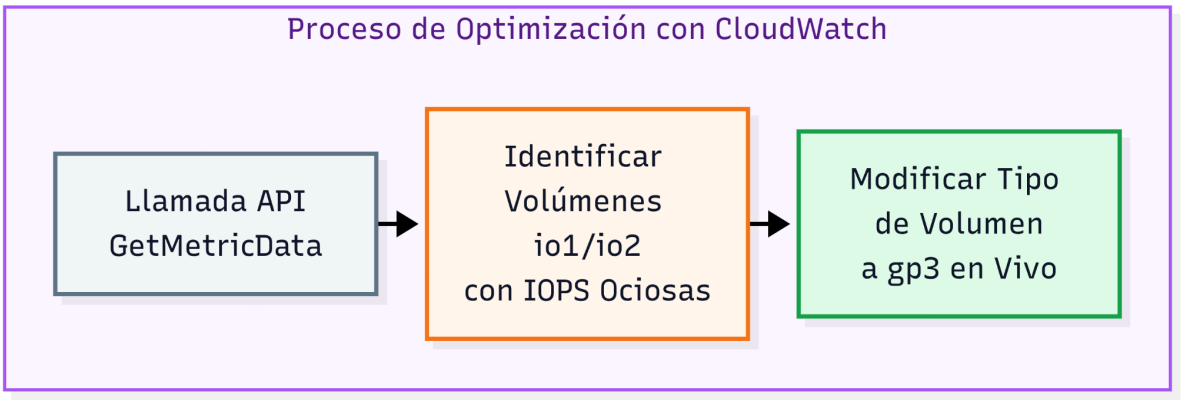
Migración de EBS io1/io2 a gp3 mediante CloudWatch

Para maximizar la eficiencia financiera en almacenamiento de bloques:

- **Límite de gp2:** Los volúmenes `gp2` escalan sus IOPS de forma lineal según el tamaño del volumen (**3 IOPS/GB**, con un rendimiento básico de 100 IOPS hasta un máximo de 16.000 IOPS). Para obtener más IOPS en `gp2`, el usuario debe comprar más capacidad en GB aunque no la necesite.

- **Límite de io1/io2:** Los volúmenes de IOPS Aprovisionadas (`io1`/`io2`) ofrecen un rendimiento garantizado para bases de datos críticas, pero cobran de forma independiente la capacidad asignada y cada IOPS aprobada, resultando muy costosos.

- **Ventaja de gp3:** Proporciona de forma gratuita un rendimiento base de **3.000 IOPS y 125 MB/s de throughput**, independientemente del tamaño del volumen. `gp3` permite aprovisionar IOPS y *throughput* de forma totalmente independiente del tamaño en GB, ofreciendo un costo por GB sensiblemente inferior.



Para ejecutar la migración, utilice la API `GetMetricData` de **Amazon CloudWatch** para evaluar las métricas `VolumeReadOps` y `VolumeWriteOps`. Si el análisis confirma que las IOPS reales consumidas están por debajo del umbral de 3.000 IOPS, reconfigure el volumen de `io1`/`io2` a `gp3` sin interrupción del servicio mediante la función *EBS Elastic Volumes*.

Visión Comparativa de Almacenamiento Complementario

- **Amazon EFS Infrequent Access (EFS-IA):** Reduce los costos de almacenamiento en **Amazon EFS** para sistemas de archivos de acceso esporádico ($0,025/GB-mes). Mediante **EFS Lifecycle Management**, los archivos no leídos en un período determinado (ej. 30 días) se trasladan automáticamente a EFS-IA.

- **Amazon FSx for Windows File Server y FSx for Lustre:** FSx for Windows admite almacenamiento SMB administrado con soporte de deduplicación de datos. **FSx for Lustre** proporciona rendimiento de alto nivel para analítica; para tareas temporales de procesamiento masivo, es posible seleccionar sistemas de archivos no replicados (*scratch file systems*) para reducir los costos al mínimo.

- **AWS Snowball y AWS Storage Gateway: AWS Snowball** evita costos de transferencia de red en migraciones petagráficas masivas. **AWS Storage Gateway** extiende el almacenamiento local a la nube, optimizando retenciones mediante volúmenes almacenados en S3 y EBS.

4. Arquitectura de Bases de Datos Costo-Eficientes

El diseño de bases de datos exige evitar el sobredimensionamiento de instancias relacionales. Alinear el motor de base de datos con la fluctuación de la carga elimina el gasto fijo en entornos con tráfico variable.

Amazon Aurora Serverless (v1 vs. v2) para Cargas Intermitentes

Amazon Aurora Serverless elimina el costo continuo de instancias aprovisionadas RDS Multi-AZ tradicionales en entornos con uso intermitente o impredecible, ajustando automáticamente las **Unidades de Capacidad de Aurora (ACUs)** .

Característica	Amazon Aurora Serverless v1	Amazon Aurora Serverless v2
Mecánica de Escalado	Escala duplicando o dividiendo a la mitad las ACUs. Requiere puntos de escalado (*cooldown periods*).	Escala instantáneamente en **fracciones de ACU** (hasta 0.5 ACUs) de forma fluida y continua.
Pausa a Cero (Scale to 0)	**Soportado:** Puede pausarse totalmente a 0 ACUs tras períodos de inactividad (elimina cobro de cómputo).	**No soporta pausa a 0:** Escala hasta un mínimo de 0.5 ACUs para mantener la disponibilidad.
Disponibilidad y Multi-AZ	Diseñado primordialmente para entornos de desarrollo, pruebas o cargas no críticas.	**Soporta Multi-AZ y Read Replicas:** Diseñado para cargas de producción críticas y transaccionales.
Conexiones Activas	Las modificaciones de escala pueden interrumpir transacciones de larga duración si no hay puntos de pausa.	Escala de forma transparente sin interrumpir conexiones o transacciones activas.

La regla de examen determina: *Aurora Serverless v1 es óptimo para entornos Dev/Test que requieren pausarse totalmente a cero. Aurora Serverless v2 es la opción adecuada para producción empresarial con tráfico altamente volátil que exige Multi-AZ e instantaneidad*.

Comparativa y Selección de Modelos en Relacionales y NoSQL

Amazon DynamoDB: Provisioned Capacity vs. On-Demand

- **Provisioned Capacity Mode (con Auto Scaling):** Exige definir las Unidades de Capacidad de Lectura y Escritura (RCU y WCU). Es la alternativa **más económica para cargas de trabajo estables, predecibles o con crecimiento gradual** , especialmente cuando se combina con reservas de capacidad.

- **On-Demand Capacity Mode:** Cobra por cada petición de lectura/escritura realizada. Es la solución adecuada para aplicaciones con patrones de tráfico desconocidos, spikes imprevistos o volúmenes de transacciones muy bajos.

- **Eficiencia en Índices:** Limitar la creación de Índices Secundarios Globales (GSI) e Índices Secundarios Locales (LSI) a los atributos estrictamente necesarios evita la duplicación de escrituras y reduce el consumo global de WCUs.

RDS Multi-AZ, Read Replicas y ElastiCache

- **RDS Multi-AZ vs. Read Replicas:** Un despliegue **Multi-AZ** replica datos de forma síncrona en una segunda Zona de Disponibilidad con el objetivo exclusivo de proporcionar **alta disponibilidad y conmutación por error (** ***failover*** **)** ; el nodo secundario no puede procesar tráfico de lectura. Las **Read Replicas** utilizan replicación asíncrona para **descongestionar las lecturas del nodo primario** , evitando tener que escalar la instancia de base de datos a un tamaño superior.

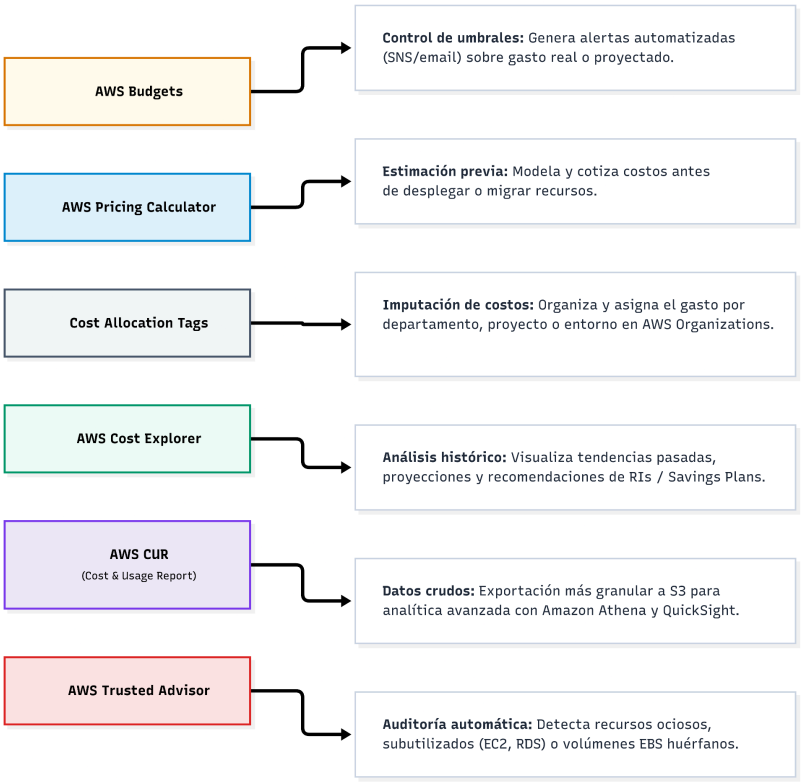
- **Amazon ElastiCache:** Colocar una capa de memoria (*in-memory*) con **ElastiCache (Redis o Memcached)** frente a una base de datos RDS permite almacenar en caché las consultas más recurrentes, reduciendo la carga de IOPS y CPU en la base de datos subyacente y permitiendo reducir el tamaño de la instancia relacional.

5. Herramientas de Gobernanza, Control y Análisis de Gastos

La optimización continua en AWS exige la implementación de mecanismos de visibilidad, presupuestos proactivos y categorización estricta de los recursos.

Desglose Analítico de Servicios de Control Financiero

Un Arquitecto de Soluciones debe dominar las siguientes herramientas para garantizar el control sobre los costos de la organización:



- **AWS Cost Explorer:** Herramienta visual para analizar el historial de gastos de los últimos 12 meses, predecir costos para los próximos 12 meses y revisar recomendaciones para la compra de Reserved Instances y Savings Plans.

- **AWS Cost & Usage Reports (CUR):** Proporciona el nivel de detalle más alto disponible sobre datos de costos y uso. Exporta informes detallados a un bucket de **Amazon S3** en formato CSV o Parquet para análisis mediante Amazon Athena.

- **AWS Budgets:** Permite configurar presupuestos personalizados y enviar **alertas automáticas (vía e-mail o Amazon SNS)** cuando los costos o el uso **reales o proyectados** superan los umbrales fijados. *Atención en el examen:* AWS Budgets es la herramienta idónea para la prevención proactiva antes de que ocurra el exceso de gasto.

- **Cost Allocation Tags (Etiquetas de Asignación de Costos):** Parámetros clave-valor (ejemplo: CostCenter: 8042, Environment: Production) adjuntos a los recursos. Al activarse desde la consola de facturación, permiten desglosar el gasto e imputar costos departamentales a través de **AWS Organizations** .

- **AWS Trusted Advisor:** Herramienta de auditoría automatizada que identifica oportunidades inmediatas de ahorro escaneando:
    - Instancias EC2 con bajo uso de CPU (≤ 10%).
    - Direcciones IP Elásticas (*Elastic IP*) no asociadas a instancias en ejecución.
    - Volúmenes Amazon EBS no adjuntos (*unattached EBS volumes*).
    - Clústeres de Amazon DB ociosos.

- **AWS Pricing Calculator:** Herramienta de modelado financiero basada en web utilizada para estimar los costos esperados de un modelo arquitectónico antes de su despliegue en producción.

6. Lógica de Decisión, Palabras Clave y Trampas Comunes en el Examen Real

Para maximizar el rendimiento en el examen SAA-C03, el estudiante debe desarrollar la habilidad de conectar de forma inmediata las expresiones clave de los escenarios con su solución arquitectónica optimizada.

Tabla de Mapeo: Palabras Clave vs. Solución Arquitectónica Optimizada

Frase o Patrón de Requisito en el Examen	Solución Arquitectónica Optimizada (Dominio 4)
"Patrón de acceso a datos desconocido, cambiante o impredecible sin impacto operativo"	**Amazon S3 Intelligent-Tiering**

"Cargas de trabajo por lotes (batch) o procesamientos tolerantes a interrupciones al menor costo"	**EC2 Spot Instances** integradas con **AWS Batch** o ECS capacity providers.
"Carga de trabajo continua en estado estable (steady-state) durante un período de 1 a 3 años"	**Compute Savings Plans** o **Instance Savings Plans** / **Reserved Instances**.
"Base de datos de desarrollo/pruebas de uso intermitente que debe detener el cobro de cómputo sin tráfico"	**Amazon Aurora Serverless v1** (configurada con pausa a cero ACUs).
"Base de datos de producción con tráfico impredecible que exige alta disponibilidad y Multi-AZ"	**Amazon Aurora Serverless v2**
"Notificar proactivamente a los administradores antes de que los costos proyectados superen el límite"	**AWS Budgets** (configurando alertas basadas en previsiones proyectadas).
"Identificar automáticamente volúmenes EBS no adjuntos e IP elásticas no asociadas"	**AWS Trusted Advisor**
"Optimizar el costo de un volumen EBS manteniendo 3.000 IOPS sin modificar el tamaño en GB"	Identificar con `GetMetricData` de CloudWatch y **migrar de io1/io2 a gp3**.
"Desplegar código de aplicación web sin administrar infraestructura ni pagar por herramientas de despliegue"	**AWS Elastic Beanstalk** (el servicio PaaS es gratuito, solo cobra recursos subyacentes).

Análisis de Trampas Comunes y Distractores en el Examen

La Falsa Economía de Detener Instancias EC2 (*Stopped Instances*):

- *El Distractor:* Elegir una solución que promete detener instancias EC2 durante los fines de semana para "eliminar el 100% de los costos del sistema".

- *Mecanismo de la Trampa Financiera:* Detener la instancia interrumpe el cobro de vCPU y memoria, pero los **volúmenes EBS adjuntos continúan facturándose al 100% de la capacidad provisionada (GB/mes)** , sumado a las IOPS aprovisionadas y al almacenamiento de los *snapshots* en S3.

La Trampa de Uso Incorrecto de S3 Standard-IA:

- *El Distractor:* Seleccionar S3 Standard-IA para almacenar datos temporales que se eliminan o sobreescriben a los 10 días de su creación.

- *Mecanismo de la Trampa Financiera:* S3 Standard-IA impone un **cobro mínimo obligatorio de 30 días de retención** . Si un objeto se elimina al día 10, AWS facturará los 20 días restantes como penalización por eliminación anticipada, sumado a la tarifa por recuperación de datos por GB, resultando más costoso que mantener los datos en S3 Standard.

Uso de Instancias On-Demand para Aplicaciones en Estado Estable:

- *El Distractor:* Seleccionar On-Demand para la infraestructura base de una aplicación corporativa que operará de forma ininterrumpida durante 3 años.

- *Mecanismo de la Trampa Financiera:* On-Demand proporciona la mayor flexibilidad pero al precio más alto. Ignorar los compromisos de **Savings Plans o Reserved Instances** para cargas estables descuida el requisito de optimización de costos.

Confusión entre AWS Trusted Advisor y AWS Budgets:

- *El Distractor:* Seleccionar AWS Trusted Advisor para enviar alertas preventivas por e-mail cuando la proyección mensual de facturación supere un límite prefijado.

- *Mecanismo de la Trampa Financiera:* Trusted Advisor escanea y recomienda optimizaciones sobre recursos ya desplegados u ociosos. La herramienta diseñada para supervisar umbrales financieros y emitir alertas proactivas sobre costos reales o **proyectados** es **AWS Budgets** .

Resumen Final de Reglas de Oro para el Examen

1. **Automatización sobre Procesos Manuales:** En el Dominio 4, las soluciones administradas y automáticas (*S3 Intelligent-Tiering*, *Auto Scaling Target Tracking*, *Aurora Serverless*) son arquitectónicamente superiores a la ejecución de scripts manuales o tareas programadas en *Cron*.

2. **Priorice la Eliminación del Desperdicio:** Todo recurso ocioso (IPs elásticas sin asociar, volúmenes EBS sin adjuntar o bases de datos sobredimensionadas) debe identificarse mediante **AWS Trusted Advisor** y métricas de **CloudWatch**.

3. **Pague Sólo por el Uso Real:** Para cargas tolerantes a fallos, elija **Spot Instances**; para cargas intermitentes, opte por **Serverless (Lambda/Fargate)**; y para cargas estables, aplique **Savings Plans**.