**Preguntas Conceptuales Cortas**

1. **¿Qué son las VPC, subnets y security groups en AWS y cómo interactúan?**

**Definiciones:**

* **VPC (Virtual Private Cloud):** Es una red virtual privada en AWS donde se pueden lanzar recursos como instancias EC2, bases de datos, y balanceadores de carga.
* **Subnets:** Son segmentos dentro de una VPC que dividen la red en zonas privadas y públicas.
* **Security Groups:** Son reglas de firewall a nivel de instancia que controlan el tráfico de entrada y salida.

**Interacción:**

* **La VPC** contiene **subnets** donde se despliegan recursos.
* **Los security groups** restringen o permiten el tráfico hacia los recursos dentro de la VPC.
* **Las subnets** pueden ser **públicas** (accesibles desde Internet) o **privadas** (solo accesibles dentro de la VPC).

**Ejemplo:**  
Si tenemos una aplicación web en EC2:

1. **La VPC** define el entorno de red.
2. **Una subnet pública** aloja la instancia EC2 con acceso a Internet.
3. **Un security group** permite tráfico HTTP/HTTPS hacia la instancia.
4. **¿Cuándo usarías AWS Lambda frente a EC2?**

**Diferencias Claves:**

|  |  |  |
| --- | --- | --- |
| Característica | AWS Lambda | Amazon EC2 |
| Modelo | Serverless (sin servidores) | Servidor dedicado |
| Escalabilidad | Automática | Manual o con Auto Scaling |
| Costo | Pago por ejecución (ms) | Pago por hora/instancia |
| Mantenimiento | AWS gestiona todo | Necesita configuración y administración |
| Casos de uso | Eventos, microservicios, backend sin servidores | Aplicaciones web, procesamiento de datos, hosting de API |

**¿Cuándo usar AWS Lambda?**

* Para tareas pequeñas y eventos (ej: procesamiento de archivos en S3).
* Microservicios sin necesidad de servidores dedicados.
* Funciones de backend en tiempo real.

**¿Cuándo usar EC2?**

* Aplicaciones que requieren **control total del sistema operativo**.
* **Bases de datos personalizadas** o entornos de desarrollo complejos.
* Servidores web de alto tráfico con requisitos personalizados.

1. **Explica las diferencias entre SQL Server y MongoDB. ¿En qué casos usarías cada uno?**

**Comparación General:**

|  |  |  |
| --- | --- | --- |
| Característica | SQL Server | MongoDB |
| Modelo de Datos | Relacional (tablas, filas) | NoSQL (documentos JSON) |
| Escalabilidad | Vertical (más recursos a un solo servidor) | Horizontal (múltiples servidores) |
| Consultas | SQL (estructurado) | BSON (flexible) |
| Transacciones | ACID (estrictas) | Eventual consistency (menos estrictas) |
| Mejor para | Aplicaciones empresariales, bancos | Big Data, aplicaciones web dinámicas |

**Casos de Uso:**

* **Usar SQL Server** cuando se necesita:
  + Integridad transaccional y relaciones complejas entre datos.
  + Sistemas financieros o contables que requieren consistencia estricta.
* **Usar MongoDB** cuando se necesita:
  + Flexibilidad para almacenar datos sin esquema fijo.
  + Procesamiento rápido de grandes volúmenes de datos (Big Data, análisis).

**Ejemplo:**  
Si tienes un sistema de facturación de una empresa → **SQL Server**.  
Si manejas datos en tiempo real para una red social → **MongoDB**.

1. **Explica brevemente cómo funcionan colas SQS y sus casos de uso.**

**¿Qué es SQS?** Amazon **Simple Queue Service (SQS)** es un servicio de mensajería que permite la comunicación asincrónica entre aplicaciones mediante colas de mensajes.

**¿Cómo funciona?**

1. **Un productor** envía un mensaje a la cola SQS.
2. **Uno o varios consumidores** leen y procesan los mensajes.
3. **El mensaje se elimina** de la cola una vez procesado exitosamente.

**Tipos de Colas SQS:**

* **Standard Queue**: Ofrece entrega de mensajes en orden no garantizado, con alta disponibilidad.
* **FIFO Queue (First In, First Out)**: Mantiene el orden exacto de los mensajes.

**Casos de Uso:**

* **Procesamiento de tareas en segundo plano.**
  + Ejemplo: Generación de facturas después de una compra en línea.
* **Comunicación entre microservicios.**
  + Ejemplo: Un sistema donde un servicio de pedidos notifica a un servicio de inventario.
* **Balanceo de carga en sistemas distribuidos.**
  + Ejemplo: Varias instancias de una aplicación procesan trabajos en paralelo sin sobrecarga.

**Resumen de Respuestas**

|  |  |
| --- | --- |
| Pregunta | Resumen |
| 1. VPC, Subnets, Security Groups | Red privada en AWS, división en subredes y control de tráfico. |
| 2. AWS Lambda vs EC2 | Lambda para tareas event-driven; EC2 para control total y aplicaciones grandes. |
| 3. SQL Server vs MongoDB | SQL para datos estructurados, MongoDB para datos flexibles y escalables. |
| 4. Colas SQS | Comunicación asincrónica para microservicios, balanceo de carga y procesamiento en segundo plano. |