Modulo 9

Actividad 7 **Procesamiento sin servidor (aws lambda)**

Nombre: Marco Neira

**1. Arquitectura propuesta**

El flujo serverless se plantea de la siguiente manera:

1. **Carga del archivo PDF en Amazon S3**
   * El usuario sube un contrato a un bucket S3 configurado con eventos.
2. **Evento en S3 activa AWS Lambda**
   * Lambda valida que el archivo sea PDF.
   * Si no cumple, mueve el archivo a una carpeta de errores (/invalid/).
3. **Procesamiento del texto**
   * La función Lambda extrae texto básico (usando librerías como PyPDF2 o Textract).
   * Genera un resumen de las cláusulas clave.
4. **Notificación automática**
   * El resultado se publica en un **tópico SNS**.
   * SNS envía correo electrónico al usuario registrado.

**S3 (upload) → Lambda (validación/procesamiento) → SNS (notificación)**

**2. Justificación del uso de AWS Lambda**

* **Escalabilidad automática:** procesa cada archivo sin necesidad de gestionar servidores.
* **Modelo “pay-as-you-go”:** solo se paga cuando se ejecuta la función.
* **Simplicidad operativa:** elimina la necesidad de configurar y mantener EC2 o contenedores persistentes.
* **Integración nativa:** Lambda se conecta de forma directa con S3, SNS, CloudWatch e IAM.
* **Velocidad de despliegue:** permite iterar rápido en la lógica de negocio sin preocuparse por infraestructura.

**3. Servicios adicionales de AWS utilizados**

* **Amazon S3:** almacenamiento de contratos (entrada y errores).
* **AWS Lambda:** validación, procesamiento de texto y orquestación.
* **Amazon SNS:** envío de notificaciones por correo electrónico.
* **AWS IAM:** control de accesos (roles para Lambda y S3).
* **Amazon CloudWatch:** monitoreo de logs, métricas y alertas.
* **Amazon Textract:** para análisis más avanzado de texto en PDF.

**4. Estrategia para manejo de errores**

* Si el archivo no es PDF: moverlo automáticamente a una carpeta de errores en S3 (/invalid/).
* Si el análisis falla:

Registrar el evento en **CloudWatch Logs**.

Notificar al usuario mediante SNS con un mensaje de error.

* Configurar reintentos automáticos de Lambda en caso de fallas temporales.

**5. Buenas prácticas de seguridad y rendimiento**

* **Seguridad**

Usar **roles IAM con privilegios mínimos**.

Cifrar datos en reposo (S3 con SSE) y en tránsito (HTTPS).

Configurar políticas de bucket S3 para restringir accesos externos.

* **Rendimiento**

Optimizar el código Lambda y limitar dependencias.

Usar **timeouts adecuados** para evitar ejecuciones largas.

Procesar en paralelo si llegan muchos archivos al mismo tiempo.

Almacenar resultados en S3 en formato ligero (ej. JSON).

**6. Reflexión final: desafíos anticipados**

* Manejo de **gran volumen de archivos simultáneos** que podría requerir colas intermedias (ej. SQS).
* Limitaciones de tiempo de ejecución de Lambda (máx. 15 min).
* Posible necesidad de análisis avanzado de documentos (OCR, IA), que implicaría integrar Textract o Comprehend.
* Coordinación de múltiples funciones Lambda si el flujo se complejiza (ej. usando Step Functions).