LABORATORIO C9 V1:

IMPLEMENTACION RAG CON AMAZON BEDROCK

Desplegar una arquitectura de Retrieval-Augmented Generation **(RAG**) end-to-end en Amazon Web Services, integrando servicios de computación, almacenamiento y modelos fundacionales de IA para resolver problemas de consulta de información privada de forma segura y eficiente. El alumno deberá:

- Configurar IAM y Networking para garantizar una infraestructura protegida y funcional.
- Orquestar Bedrock Knowledge Bases con S3 y modelos Nova Lite de bajo costo.
- Desplegar una interfaz web en Streamlit sobre EC2 para interacción en tiempo real.

1. Configuración de Seguridad y Acceso (IAM)

Como buena práctica de arquitectura, nunca utilizaremos el usuario raíz (Root) para este laboratorio.

1.1. Crear Usuario de Administración (IAM User)

1. Inicia sesión como **Root** y ve a **IAM > Users > Create user.**
2. Nombre: admin-bedrock-lab
3. Permisos: Selecciona **Attach policies directly** y busca AdministratorAccess.
4. Cierra sesión Root e ingresa con este nuevo usuario.

1.2. Crear Politica de Permisos

Este rol permite que nuestro servidor web hable con el cerebro de la IA sin usar llaves de acceso.

1. Ve a **IAM > Roles > Create Policy**.
2. Selecciona el editor JSON y pega lo siguiente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "bedrock:InvokeModel",
                "bedrock:RetrieveAndGenerate",
                "bedrock:Retrieve"
            ],
            "Resource": "*"
        }
    ]
}
```

---

3. Guárdala como PermisosLaboratorioBedrock

1.3. Crear el Rol para la EC2

1. Ve a **IAM** > **Roles** > **Create Role**.
2. Selecciona **AWS Service** y luego **EC2**.
3. Busca y selecciona la política que creaste: PermisosLaboratorioBedrock.
4. Nómbralo Rol-EC2-Bedrock-Lab.

2. Configuración de Red (Security Groups)

Para que se pueda ver la aplicación desde internet, debemos abrir los puertos correctos.

1. Ve a **EC2 > Security Groups > Create**.
2. **Nombre:** SG-Laboratorio-IA
3. Agrega las siguientes **Inbound Rules**:
    - **Puerto 22 (SSH):** Origen "My IP" (para tu conexión segura).
    - **Puerto 8501 (TCP):** Origen "My IP" (puerto por defecto de Streamlit).

3. Preparación de Datos (S3)

1. Crea un bucket en **S3** con un nombre único (ej: datos-ia-usach-tunombre).
2. Sube un archivo PDF con información técnica o académica.

4. El Motor de Conocimiento (Knowledge Base)

Aquí es donde ocurre la magia del **RAG (Retrieval-Augmented Generation)**.

1. En el menú lateral de Bedrock, ve a **Build** > **Knowledge Bases** y haz clic en el botón naranja **Create**.
2. **Paso 1 (Detalles):**
    - **Name:** kb-laboratorio-ia.
    - **IAM Permissions:** Deja seleccionado **Create and use a new service role**.
    - **Data Source:** Selecciona **Amazon S3**.
3. **Paso 2 (Configurar Origen):**
    - **S3 URI:** Haz clic en **Browse S3** y selecciona el bucket que creaste en el punto 2.
    - **Parsing Strategy:** Mantén **Amazon Bedrock default parser**.
    - **Chunking Strategy:** Mantén **Default chunking**.
4. **Paso 3 (Vector Store - Muy Importante):**
    - **Embedding Model:** Elige **Titan Text Embeddings v2**.
    - **Vector Store:** Elige **Quick create a new vector store**. En **Vector store type** selecciona **Amazon OpenSearch Serverless**.
5. **Finalizar y Sincronizar:** Una vez creada la KB, pulsa el botón **Sync**. Anota el **Knowledge Base ID**.

5. Despliegue de la Aplicación Web (EC2)

1. Lanza una instancia **t3.micro** (Free Tier) usando el Security Group (SG-Laboratorio-IA) y el IAM Role (Rol-EC2-Bedrock-Lab) creados en el punto 1.
2. Conéctate por SSH y ejecuta Bash:

```
sudo dnf update -y
sudo dnf install python3-pip -y
pip install boto3 streamlit
```

3. Crea el archivo de Python ejecutando la línea de comando:

```
nano app.py
```

4. Para crear la aplicación, pega el código del asistente en nano (asegúrate, tener la región correcta):

```
import streamlit as st
import boto3

client = boto3.client('bedrock-agent-runtime', region_name='us-east-1')

st.title("Asistente RAG - Capacitación USACH")
kb_id = st.text_input("Ingresa el ID de tu Knowledge Base (Ej: US6LSEJEGC):")
query = st.chat_input("¿Qué deseas preguntar sobre tus documentos?")

if query and kb_id:
    with st.spinner("Buscando en documentos..."):
        # Esta línea debe estar indentada (con espacios)
        res = client.retrieve_and_generate(
            input={'text': query},
            retrieveAndGenerateConfiguration={
                'type': 'KNOWLEDGE_BASE',
                'knowledgeBaseConfiguration': {
                    'knowledgeBaseId': kb_id,
                    'modelArn': 'arn:aws:bedrock:us-east-1::foundation-
model/amazon.nova-lite-v1:0'
                }
            }
        )
        st.write(res['output']['text'])
```

Guarda (Ctrl+O, Enter) y **salir** (Ctrl+X).

5. Lanza la web:

```
streamlit run app.py --server.port 8501
```

Accede a tu navegador y comprueba el acceso http://IpPublicaEC2:8501 y realiza preguntas a tu modelo.

6. Cierre y Validación del Laboratorio

El alumno deberá subir un PDF con:

- Captura de Pantalla 1: La consola de Bedrock mostrando la Knowledge Base con estado "Active" y la sincronización completada.
- Captura de Pantalla 2: La aplicación Streamlit funcionando (URL visible) con una respuesta coherente basada en el PDF subido.
- Captura de Pantalla 3 (Limpieza): La consola de OpenSearch Serverless indicando "No collections", validando que el alumno borró la infraestructura para evitar cobros.
- Nota: Las capturas de pantalla, deben evidenciar el ID de la cuenta del alumno.

⚠️ PASO OBLIGATORIO: Eliminación de Infraestructura

Para evitar cargos automáticos que consuman los créditos de 6 meses, los alumnos **DEBEN**:

1. **Eliminar la Knowledge Base**
2. **Eliminar en Amazon OpenSearch Services** en Serverless.
3. **Terminar (Terminate)** la instancia EC2.
4. **Vaciar y eliminar** el bucket de S3.
5. **Eliminar** Roles, Politicas y Segurity Group creados.