



28 de junio de 2025
Bogotá, Colombia



PRESENTADO POR CLOUDOPS GUILD COMMUNITY

Tu Arquitectura bajo la Lupa de la IA – Revisión Inteligente de Diagramas AWS

Mario R. Serrano Pineda

Ingeniero DevOps





Mario R. Serrano Pineda
Ingeniero DevOps

AWS
community
builder

💬 **Sobre mí**

🚀 **Apasionado por la nube, la IA y el aprendizaje colectivo.**

🎓 **AWS Community Builder** e ingeniero DevOps con enfoque en SRE, automatización e innovación educativa.

📖 Fundador de **CloudOps Guild**, una comunidad para formar a la nueva generación de ingenieros cloud con laboratorios reales, retos y bootcamps.

🧠 He liderado múltiples **challenges y talleres** enfocados en certificaciones AWS, agentes inteligentes con Amazon Bedrock, y automatización con n8n.

🌟 Un dato curioso: ¡he creado más de **40 laboratorios reales en AWS** solo este año, incluyendo proyectos con IA y sistemas multiagente!

🏃 Entrenando para una maratón. Creo que la disciplina deportiva y la tecnológica van de la mano.

🌍 Me apasiona leer, viajar, aprender idiomas y compartir conocimiento para empoderar a otras personas en tecnología.



Agenda

- | | | | |
|----|---------------------------|----|----------------------------------|
| 01 | Objetivo del workshop | 05 | Paso a paso del workshop |
| 02 | Requisitos previos | 06 | Momentos de codificación guiada |
| 03 | Fundamentos | 07 | Tips prácticos/ Buenas prácticas |
| 04 | Arquitectura del proyecto | 08 | Qué sigue? |



Objetivo del workshop

 ¿Y si una IA pudiera encontrar los puntos débiles de tu arquitectura antes que tu cliente lo haga?
Hoy vamos a construir ese detector inteligente juntos.

- ✓ Construir una aplicación práctica con IA que:
 - Analiza imágenes de diagramas arquitectónicos
 - Detecta fallos y mejora resiliencia con IA
 - Evalúa contra los pilares del AWS Well Architected Framework
 - Utiliza Amazon Bedrock + HTML + Python



Requisitos previos

¿Qué necesitan tener listo?

1. Cuenta de AWS (se entrega en el workshop)
2. Acceso a Bedrock y al modelo habilitado
3. Repo GitHub Clonado
"Clona el repositorio oficial del workshop desde URL
Contiene todo lo que necesitas: imágenes, código, instrucciones y enlaces útiles."



Fundamentos

¿Qué es la IA generativa?

1. IA Generativa: Modelo entrenado para crear contenido nuevo (texto, imagen, código...).
2. En este taller usamos IA para interpretar imágenes de arquitectura y generar texto explicativo.
3. Ejemplo práctico: "Subo una imagen, doy una instrucción y obtengo análisis y sugerencias."



Fundamentos

¿Qué es un modelo multimodal?

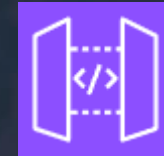
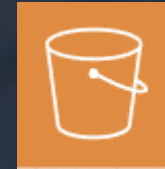
1. Modelos multimodales pueden procesar múltiples tipos de datos (texto, imagen, audio).
2. En este caso: Claude 3 Sonnet, modelo de Amazon Bedrock, procesa imágenes y responde en texto.
3. Ejemplo: "Subes un diagrama, el modelo lo entiende visualmente y te responde con observaciones inteligentes."
4. No es OCR, es *análisis visual semántico*.



Fundamentos

Servicios de AWS usados

- **Amazon S3:** Servicio de almacenamiento de objetos que permite guardar imágenes, documentos y archivos estáticos en la nube.
- **Amazon API Gateway:** Servicio que expone una API para recibir solicitudes HTTP y enrutar peticiones a servicios backend como Lambda.
- **AWS Lambda:** Servicio de cómputo sin servidores que ejecuta tu código en respuesta a eventos, sin necesidad de administrar servidores.
- **Amazon Bedrock:** Plataforma que proporciona acceso a modelos de IA generativa (como Claude o Nova Pro) mediante una API segura.
- **HTML local/simple:** Interfaz web minimalista que permite al usuario seleccionar y subir una imagen desde su navegador.

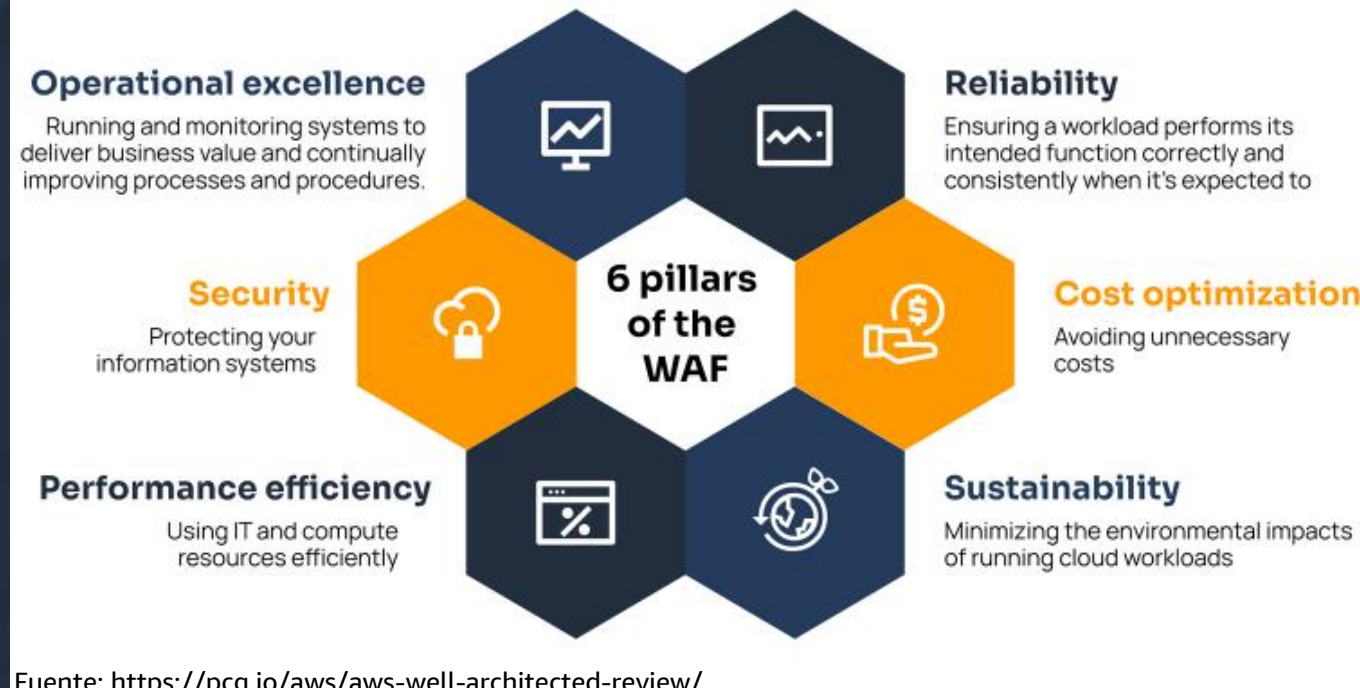


Fundamentos

¿Qué es el AWS Well-Architected Framework?

Conjunto de **buenas prácticas de arquitectura en AWS**

- Se basa en **6 pilares**: Operacional, Seguridad, Fiabilidad, Rendimiento, Optimización de costos, Sostenibilidad
- En este taller nos centramos en detectar problemas comunes como:
 - Single Point of Failure
 - Falta de redundancia
 - No uso de AZs
 - Falta de balanceadores



Fuente: <https://pcg.io/aws/aws-well-architected-review/>

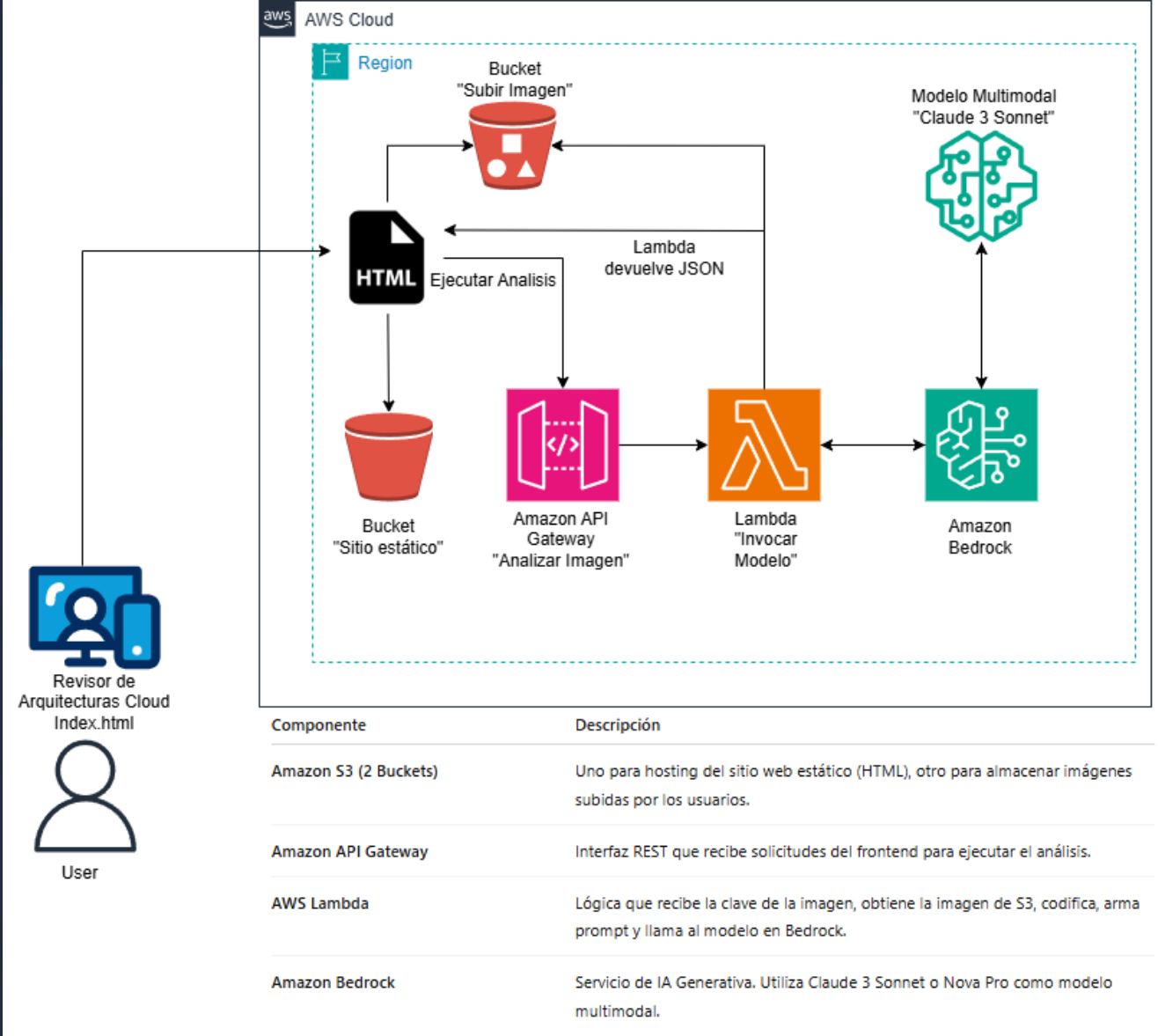


“¿Listos para poner esto en práctica?”

“Ya tienes los conceptos. Ahora vamos a convertir este conocimiento en algo tangible. ¡Hora de enseñarle a la IA cómo revisar arquitecturas!”



Arquitectura del proyecto



[Usuario en navegador]

[HTML en S3 (Sitio Estático)]

[Formulario HTML]
- Subir imagen (input file)
- Ejecutar análisis (fetch)

[Imagen se sube a Bucket S3 de Imágenes]

[API Gateway (REST)]

[AWS Lambda]
- Obtener imagen de S3
- Codificar en Base64
- Armar prompt
- Enviar a modelo Claude o Nova Pro en Amazon Bedrock

[Amazon Bedrock (Claude 3 Sonnet o Nova Pro)]

← Respuesta en lenguaje natural (outputText)

[Lambda devuelve JSON]

[Frontend muestra resultado en pantalla]



Paso a paso del workshop

1. Crear los buckets en S3.
2. Crear la función Lambda.
3. Crear API Gateway para exponer Lambda.
4. Subir el HTML del frontend al bucket del sitio web.
5. Validación del flujo.



Momentos de codificación guiada

Bucket Policy - Imágenes

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowUploadFromFrontend",
      "Effect": "Allow",
      "Principal": "*",
      "Action": ["s3:PutObject", "s3:GetObject"],
      "Resource": "arn:aws:s3:::Nombre_bucket_sitio_estático/*"
    }
  ]
}
```



Momentos de codificación guiada

Bucket Policy - Imágenes

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowUploadFromFrontend",
      "Effect": "Allow",
      "Principal": "*",
      "Action": ["s3:PutObject", "s3:GetObject"],
      "Resource": "arn:aws:s3::revisor-imágenes-bucket/*"
    }
  ]
}
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3::revisordearquitecturas19052025/*"
    }
  ]
}
```



Momentos de codificación guiada

Bucket Policy – Sitio web

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "LecturaPublicaBucket",  
      "Effect": "Allow",  
      "Principal": "*",  
      "Action": "s3:GetObject",  
      "Resource": "arn:aws:s3:::Nombre_bucket_sitio_estático/*"  
    }  
  ]  
}
```



Momentos de codificación guiada

Bucket Policy – Sitio web

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "LecturaPublicaBucket",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3::: Nombre_bucket_sitio_estático /*"
    }
  ]
}
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "LecturaPublicaBucket",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::revisor-web-bucket20062025/*"
    }
  ]
}
```



Momentos de codificación guiada

Código función Lambda

```
def lambda_handler(event, context):  
    try:  
        print("📄 Lambda iniciada")  
        print("📄 Evento recibido:", event)  
  
        body = event # Ya llega como objeto plano  
        key = body.get('key')  
        bucket = 'Nombre_bucket_imágenes'  
  
        if not key:  
            return {  
                "statusCode": 400,  
                "body": json.dumps({ "error": "No se especificó la clave de la imagen." })  
            }  
    
```



Momentos de codificación guiada

Código función Lambda

```
def lambda_handler(event, context):  
    try:  
        print("🚀 Lambda iniciada")  
        print("📦 Evento recibido:", event)  
  
        body = event # Ya llega como objeto plano  
        key = body.get('key')  
        bucket = 'Nombre_bucket_imágenes'  
  
        if not key:  
            return {  
                "statusCode": 400,  
                "body": json.dumps({ "error": "No se especificó la clave de la imagen." })  
            }  
    
```



Momentos de codificación guiada

Código función Lambda

```
prompt = {
  "messages": [
    {
      "role": "user",
      "content": [
        {
          "type": "image",
          "source": {
            "type": "base64",
            "media_type": "image/png",
            "data": encoded_image
          }
        },
        {
          "type": "text",
          "text": "A continuación recibirás un diagrama de arquitectura de AWS en formato imagen.\n\nTu tarea es analizarlo de forma precisa y únicamente basada en su contenido visual.\n\nRealiza el siguiente análisis:\n\n1. **Componentes identificados**: Enumera los servicios y recursos de AWS que se pueden reconocer visualmente (por logotipo, etiquetas o estructuras comunes). No incluyas servicios que no estén explícitos en el diagrama.\n\n2."
        }
      ]
    },
    {
      "anthropic_version": "bedrock-2023-05-31",
      "max_tokens": 1024
    }
  ]
}
```



Momentos de codificación guiada

Código función Lambda

```
prompt = {  
  "messages": [  
    {  
      "role": "user",  
      "content": [  
        {  
          "type": "image",  
          "source": {  
            "type": "base64",  
            "media_type": "image/png",  
            "data": encoded_image  
          }  
        },  
        {  
          "type": "text",  
          "text": "A continuación recibirás un diagrama de  
arquitectura de AWS en formato imagen.\n\nTu tarea es analizarlo de forma precisa  
y únicamente basada en su contenido visual.\n\nRealiza el siguiente  
análisis:\n\n1. **Componentes identificados**: Enumera los servicios y recursos  
de AWS que se pueden reconocer visualmente (por logotipo, etiquetas o estructuras  
comunes). No incluyas servicios que no estén explícitos en el diagrama.\n\n2."  
        }  
      ]  
    }  
  ],  
  "anthropic_version": "bedrock-2023-05-31",  
  "max_tokens": 1024  
}
```



Momentos de codificación guiada

Código función Lambda

```
print("🧠 Enviando prompt a Claude...")
try:
    response = bedrock.invoke_model(
        modelId="anthropic.claude-3-sonnet-20240229-v1:0",
        body=json.dumps(prompt),
        contentType="application/json",
        accept="application/json"
    )
except Exception as invoke_error:
    print("❌ Error en invoke_model:", str(invoke_error))
    raise invoke_error
```



Momentos de codificación guiada

Código función Lambda

```
print("🧠 Enviando prompt a Claude...")
try:
    response = bedrock.invoke_model(
        modelId="anthropic.claude-3-sonnet-20240229-v1:0",
        body=json.dumps(prompt),
        contentType="application/json",
        accept="application/json"
    )
except Exception as invoke_error:
    print("❌ Error en invoke_model:", str(invoke_error))
    raise invoke_error
```



Momentos de codificación guiada

INDEX HTML

```
<p id="uploadStatus"></p>
<p id="spinner">⌛ Analizando el diagrama... por favor
espera.</p>
<div id="result">Aún no se ha realizado ningún análisis.</div>

<script>
  const bucketName = 'Bucket_donde_se_suben_imagenes';
  const region = 'region_API';
  const apiUrl = 'URL_Invoke de la API';

  async function uploadImage() {
    const file =
document.getElementById('fileInput').files[0];
    if (!file) {
      alert("No se seleccionó ninguna imagen.");
      return;
    }

    const s3Key = `uploads/${Date.now()}_${file.name}`;
    const uploadUrl =
`https://${bucketName}.s3.${region}.amazonaws.com/${s3Key}`;

    const response = await fetch(uploadUrl, {
      method: 'PUT',
      body: file,
      headers: { 'Content-Type': file.type }
    });
  }
}
```

```
58
59   <script>
60     const bucketName = 'Bucket_donde_se_suben_imagenes';
61     const region = 'region_API';
62     const apiUrl = 'URL_Invoke de la API';
63
```



Momentos de codificación guiada

INDEX HTML

```
<p id="uploadStatus"></p>
<p id="spinner">⌚ Analizando el diagrama... por favor
espera.</p>
<div id="result">Aún no se ha realizado ningún análisis.</div>

<script>
  const bucketName = 'Bucket_donde_se_suben_imagenes';
  const region = 'region_API';
  const apiUrl = 'URL_Invoke de la API';

  async function uploadImage() {
    const file =
document.getElementById('fileInput').files[0];
    if (!file) {
      alert("No se seleccionó ninguna imagen.");
      return;
    }

    const s3Key = `uploads/${Date.now()}_${file.name}`;
    const uploadUrl =
`https://${bucketName}.s3.${region}.amazonaws.com/${s3Key}`;

    const response = await fetch(uploadUrl, {
      method: 'PUT',
      body: file,
      headers: { 'Content-Type': file.type }
    });
  }
}
```

```
58
59   <script>
60     const bucketName = 'Bucket_donde_se_suben_imagenes';
61     const region = 'region_API';
62     const apiUrl = 'URL_Invoke de la API';
63
```



Momentos de codificación guiada

INDEX HTML

```
59 <script>
60   const bucketName = 'revisor-imagen-bucket21062025';
61   const region = 'us-east-1';
62   const apiUrl = 'https://agx8kxwqe3.execute-api.us-east-2.amazonaws.com/Dev/analyze';
63
```

```
<p id="uploadStatus"></p>
<p id="spinner">⌚ Analizando el diagrama... por favor espera.</p>
<div id="result">Aún no se ha realizado ningún análisis.</div>

<script>
  const bucketName = 'Bucket donde se suben imagenes';
  const region = 'region_API';
  const apiUrl = 'URL_Invoke de la API';

  async function uploadImage() {
    const file = document.getElementById('fileInput').files[0];
    if (!file) {
      alert("No se seleccionó ninguna imagen.");
      return;
    }

    const s3Key = `uploads/${Date.now()}_${file.name}`;
    const uploadUrl = `https://${bucketName}.s3.${region}.amazonaws.com/${s3Key}`;

    const response = await fetch(uploadUrl, {
      method: 'PUT',
      body: file,
      headers: { 'Content-Type': file.type }
    });
  }

```



Tips prácticos/ Buenas prácticas

◆ 1. Usa imágenes claras y legibles

Exporta el diagrama desde Draw.io en formato PNG o JPG, evitando fondos oscuros o muy cargados.

◆ 2. Incluye íconos estándar de AWS

Los modelos reconocen mejor los elementos cuando usas los íconos oficiales de arquitectura AWS.

◆ 3. Usa prompts bien enfocados

Sé directo: "Revisa este diagrama y dime si cumple con buenas prácticas de alta disponibilidad, resiliencia y seguridad en AWS."

◆ 4. Verifica permisos S3 y Bedrock

Asegúrate de que tu función Lambda tenga permisos para acceder al bucket S3 y hacer InvokeModel en Bedrock.

◆ 5. Prueba primero con un diagrama de ejemplo

Antes de usar tu propia arquitectura, valida el flujo con los diagramas incluidos en el repositorio.

◆ 6. Revisa la respuesta IA antes de actuar

La IA ofrece recomendaciones, no decisiones. Valida todo antes de aplicarlo en entornos reales.

◆ 7. Captura tus resultados

Guarda las respuestas generadas por la IA para analizarlas luego o compartirlas con tu equipo.

◆ 8. Comparte y mejora

Sube tus resultados, aporta feedback al repo y colabora con la comunidad CloudOps Guild





Qué sigue?





Hoy construiste...

- Una aplicación capaz de analizar arquitecturas con IA
- Usaste servicios como S3, Lambda, API Gateway y Bedrock
- Probaste cómo la IA puede ayudarte a mejorar soluciones en AWS

Súmate a CloudOps Guild



-  Aprende con más labs, retos y agentes IA en AWS
-  Síguenos en YouTube: [@CloudOpsGuild](https://www.youtube.com/CloudOpsGuild)
-  Únete al canal de LinkedIn <https://www.linkedin.com/company/cloudopsguild>

¿Qué puedes hacer ahora?

-  Agregar puntaje por pilar del Well-Architected Framework
-  Generar reportes con gráficas
-  Extraer texto con Amazon Textract
-  Generar IaC automáticamente desde el análisis

Comparte lo que creaste

“Ahora no solo diseñas arquitecturas... también las puedes auditar con IA.”

-  Comparte tu experiencia en redes y con tu equipo
-  Da una estrella al repo y deja tu feedback

Etiqueta: #CloudOpsGuild #AWSCommunityDay #IAenAWS



“

**Aprender es fácil.
Aplicarlo es un reto.
Pero compartir lo aprendido...
eso es lo que te convierte en un líder cloud.**





¡Gracias!

Mario R. Serrano Pineda

Marosepi2020@gmail.com

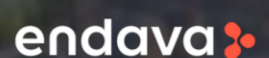
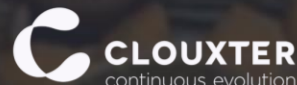
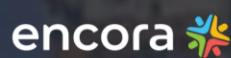
<https://www.linkedin.com/in/mario-rodrigo-serrano-pineda/>

<https://medium.com/@marioserranopineda>

<https://github.com/marosepi2020>



Por favor, completa la encuesta de la sesión.





28 de junio de 2025
Bogotá, Colombia

