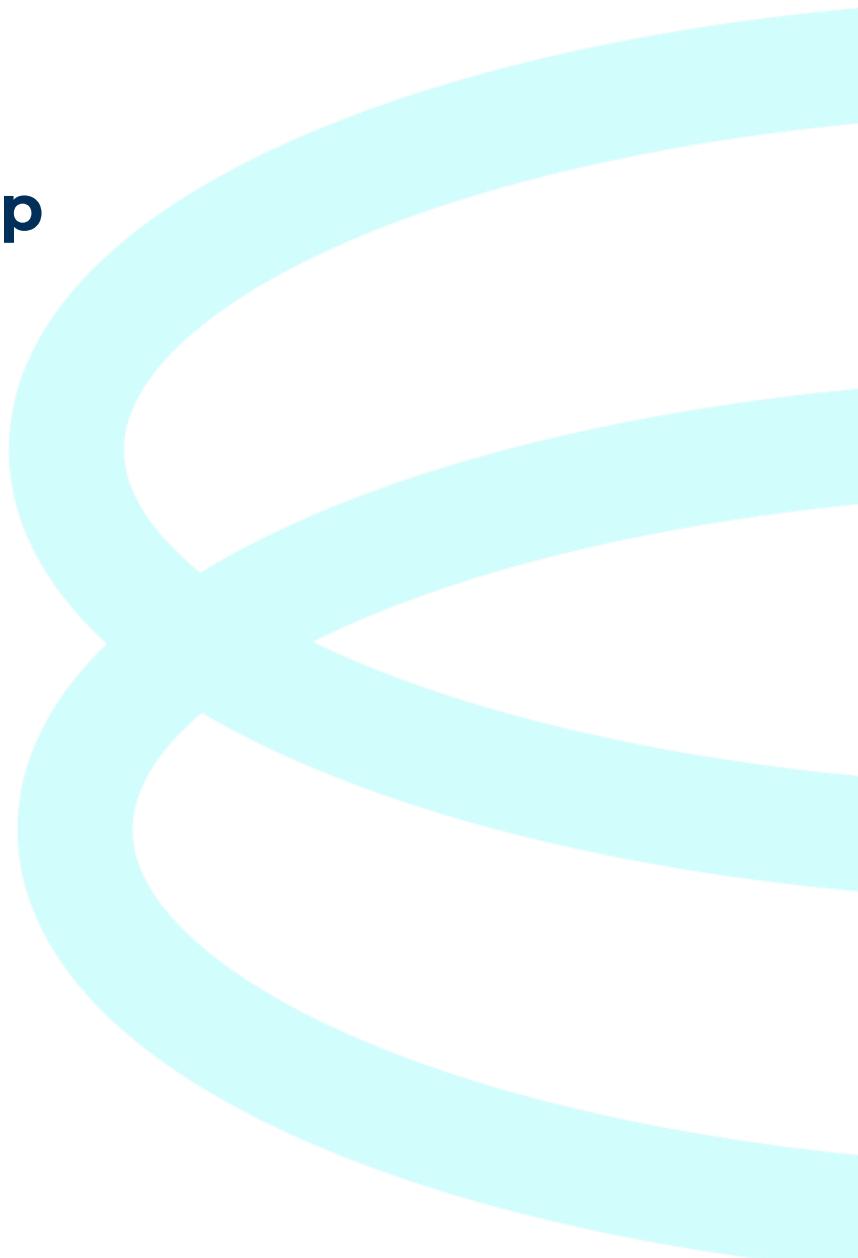




## Bintec Workshop



09/2025



# Contenido

1.	Caso de Uso.....	3
	Bintec Demo - Sistema de Análisis Financiero con IA Explicable 	3
2.	Arquitectura.....	7
	Arquitectura Multi-Agente Cibest 	8
	Componentes del Sistema .....	8
	Capa Frontend 	8
	Servicios Backend 	8
	Sistema Multi-Agente de IA 	8
	Servicios de Soporte 	9
	Controles de Seguridad 	9
	Flujo de Datos 	9
3.	Workshop Bintec: Implementación Paso a Paso .....	10

## 1. Caso de Uso

### Bintec Demo - Sistema de Análisis Financiero con IA Explicable



Grupo CIBest, líder en servicios financieros, enfrenta el desafío de modernizar sus procesos de análisis y consultoría financiera. La presión del mercado exige respuestas más rápidas, precisas y transparentes para sus clientes corporativos.

#### Situación Actual

- ⌚ Consultas complejas toman 2-3 días en resolverse
- 👥 Requieren coordinación entre múltiples departamentos
- 📋 Análisis regulatorio manual y propenso a errores
- 🔍 Falta de trazabilidad en el proceso de toma de decisiones
- 💼 Clientes demandan explicaciones claras del razonamiento

#### Visión Estratégica

Implementar un **sistema multi-agente de IA explicable** usando Amazon Bedrock que pueda:

- Analizar consultas financieras complejas en tiempo real
- Proporcionar razonamiento transparente paso a paso
- Mantener altos estándares de precisión y confiabilidad
- Escalar para atender múltiples clientes simultáneamente

#### Alcance del Proyecto

## Fase 1: Demo Interactivo Multi-Agente

Implementación de un sistema de análisis con **cuatro agentes especializados**:

### **Agente Documental**

- Análisis de documentación regulatoria
- Procesamiento de normativas CNBV/Banxico
- Identificación de requisitos de cumplimiento

### **Agente de Riesgo**

- Evaluación de riesgos operacionales y crediticios
- Modelos de stress testing
- Análisis de probabilidades e impacto

### **Agente Regulatorio**

- Marco normativo aplicable actualizado
- Cambios regulatorios recientes
- Impacto en operaciones financieras

### **Agente de Negocio**

- Oportunidades de crecimiento
- Análisis de mercado y competencia
- Proyecciones financieras personalizadas

## **Progreso Actual**

### **Completado**

- Frontend Next.js con flujo de usuario completo
- Arquitectura base de componentes React
- Integración con AWS Amplify
- Diseño UX/UI corporativo

### **Por Desarrollar en el Taller**

1. **Agentes IA Especializados** (4 Lambdas)
2. **Orquestador Multi-Agente** (Lambda Principal)
3. **Integración con Amazon Bedrock**
4. **Sistema de Email Automático** (SES)

### **Plan de Acción del Workshop**

## 1. Configuración del Entorno

- Clonar repositorio base
- Configurar credenciales AWS
- Inicializar Amplify CLI

## 2. Desarrollo de Agentes IA

- Crear 4 funciones Lambda especializadas
- Implementar prompts específicos por dominio
- Configurar Amazon Bedrock Claude 3

## 3. Orquestación Multi-Agente

- Desarrollar Lambda orquestador
- Implementar procesamiento paralelo
- Crear lógica de consolidación de resultados

## 4. Integración Frontend-Backend

- Conectar API Gateway
- Implementar llamadas asíncronas
- Configurar manejo de estados

## 5. Sistema de Comunicación

- Configurar Amazon SES
- Implementar templates de email
- Probar envío automático

## 6. Despliegue y Pruebas

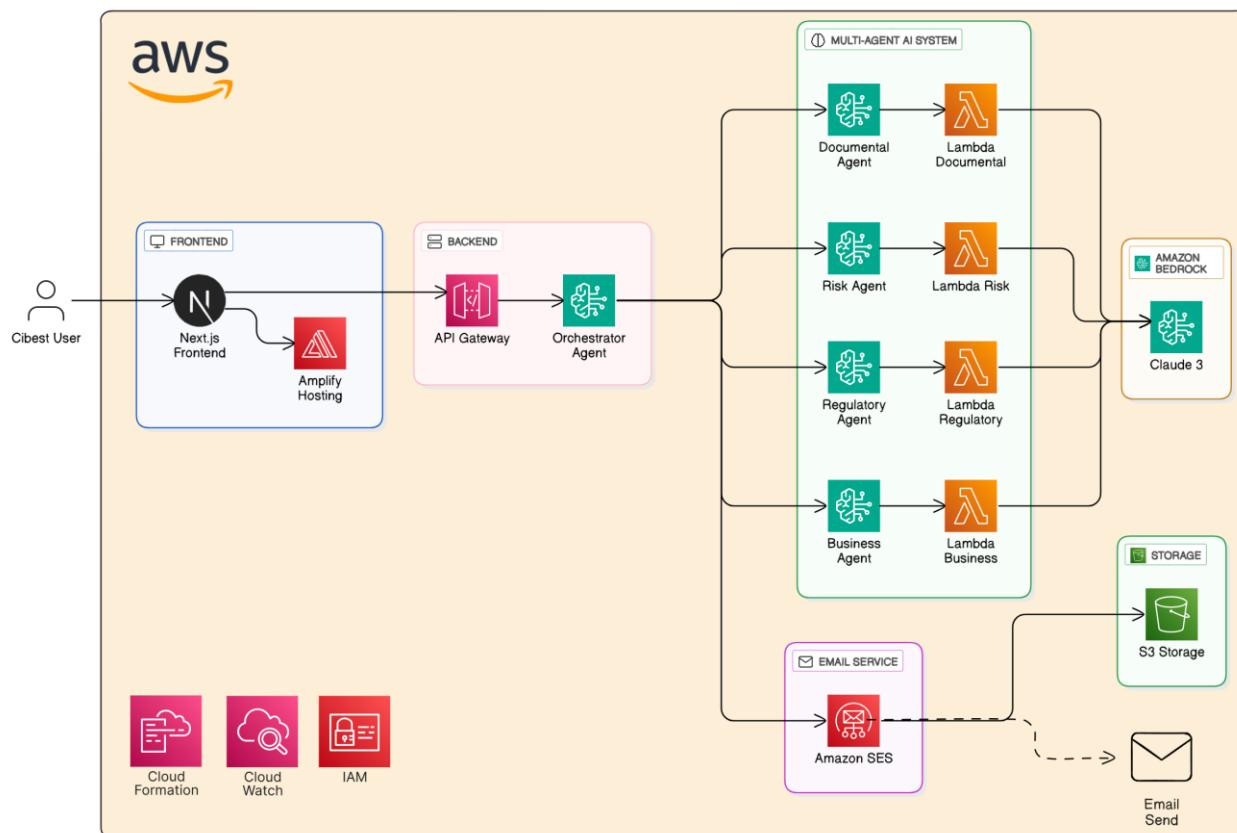
- Deploy completo en AWS
- Pruebas end-to-end
- Validación de IA explicable

## Resultados Esperados

Al final del workshop, los participantes habrán construido:

- Sistema funcional de análisis financiero con IA
- 4 agentes especializados con razonamiento transparente
- Arquitectura escalable en AWS
- Demo interactivo listo para producción

## 2. Arquitectura



# Arquitectura Multi-Agente Cibest E

## Componentes del Sistema

### Capa Frontend

#### Interfaz de Usuario:

- **Next.js Frontend** desplegado con **Amplify Hosting**
- Acceso web responsive para los usuarios finales (Cibest User)
- Conexión directa y segura con la capa de backend

#### Características:

- Experiencia de usuario intuitiva
- Integración rápida con servicios backend
- Actualizaciones y despliegues continuos con Amplify

### Servicios Backend

#### Capa API:

- **Amazon API Gateway** como punto de entrada de las solicitudes del usuario
- Control de seguridad, autenticación y enrutamiento de peticiones

#### Orquestación:

- **Orchestrator Agent** que recibe la petición desde API Gateway y la distribuye al agente especializado correspondiente

#### Funciones Clave:

- Administración centralizada del flujo de solicitudes
- Delegación a agentes especializados según contexto (documental, riesgo, regulatorio, negocio)
- Integración con el motor de IA generativa (Bedrock Claude 3)

### Sistema Multi-Agente de IA

#### Agentes Especializados:

- **Documental Agent** + Lambda Documental → Procesamiento y análisis de documentos

- **Risk Agent** + Lambda Risk → Identificación y evaluación de riesgos
- **Regulatory Agent** + Lambda Regulatory → Cumplimiento normativo y regulatorio
- **Business Agent** + Lambda Business → Evaluación de impacto y perspectiva de negocio

#### **Motor de IA Generativa:**

- **Amazon Bedrock con Claude 3**
- Habilita razonamiento avanzado y soporte contextual
- Potencia las respuestas de cada agente especializado

#### **Funciones Clave:**

- División de responsabilidades entre agentes
- Respuestas contextualizadas y especializadas
- Escalabilidad mediante funciones Lambda desacopladas

## **Servicios de Soporte**

#### **Almacenamiento:**

- **Amazon S3 Storage**
- Guarda documentos, resultados procesados y evidencias

#### **Servicio de Correos:**

- **Amazon SES**
- Permite enviar notificaciones automáticas, reportes y resultados al usuario

## **Controles de Seguridad**

- Seguridad integrada en **API Gateway** para el acceso controlado
- Aislamiento de lógica en **Lambdas independientes**
- Cumplimiento normativo habilitado por el **Regulatory Agent**
- Protección de datos con almacenamiento seguro en **S3** y comunicación cifrada

## **Flujo de Datos**

1. El usuario (CI Best User) interactúa con la aplicación en **Next.js** alojada en **Amplify Hosting**.
2. La solicitud pasa por **API Gateway**, que la envía al **Orchestrator Agent**.
3. El orquestador determina qué agente especializado manejará la petición.

4. El agente seleccionado ejecuta lógica en **Lambda** y puede invocar **Claude 3 en Amazon Bedrock** para generación avanzada.
5. Los resultados son almacenados en **S3** o enviados al usuario por **Amazon SES**.
6. La respuesta se muestra al usuario en la interfaz frontend.

## 3. Workshop Bintec: Implementación Paso a Paso

### TAREA 1: Preparación del Entorno

*Tiempo estimado: 10 minutos*

- “Si no tienes configurada la AWS CLI, puedes descargarla [aquí](#).
- Si no tienes Node.js, instálalo desde [aquí](#).”

#### **Paso 1.1: Verificar Requisitos y Configurar AWS**

**¿Qué vamos a hacer?** Preparar el entorno de trabajo.

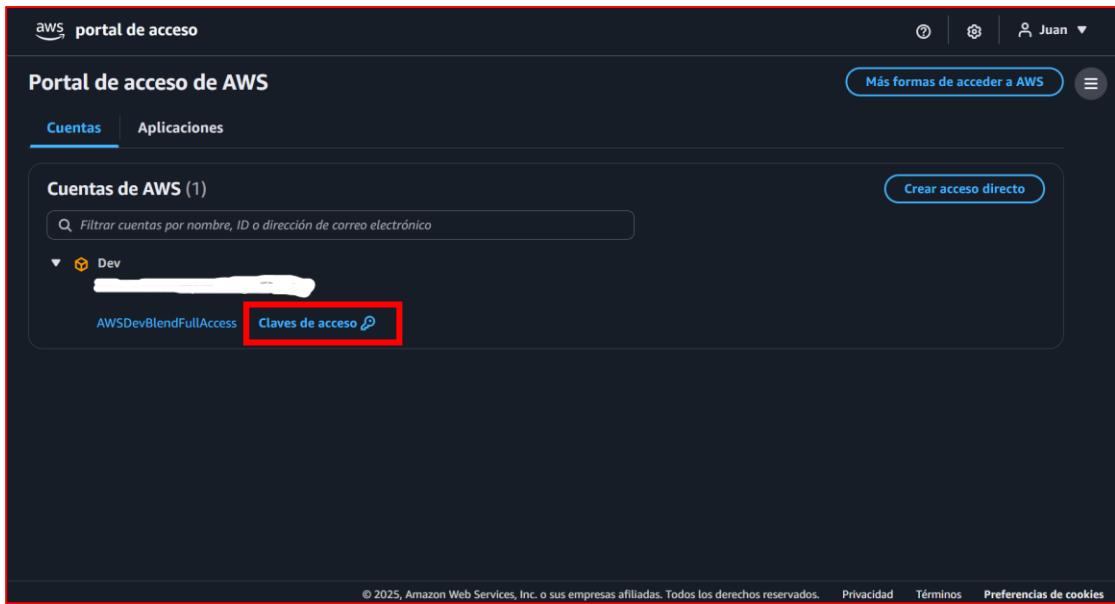
##### **Acciones:**

1. Verificar AWS CLI:

**`aws –version`**

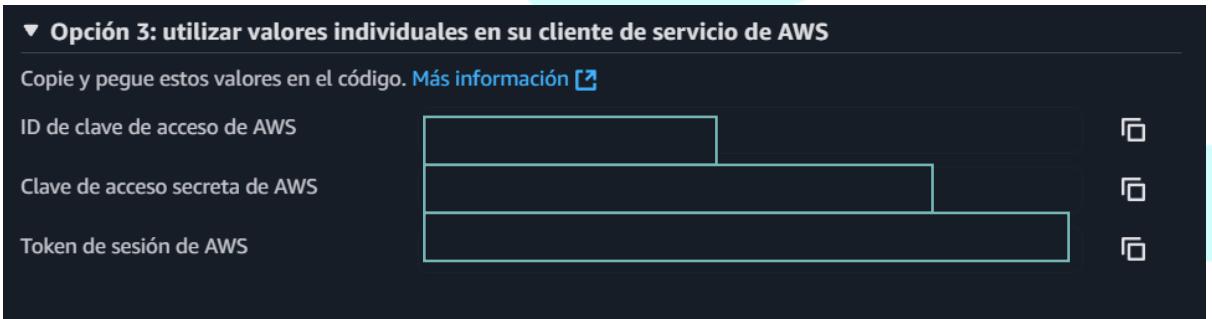
2. Obtener y registrar las credenciales de AWS:

2.1. Ingresar a la cuenta de AWS



The screenshot shows the AWS IAM console with a red box highlighting the 'Claves de acceso' (Access keys) button for a specific user. The button is located in the 'AWSDevBlendFullAccess' role under the 'Claves de acceso' section.

## 2.2. Copiamos una a una las credenciales de acceso a la cuenta y las configuramos en el siguiente paso:



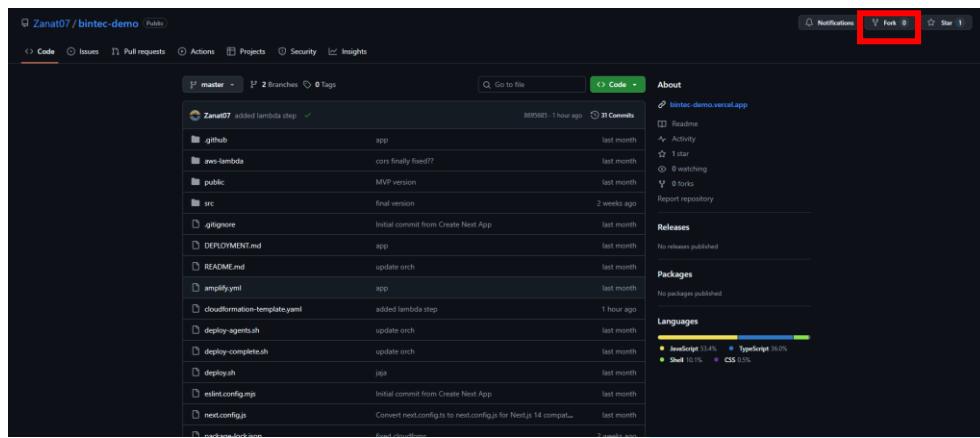
The screenshot shows the 'Access keys' section of the AWS IAM console. It displays three input fields for copying values: 'ID de clave de acceso de AWS', 'Clave de acceso secreta de AWS', and 'Token de sesión de AWS'. Each field has a copy icon (a square with a white 'C') to its right.

## 2.3 Ejecutamos el comando: **aws configure**

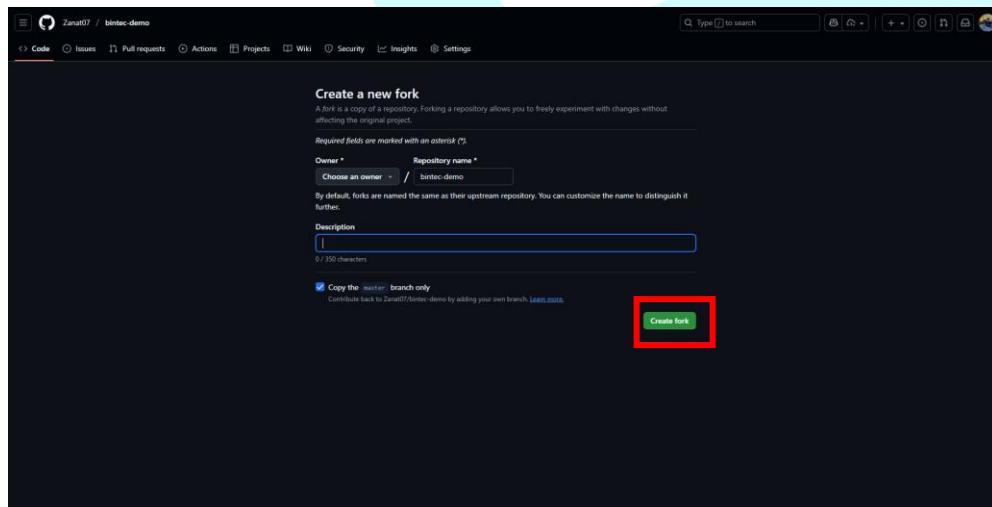
```
PS C:\Users\nuvu-pc-00056> aws configure
AWS Access Key ID [*****RGZP]: 
AWS Secret Access Key [*****gY2R]: 
Default region name [us-east-1]: 
Default output format [json]: |
```

### 3. Clonar repositorio:

**Accedemos al repo <https://github.com/Zanat07/bintec-demo.git>**



## Seleccionamos nuestro usuario y creamos el fork

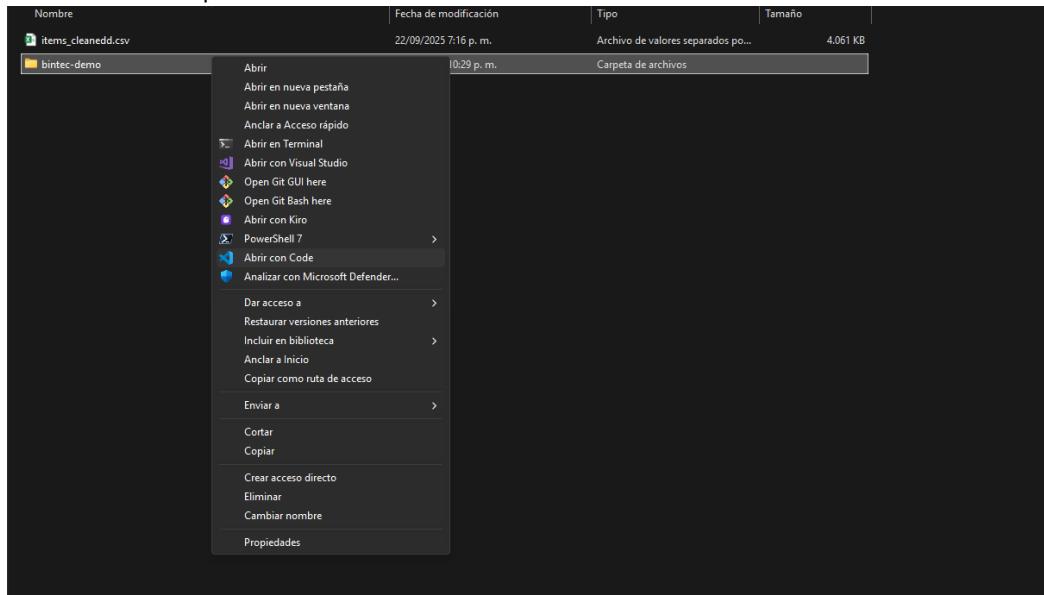


## Clonar tu fork:

Una vez hecho el fork, clona tu propio repositorio usando el siguiente comando (reemplaza `<tu-usuario>` por tu nombre de usuario en GitHub):

```
git clone https://github.com/<tu-usuario>/bintec-demo.git
```

#### 4. Abrimos el repositorio con nuestro IDE:



#### TAREA 2: Desplegar Infraestructura con CloudFormation

Tiempo estimado: 15 minutos

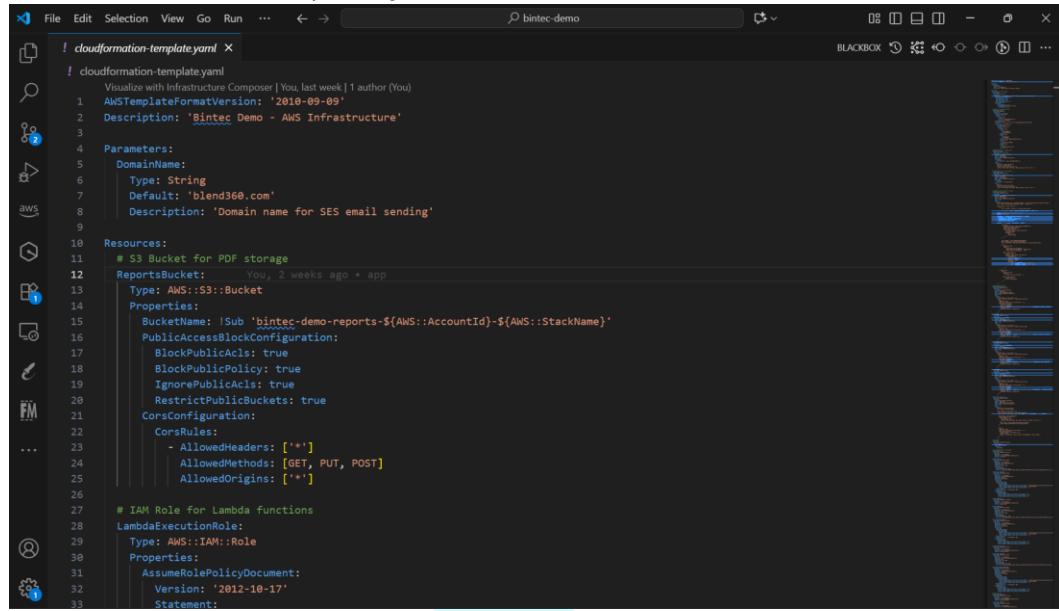
##### Paso 2.1: Revisar Template CloudFormation

**¿Qué vamos a hacer?** Entender qué vamos a desplegar.

**Acciones:**

###### 1. Abrir el archivo

## cloudformation-template.yaml



```
cloudformation-template.yaml
1 AWS::TemplateFormatVersion: '2010-09-09'
2 Description: 'Bintec Demo - AWS Infrastructure'
3
4 Parameters:
5   DomainName:
6     Type: String
7     Default: 'blend360.com'
8     Description: 'Domain name for SES email sending'
9
10 Resources:
11   # S3 Bucket for PDF storage
12   ReportsBucket: !Sub 'bintec-demo-reports-${AWS::AccountId}-${AWS::StackName}'
13   Type: AWS::S3::Bucket
14   Properties:
15     BucketName: !Sub 'bintec-demo-reports-${AWS::AccountId}-${AWS::StackName}'
16     PublicAccessBlockConfiguration:
17       BlockPublicAcls: true
18       BlockPublicPolicy: true
19       IgnorePublicAcls: true
20       RestrictPublicBuckets: true
21     CorsConfiguration:
22       CorsRules:
23         - AllowedHeaders: ['*']
24         AllowedMethods: [GET, PUT, POST]
25         AllowedOrigins: ['*']
26
27   # IAM Role for Lambda functions
28   LambdaExecutionRole:
29     Type: AWS::IAM::Role
30     Properties:
31       AssumeRolePolicyDocument:
32         Version: '2012-10-17'
33         Statement:
```

## 2. Revisar los recursos que se van a crear:

- 2 Funciones Lambda (Orquestador y Email)
- API Gateway con endpoints
- Roles y permisos IAM
- Outputs con URLs importantes

### 2.1. git checkout -b JuanTinjaca

```
PS C:\Users\nuvu-pc-00056\Desktop\bintec-demo> git checkout -b Juan_Tinjaca
Switched to a new branch 'Juan_Tinjaca'
```

## 2.2. Revisamos los parametros de dominio para el envío de correos

```
4 Parameters:
5   DomainName:
6     Type: String
7     Default: 'blend360.com'
8     Description: 'Domain name for SES email sending'
9
```

## 2.3. Configurar el correo que se usará para enviar.

```
64  # Lambda function for sending emails
65  SendEmailFunction:
66    Type: AWS::Lambda::Function
67    Properties:
68      FunctionName: bintec-send-email
69      Runtime: nodejs18.x
70      Handler: index.handler
71      Role: !GetAtt LambdaExecutionRole.Arn
72      Environment:
73        Variables:
74          SES_FROM_EMAIL: 'juan.tinjaca@blend360.com'
75      Code:
76        ZipFile: |
77          const AWS = require('aws-sdk');
78          const ses = new AWS.SES();
79          exports.handler = async (event) => {
80            return { statusCode: 200, body: JSON.stringify({ success: true }) };
81          };
82
```

## 2.4. git status

Para confirmar nuestros cambios

```
PS C:\Users\nuvu-pc-00056\Desktop\bintec-demo> git status
On branch Juan_Tinjaca
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   cloudformation-template.yaml

no changes added to commit (use "git add" and/or "git commit -a")
```

## 2.5. git add .

Para cargar nuestros cambios

```
PS C:\Users\nuvu-pc-00056\Desktop\bintec-demo> git add .
warning: in the working copy of 'cloudformation-template.yaml', LF will be replaced by CRLF the next time Git touches it
```

## 2.6. git commit -m "Mi despliegue bintec Nombre Apellido"

```
PS C:\Users\nuvu-pc-00056\Desktop\bintec-demo> git commit -m "despliegue de Juan Tinjaca"
[Juan_Tinjaca 8ce5984] despliegue de Juan Tinjaca
 1 file changed, 70 insertions(+), 65 deletions(-)
```

## 2.7. git push --set-upstream origin Nombre\_Apellido

```
PS C:\Users\nuvu-pc-00056\Desktop\bintec-demo> git push --set-upstream origin Juan_Tinjaca
```

## 2.8. Confirmamos actualización

```
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 850 bytes | 850.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'Juan_Tinjaca' on GitHub by visiting:
remote:     https://github.com/Zanat07/bintec-demo/pull/new/Juan_Tinjaca
remote:
To https://github.com/Zanat07/bintec-demo.git
 * [new branch]      Juan_Tinjaca -> Juan_Tinjaca
branch 'Juan_Tinjaca' set up to track 'origin/Juan_Tinjaca'.
```

### TAREA 3: Desplegar Stack

**¿Qué vamos a hacer?** Crear toda la infraestructura de backend automáticamente.

**Acciones:****1. Configurar secret token para despliegue (paso 2.2. Tarea 1)**

▼ Option 1: Set AWS environment variables

Paste the following text into PowerShell to set the AWS environment variables. [Learn more](#)

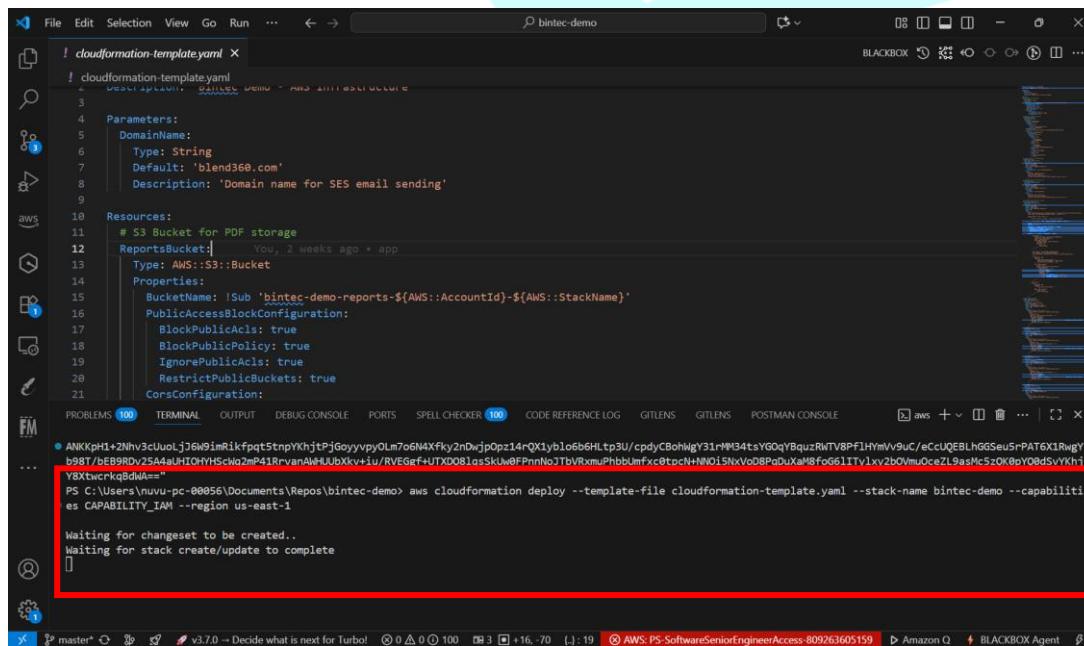
```
$Env:AWS_ACCESS_KEY_ID="ASIA47CRYL7WCULOCRYS"
$Env:AWS_SECRET_ACCESS_KEY="lERiLybstl45d51yybDV2WLDCv8pq0rVJi2ysi"
$Env:AWS_SESSION_TOKEN="IQoJb3JpZ2luX2VjEPb//////////wEaCXVzLWvhc3QtMSJGMEQCIFG5Yh8cbANKx/He5RHjtuv...
```

**2. Pegamos las variables con el secret token en nuestra consola**

```
PS C:\Users\nuvu_pc-00056\Desktop\bintec-demo> $Env:AWS_ACCESS_KEY_ID="ASIA47CRYL7WCULOCRYS"
PS C:\Users\nuvu_pc-00056\Desktop\bintec-demo> $Env:AWS_SECRET_ACCESS_KEY="lERiLybstl45d51yybDV2WLDCv8pq0rVJi2ysi"
PS C:\Users\nuvu_pc-00056\Desktop\bintec-demo> $Env:AWS_SESSION_TOKEN="IQoJb3JpZ2luX2VjEPb//////////wEaCXVzLWvhc3QtMSJGMEQCIFG5Yh8cbANKx/He5RHjtuv...
```

**3. Ejecutar despliegue:**

```
aws cloudformation deploy --template-file cloudformation-template.yaml --stack-name bintec-demo --capabilities CAPABILITY_IAM --region us-east-1
```

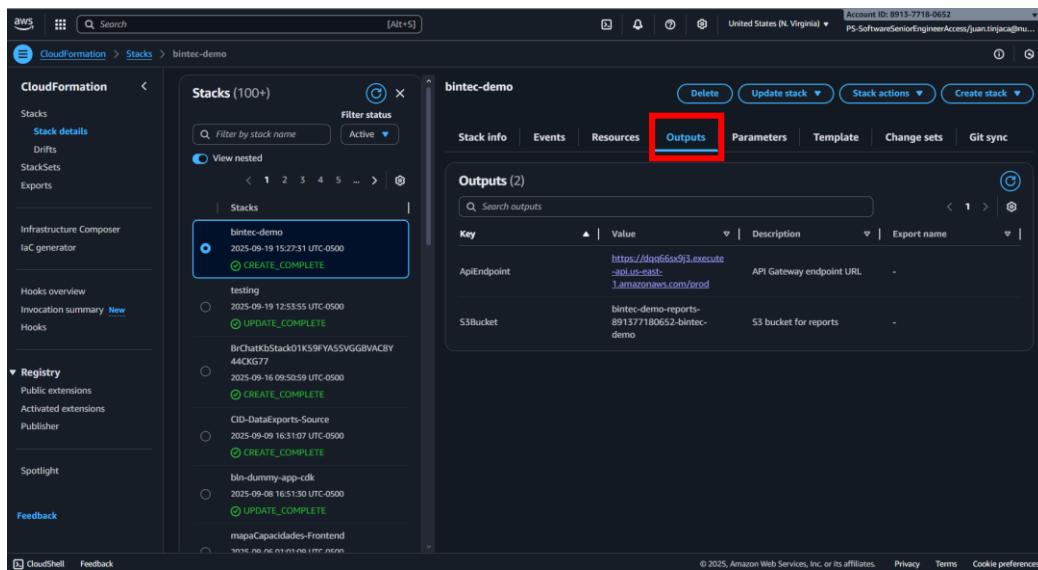
**4. Esperar confirmación (5-8 minutos)****Paso 3.1: Verificar Recursos Creados**

**¿Qué vamos a hacer?** Confirmar que todo se creó correctamente.

**Acciones:****1. Ir a la consola AWS → CloudFormation**

2. Buscar el stack bintec-demo-stack

3. Ir a la pestaña **Outputs**



Key	Value	Description	Export name
ApiEndpoint	https://d0966a9f2.execute-api.us-east-1.amazonaws.com/prod	API Gateway endpoint URL	-
S3Bucket	bintec-demo-reports-891377180652-bintec-demo	S3 bucket for reports	-

4. **COPiar** la ApiEndPoint (la necesitaremos después)

## TAREA 4: Configurar Amazon Bedrock

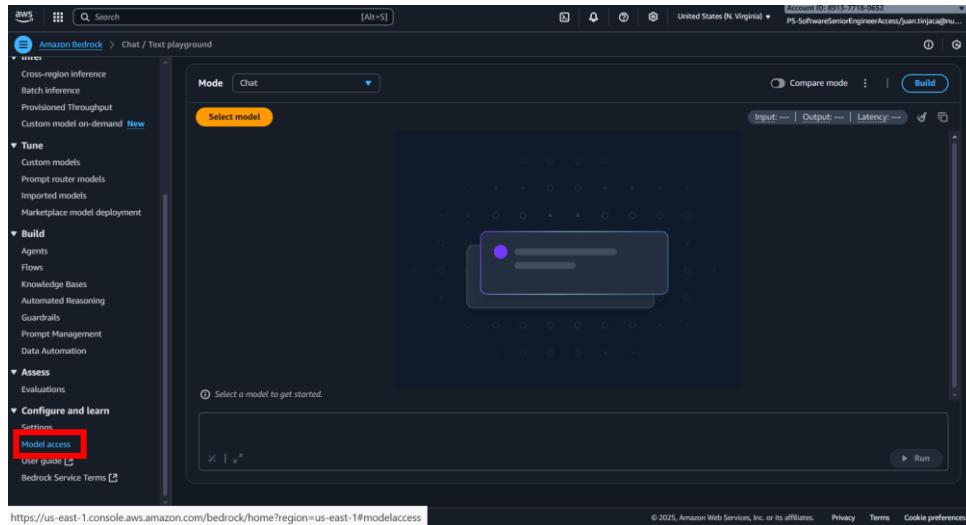
*Tiempo estimado: 5 minutos*

### Paso 4.1: Habilitar Modelos IA

**¿Qué vamos a hacer?** Activar Claude 3 para nuestros agentes.

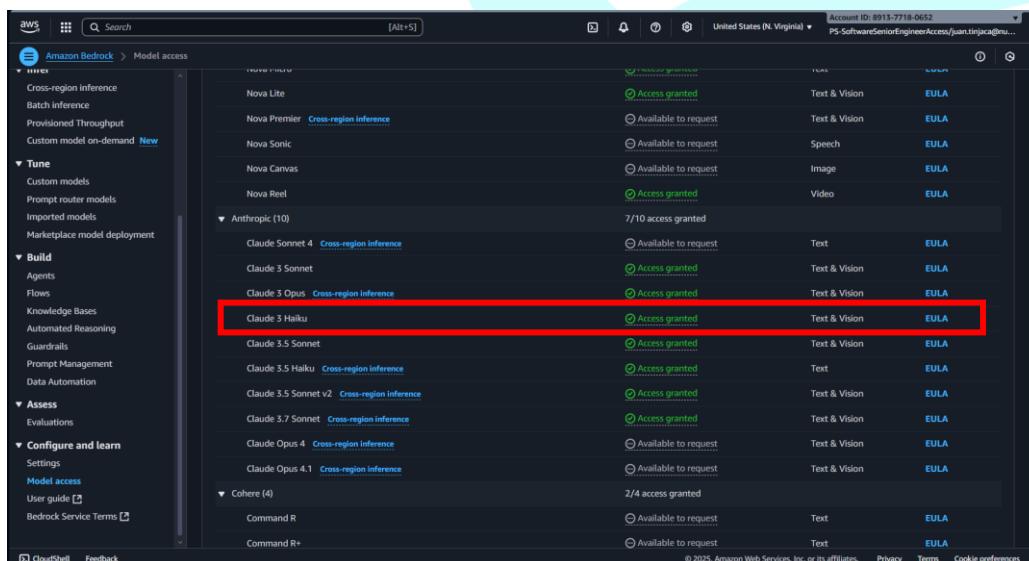
## Acciones:

1. Consola AWS → **Amazon Bedrock**
2. Menú izquierdo → **Model access**



The screenshot shows the Amazon Bedrock console with the 'Model access' section selected. The sidebar on the left has a red box around the 'Model access' link under the 'Configure and learn' section. The main area shows a 'Select a model to get started.' message and a 'Run' button.

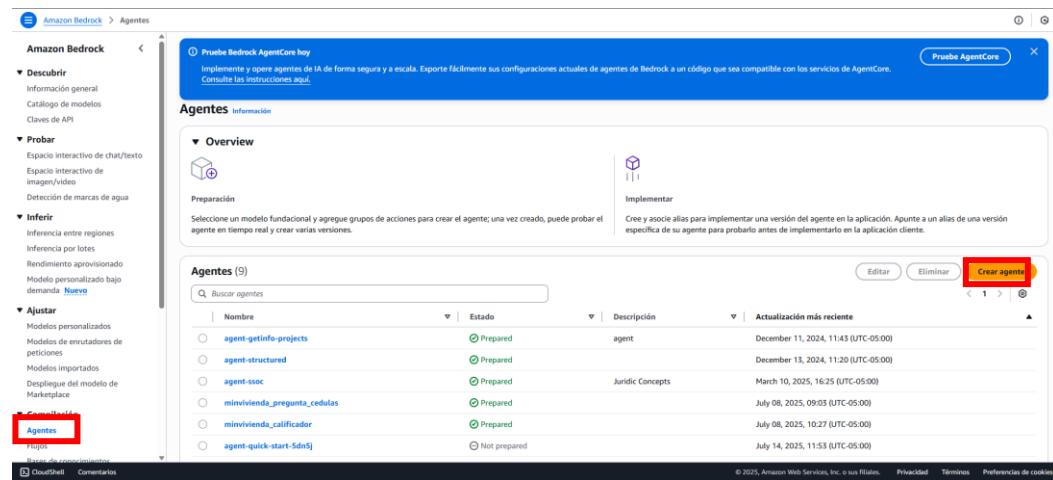
3. **Request model access**
4. **Seleccionar Anthropic Claude 3 Haiku**



The screenshot shows the 'Model access' section of the Amazon Bedrock console. The 'Anthropic (10)' section is expanded, showing various models. The 'Claude 3 Haiku' model is highlighted with a red box. The table below lists the models and their access status.

Model	Type	Access Status	Region
Nova Lite	Cross-region inference	Access granted	Text & Vision
Nova Premier	Cross-region inference	Available to request	Text & Vision
Nova Sonic	Cross-region inference	Available to request	Speech
Nova Canvas	Cross-region inference	Available to request	Image
Nova Reel	Cross-region inference	Access granted	Video
Claude 3 Haiku	Cross-region inference	Access granted	Text & Vision
Claude 3.5 Sonnet	Cross-region inference	Access granted	Text & Vision
Claude 3.5 Haiku	Cross-region inference	Access granted	Text
Claude 3.5 Sonnet v2	Cross-region inference	Access granted	Text & Vision
Claude 3.7 Sonnet	Cross-region inference	Access granted	Text & Vision
Claude Opus 4	Cross-region inference	Available to request	Text & Vision
Claude Opus 4.1	Cross-region inference	Available to request	Text & Vision
Command R	Cross-region inference	Available to request	Text
Command R+	Cross-region inference	Available to request	Text

5. **Request model access**
6. **Crear agente regulatorio -> Agentes -> Crear agente**



Prueba Bedrock AgentCore hoy

Implemente y ejecute agentes de IA de forma segura y a escala. Exporte fácilmente sus configuraciones actuales de agentes de Bedrock a un código que sea compatible con los servicios de AgentCore. Consulte las instrucciones aquí.

Prueba AgentCore

Agentes

Preparación

Implementar

Crear y asocie alias para implementar una versión del agente en la aplicación. Apunte a un alias de una versión específica de su agente para probarlo antes de implementarlo en la aplicación cliente.

Agents (9)

Nombre Estado Descripción Actualización más reciente

Nombre	Estado	Descripción	Actualización más reciente
agent-getinfo-projects	Prepared	agent	December 11, 2024, 11:43 (UTC-05:00)
agent-structured	Prepared		December 13, 2024, 11:20 (UTC-05:00)
agent-sscc	Prepared	Juridic Concepts	March 10, 2025, 16:25 (UTC-05:00)
minivivienda_pregunta_cedulas	Prepared		July 08, 2025, 09:03 (UTC-05:00)
minivivienda_calificador	Prepared		July 08, 2025, 10:27 (UTC-05:00)
agent-quick-start-5dn5j	Not prepared		July 14, 2025, 11:53 (UTC-05:00)

Editar Eliminar Crear agente

## Crear agente



### Nombre

Los caracteres válidos son a-z, A-Z, 0-9, \_ (guión bajo) y - (guion). El nombre puede tener hasta 100 caracteres

### Descripción - opcional

La descripción puede tener hasta 200 caracteres.

### Colaboración entre múltiples agentes

Más información sobre la colaboración entre múltiples agentes [\[ \]](#)

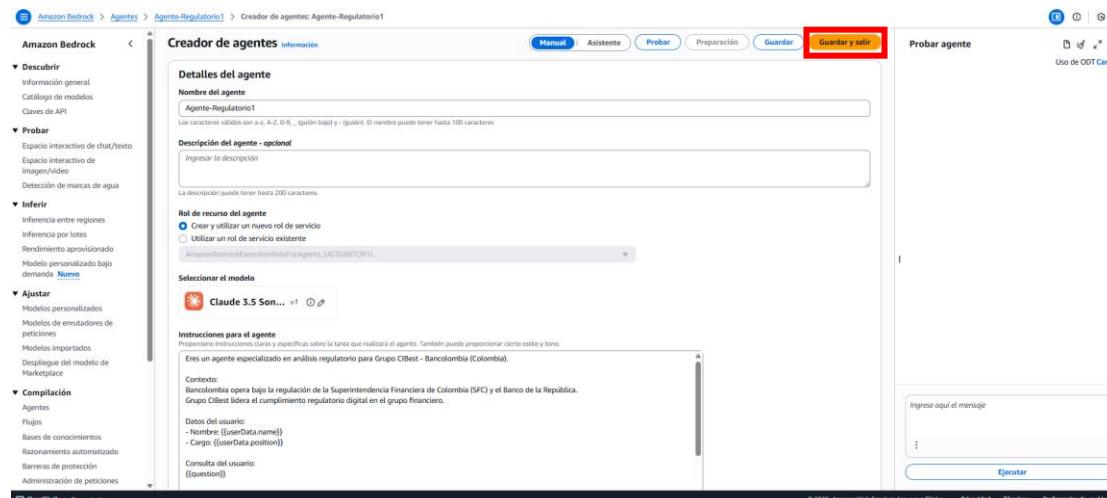
Habilitar la colaboración entre múltiples agentes

Si tiene más de un agente, la colaboración entre múltiples agentes permite a este agente asociar a los demás como agentes colaboradores para orquestar respuestas. Podrá modificar esta opción más adelante.

Cancelar

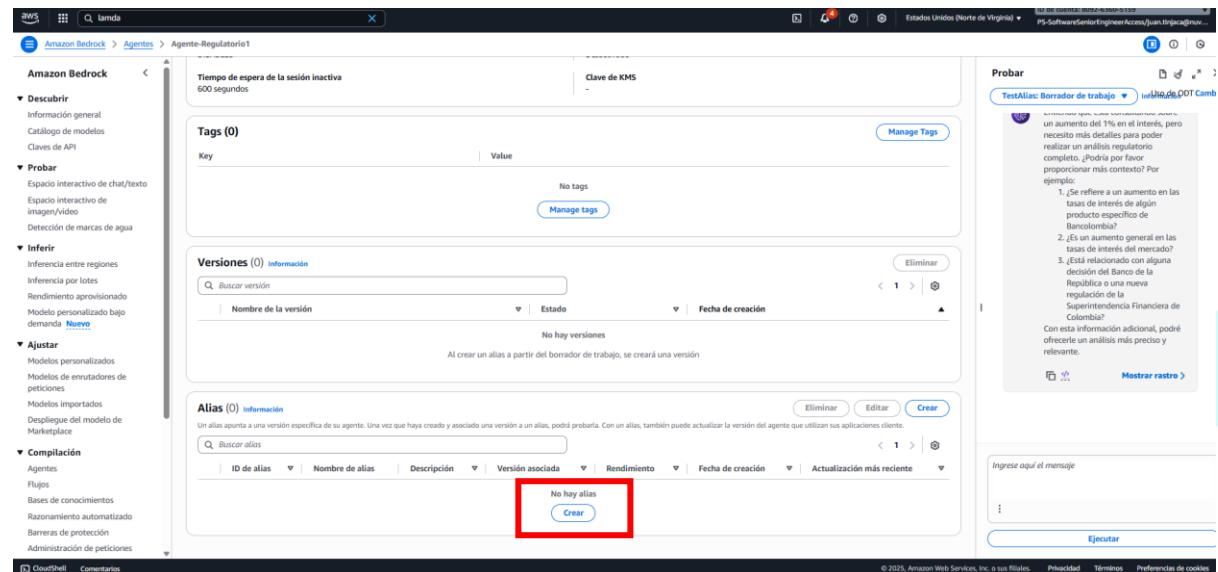
Crear

## 7. Configurar agente -> Pegamos el prompt en “instrucciones para el agente”



- Eres un agente especializado en análisis regulatorio para Grupo CIBest - Bancolombia (Colombia).
- Contexto: Bancolombia opera bajo la regulación de la Superintendencia Financiera de Colombia (SFC) y el Banco de la República. Grupo CIBest lidera el cumplimiento regulatorio digital en el grupo financiero.
- Instrucciones: Genera un análisis regulatorio detallado siguiendo **EXACTAMENTE** la siguiente estructura y formato:
- **EVALUACIÓN INICIAL:**
- Describe el marco regulatorio actual aplicable a Bancolombia según la SFC y el Banco de la República.
- Menciona las normativas, circulares y disposiciones más relevantes para el tipo de operación o tema consultado.
- **IDENTIFICACIÓN DE GAPS:**
- Señala las posibles brechas regulatorias frente a las normas de la SFC y actualizaciones pendientes que podrían afectar a Bancolombia.
- Identifica áreas de riesgo o cumplimiento parcial.
- **PROYECCIÓN:**
- Explica el impacto regulatorio proyectado en Colombia, indicando porcentajes estimados de afectación, nivel de cumplimiento esperado y cronograma tentativo según la SFC.
- Incluye posibles cambios normativos próximos o tendencias regulatorias relevantes.
- **RECOMENDACIONES:**
- Propón acciones concretas que el **Grupo CIBest** y el rol de **{{userData.position}}** deberían considerar para fortalecer el cumplimiento y mitigar riesgos regulatorios.
- Fundamenta las recomendaciones en los marcos SARLAFT, circulares de la SFC y normas aplicables del Banco de la República.
- Notas:
- Menciona siempre referencias específicas a circulares o resoluciones SFC.
- Incluye aspectos de SARLAFT, gestión de riesgos y cumplimiento financiero.
- Usa un tono profesional, técnico y normativo.

## 8. Creamos el Alias para identificar el agente y probamos.



The screenshot shows the AWS Lambda console with the function 'Agente-Regulatorio1' selected. The 'Alias' section is highlighted with a red box, showing the 'No hay alias' (No alias) message and the 'Crear' (Create) button. The 'Crear alias' button is also highlighted with a red box.

## Crear alias

### Nombre de alias

Agente-regulatorio-alias

Los caracteres válidos son a-z, A-Z, 0-9, \_ (guión bajo) y - (guion). El nombre puede tener hasta 100 caracteres

### Descripción - opcional

Ingresar la descripción

Los caracteres válidos son a-z, A-Z, 0-9, \_ (guion bajo) y - (guion). El nombre puede tener hasta 200 caracteres

### Asociar una versión

- Crear una nueva versión y asociarla a este alias.
- Utilizar una versión existente para asociar este alias.

### Seleccionar rendimiento

El borrador de trabajo actual utiliza el modelo Claude 3.5 Sonnet. Seleccione el rendimiento para este modelo a continuación.

- On-demand (ODT)
- Provisioned Throughput (PT)

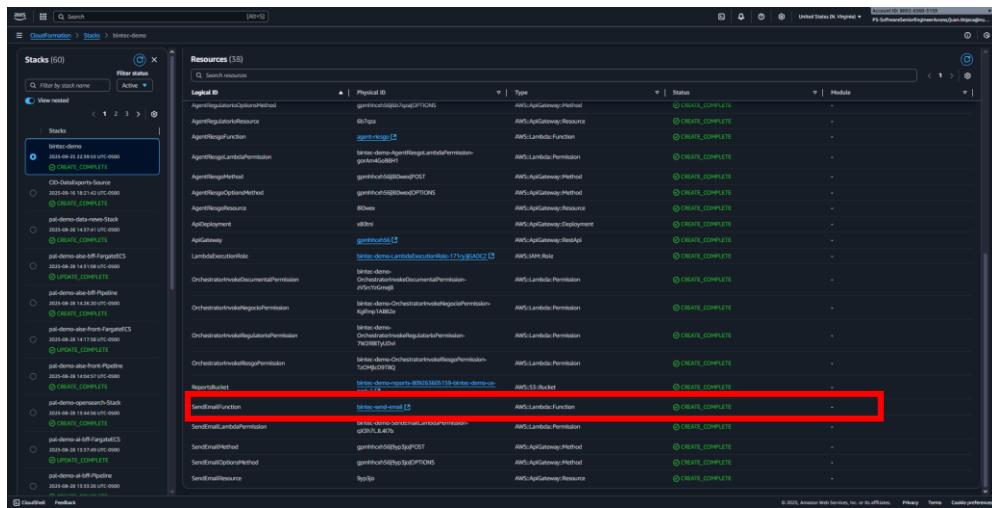
[Cancelar](#)

[Crear alias](#)

## TAREA 5: Configurar Lambda de Emails para envíos y Agente Regulatorio

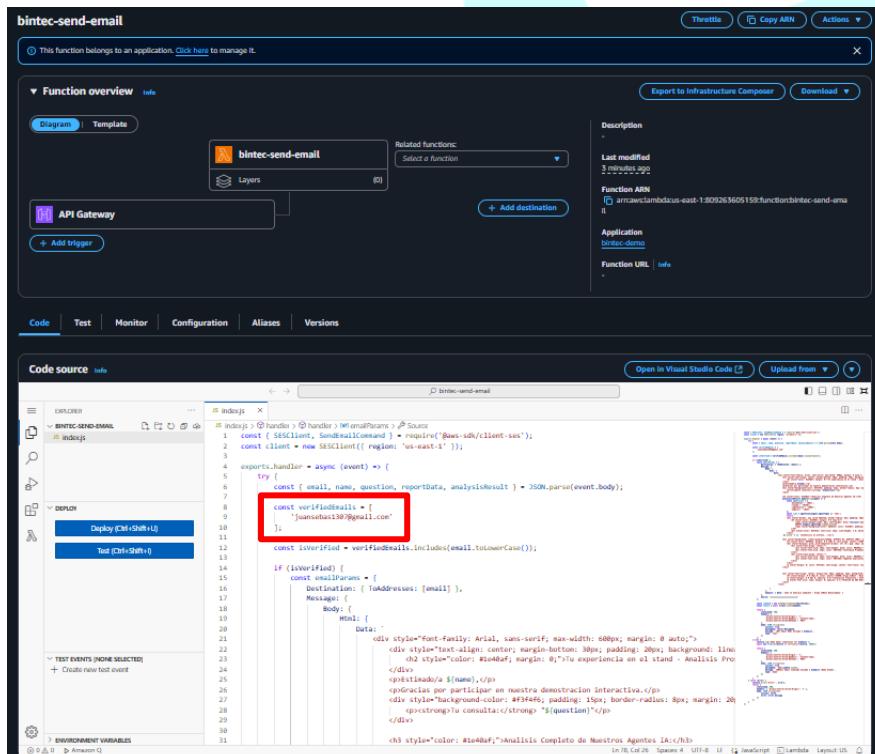
Tiempo estimado: 8 minutos

## Paso 5.1: Volvemos a nuestro stack de cloud formation y buscamos SendEmailFunction

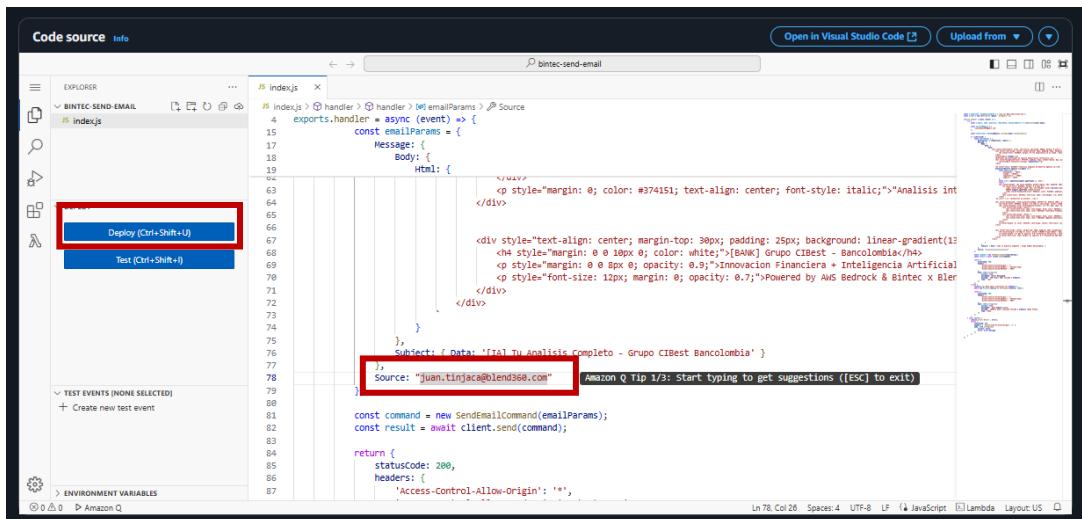


## Paso 5.2: abrimos la función y damos click en el archivo index

cambiamos el correo por el que usaremos en el workshop



## Paso 5.3: Configuramos el correo de envío y desplegamos la función



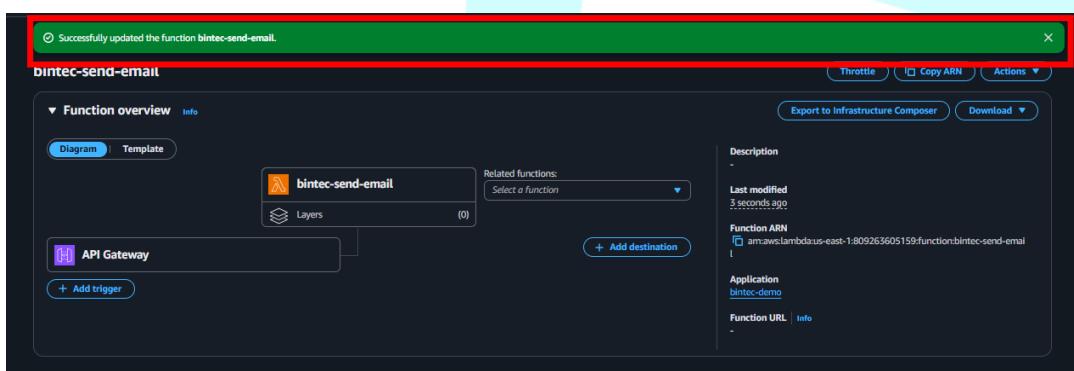
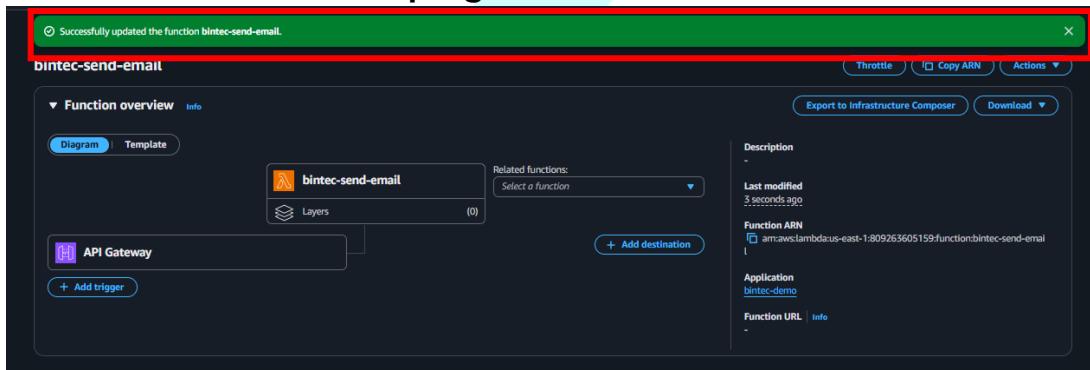
```

  exports.handler = async (event) => {
    const emailParams = {
      Message: {
        Body: {
          Html: `

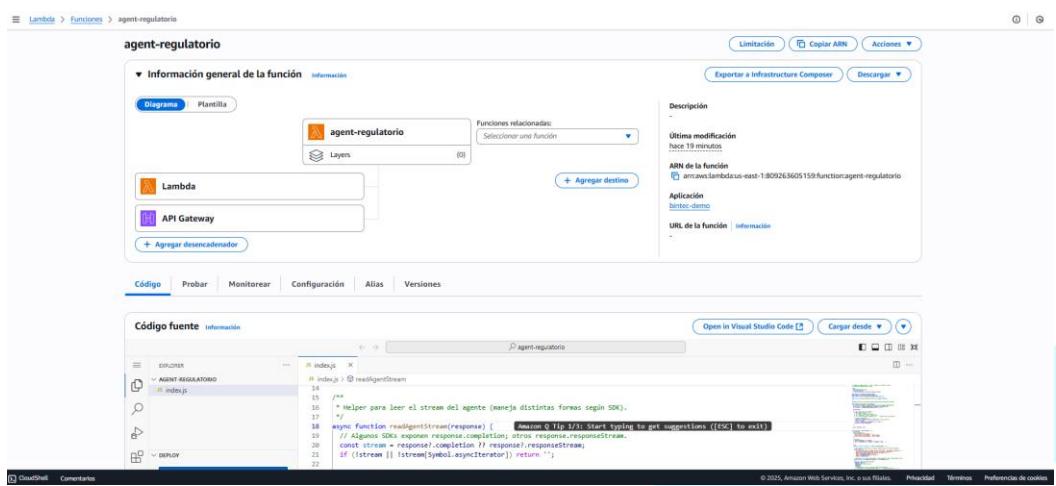
          <div style="margin: 0; color: #374151; text-align: center; font-style: italic;">Analisis int
          </div>
          <div style="text-align: center; margin-top: 30px; padding: 25px; background: linear-gradient(13
          <h4 style="margin: 0 0 10px 0; color: white;">[BANK] Grupo CIBest - Bancolombia</h4>
          <p style="margin: 0 0 8px 0; opacity: 0.9;">Innovación Financiera + Inteligencia Artificial
          <p style="font-size: 12px; margin: 0; opacity: 0.7;">Powered by AWS Bedrock & Bintec x Bler
          </div>
        },
        Subject: 'Final Tu Analisis Completo - Grupo CIBest Bancolombia',
        Source: 'juan.tinjaca@blend360.com'
      }
    }
    const command = new SendEmailCommand(emailParams);
    const result = await client.send(command);

    return {
      statusCode: 200,
      headers: {
        'Access-Control-Allow-Origin': '*'
      }
    }
  }
}
  
```

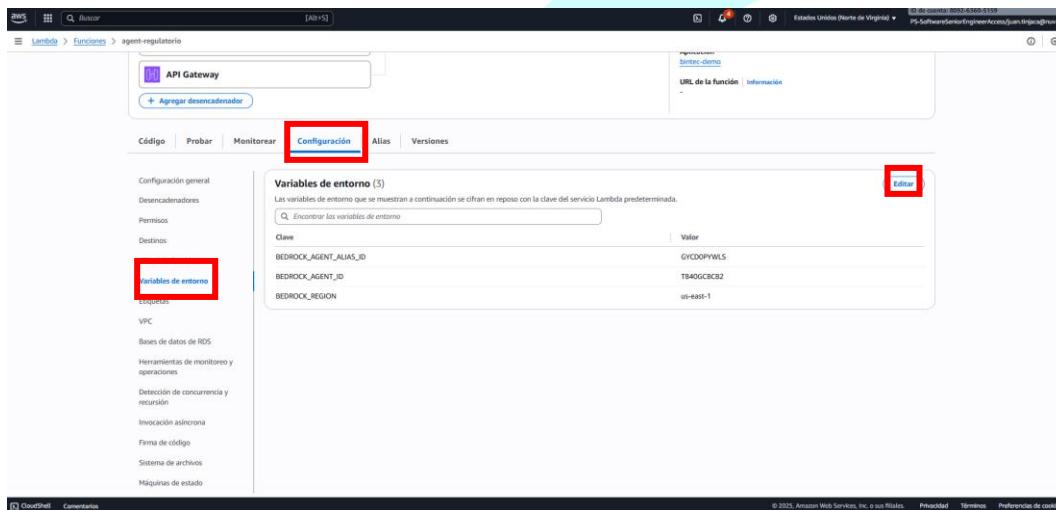
## Paso 5.4: Validamos el despliegue de la función



## Paso 5.5. Abrimos la lambda del Agente regulatorio



## Paso 5.6. Configuramos las variables de entorno



## Paso 5.7. Copiamos el Id y el Alias Id del agente (Tarea 4)

**Descripción general del agente**

Nombre	Q4P3DMTMGN
Descripción	-
Fecha de creación	October 11, 2025, 19:27 (UTC-05:00)
Permisos	arn:aws:iam::809263605159:role/service-role/AmazonBedrockExecutionRoleForAgents_UGTG88TOR1L
Entrada de usuario	DISABLED
Tiempo de espera de la sesión inactiva	600 segundos

**Tags (0)**

Key	Value
-----	-------

No tags

[Manage tags](#)

**Versiones (1) Información**

Nombre de la versión	Estado	Fecha de creación
Versión 1	PREPARED	October 11, 2025, 19:37 (UTC-05:00)

[Eliminar](#) [1](#) [>](#) [?](#)

**Alias (1) Información**

Un alias apunta a una versión específica de su agente. Una vez que haya creado y asociado una versión a un alias, podrá probarla. Con un alias, también puede actualizar la versión del agente que utilizan sus aplicaciones cliente.

ID de alias	Nombre de alias	Descripción	Versión asociada	Rendimiento	Fecha de creación	Actualización más reciente
WK0UMNR26D	Agente-regulatorio-alias	-	Versión 1	-	October 11, 2025, 19:37 (UTC-05:00)	October 11, 2025, 19:37 (UTC-05:00)

[Eliminar](#) [Editar](#) [Crear](#) [<](#) [1](#) [>](#) [?](#)

**Variables de entorno**

Puede definir variables de entorno como pares clave-valor a los que se puede obtener acceso desde el código de función. Son útiles para almacenar los ajustes de configuración sin necesidad de cambiar el código de función. [Más información](#)

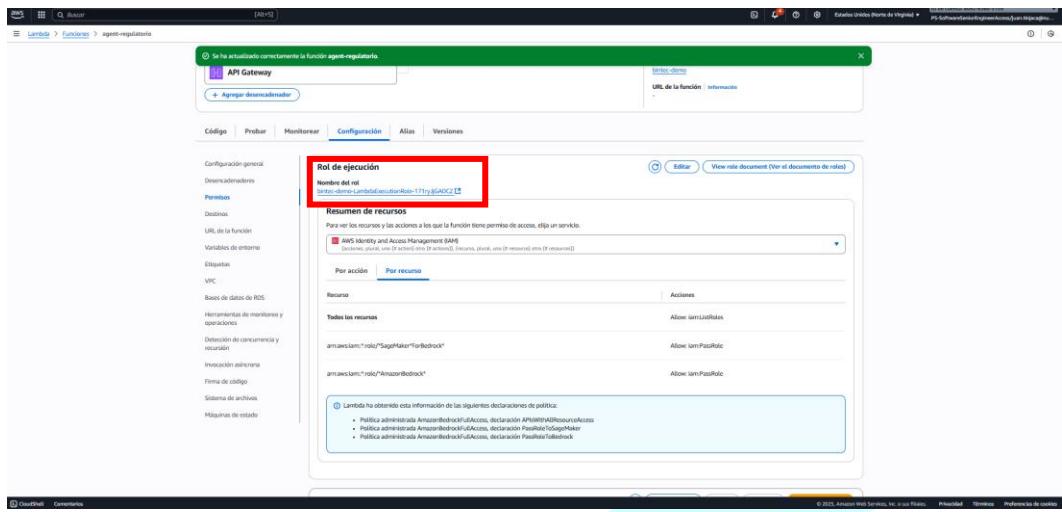
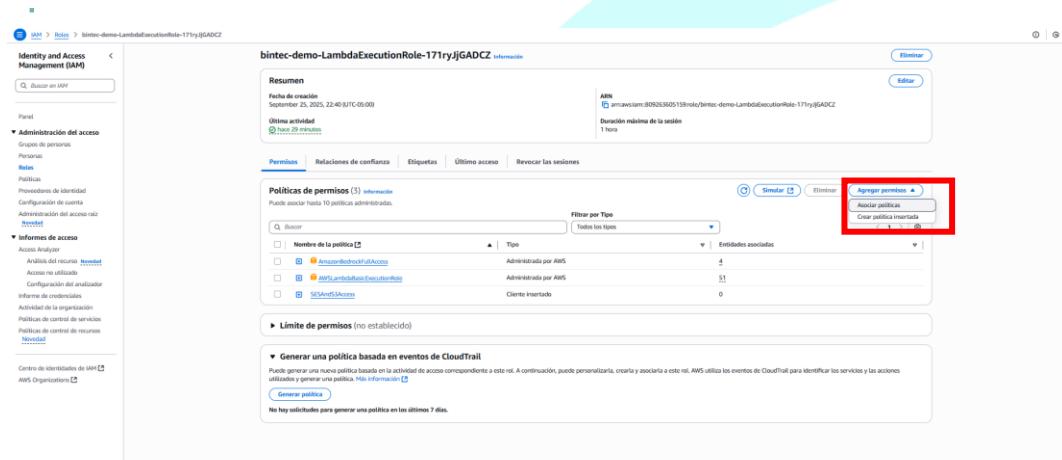
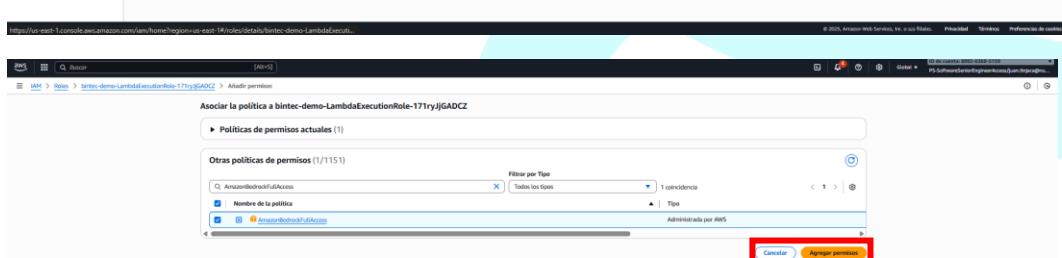
Clave	Valor	Eliminar
BEDROCK_AGENT_ALIAS_ID	WK0UMNR26D	<a href="#">Eliminar</a>
BEDROCK_AGENT_ID	Q4P3DMTMGN	<a href="#">Eliminar</a>
BEDROCK_REGION	us-east-1	<a href="#">Eliminar</a>

[Agregar variable de entorno](#)

**Configuración de cifrado**

[Cancelar](#) [Guardar](#)

## Paso 5.8. Permitimos que se comunique la lambda y bedrock

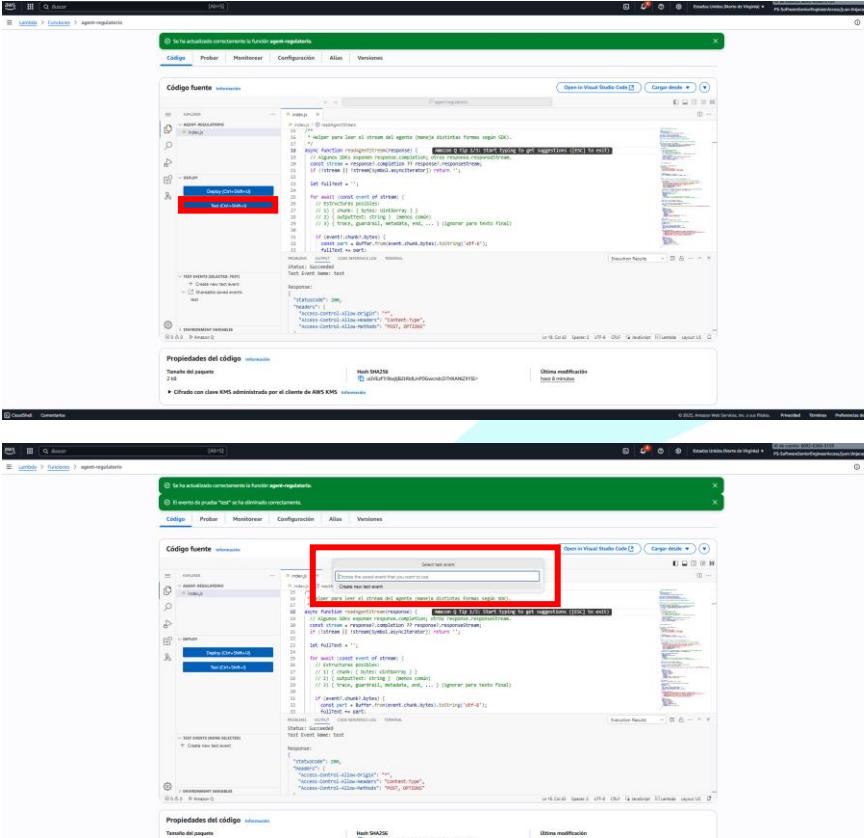




## Paso 5.9. Creamos un caso de prueba y ejecutamos la lambda

{

"question": "aumentar en un 1% el credito de mis clientes"

}



The screenshot shows the AWS Lambda console with two versions of the function 'agent-registrations' listed. The top version is the latest, and the bottom version is the previous one. Both versions have a red box highlighting the 'Deploy (Ctrl+Shift+D)' button in the 'Actions' column. The bottom version also has a red box highlighting the 'Create test event' button in the 'Test' section of the function details page.

## Paso 6.0. Guardamos y ejecutamos el test

## TAREA 6: Configurar Amazon SES

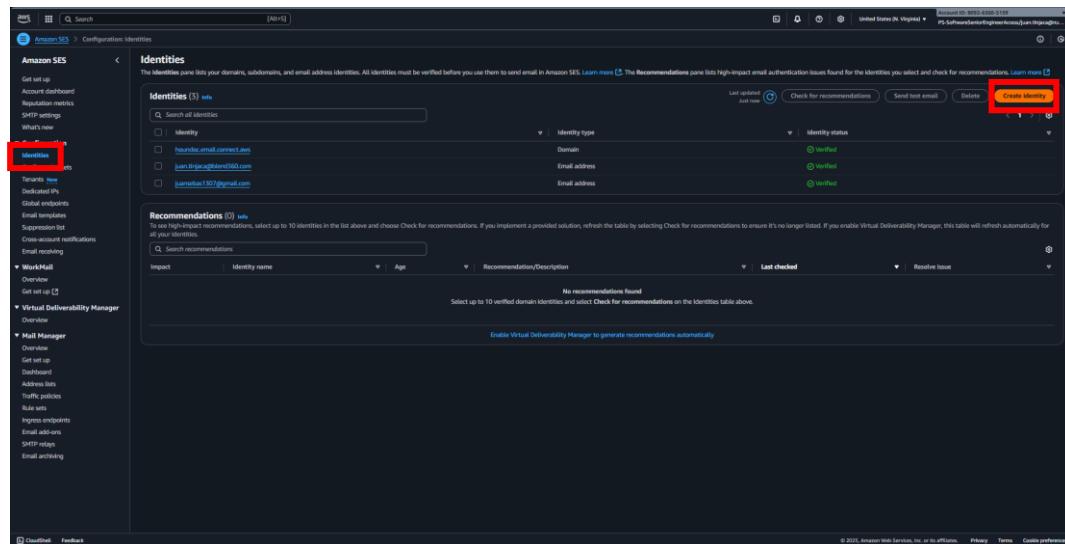
**Tiempo estimado: 10 minutos**

## Paso 6.1: Verificar Dominio de Email

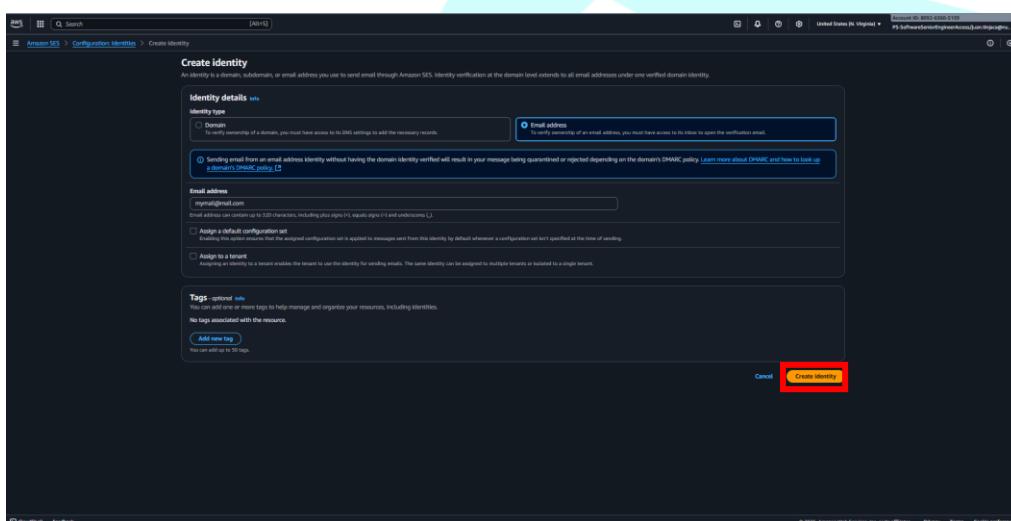
## ¿Qué vamos a hacer? Configurar el servicio de emails.

### Acciones:

1. Consola AWS → Amazon SES
2. Verified identities → Create identity



3. Identity type: Email address
4. Email address: workshop-demo@[tu-dominio].com (usar email personal para pruebas)
5. Create identity



6. Verificar email en la bandeja de entrada

Amazon Web Services – Email Address Verification Request in region US East (N. Virginia) Recibirás x

Amazon Web Services [+no-reply-aws@amazon.com](#) para mí 19:11 ( hace 9 minutos )

Dear Amazon Web Services Customer:

We have received a request to authorize this email address for use with Amazon SES and Amazon Pinpoint in region US East (N. Virginia). If you requested this verification, please go to the following URL to confirm that you are authorized to use this email address:

<https://email-verification.us-east-1.amazonaws.com/ConfirmIdentity?Signature=07fa5a252d6193add0509a748c0a3007e200e1415e7948395101620f>

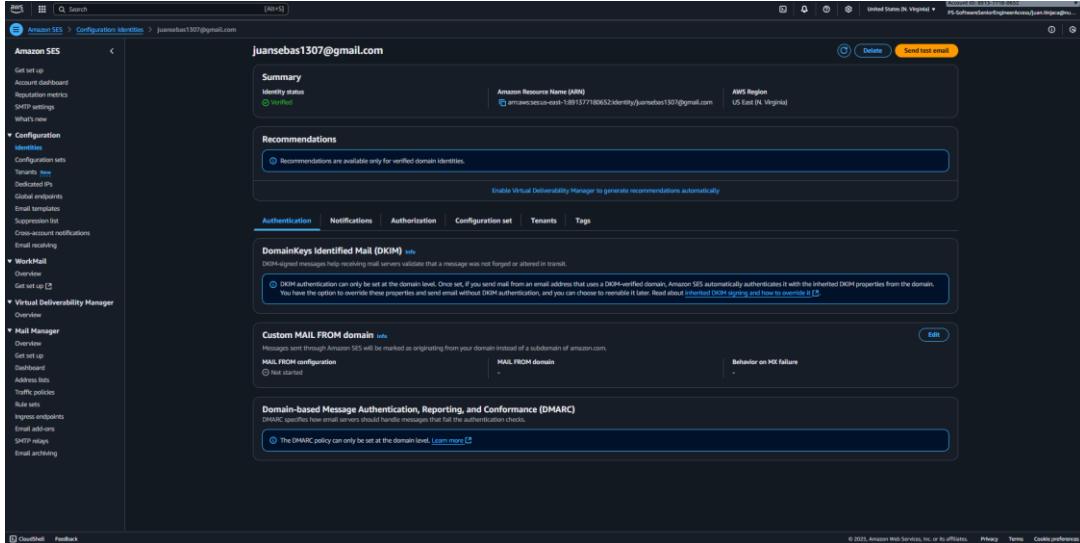
Your request will not be processed unless you confirm the address using this URL. This link expires 24 hours after your original verification request.

If you did NOT request to verify this email address, do not click on the link. Please note that many times, the situation isn't a phishing attempt, but either a misunderstanding of how to use our service, or someone setting up email-sending capabilities on your behalf as part of a legitimate service, but without having fully communicated the procedure first.

To learn more about sending email from Amazon Web Services, please refer to the Amazon SES Developer Guide at <http://docs.aws.amazon.com/ses/latest/DeveloperGuide>Welcome.html> and Amazon Pinpoint Developer Guide at <http://docs.aws.amazon.com/pinpoint/latest/userguide/welcome.html>.

Sincerely,

The Amazon Web Services Team.

A screenshot of the Amazon SES Configuration Details page for the email address juansebas1307@gmail.com. The page shows the following sections: Summary (Identity status: Verified, ARN: arn:aws:ses:us-east-1:891377180652:identity/juansebas1307@gmail.com, Region: US East (N. Virginia)), Recommendations (Recommendations are available only for verified domain identities), Authentication (DKIM (DomainKeys Identified Mail) is enabled, DKIM authentication can only be set at the domain level), Notifications, Authorization, Configuration set, Tenants, and Tags. The left sidebar includes sections for Get set up, Account dashboard, Billing metrics, SMTP settings, What's new, Configuration (Metadata, Configuration sets, Tenants, Dedicated IPs, Global endpoints, Email endpoints, Email endpoint, Configuration notifications, Email receiving), WorkMail (Overview, Get set up), and Virtual Delivery Manager (Overview, Mail Manager, Overview, Get set up, Dashboard, Address lists, Traffic policies, Rule sets, Incoming endpoints, Email add-ons, SFTP relay, Email anchoring). The bottom of the page includes CloudWatch Feedback, a copyright notice (© 2021, Amazon Web Services, Inc. or its affiliates.), and links to Privacy, Terms, and Cookies preferences.

## TAREA 7: Configurar Frontend y Variables

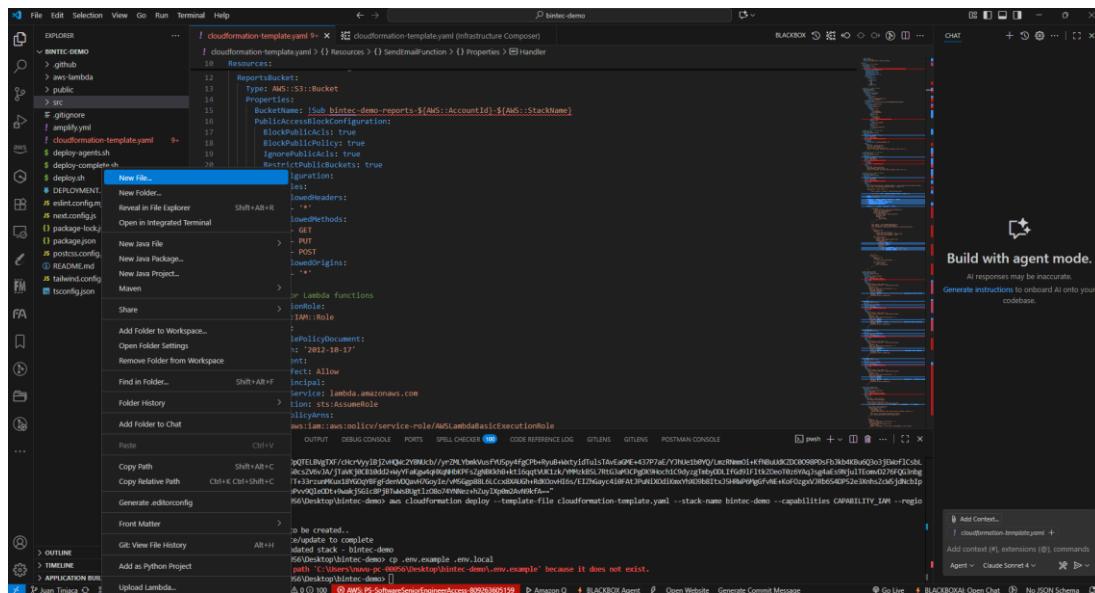
**Tiempo estimado: 8 minutos**

## Paso 7.1: Actualizar Configuración Frontend

**¿Qué vamos a hacer?** Conectar frontend con backend desplegado.

Acciones:

### 1. En el directorio del proyecto:

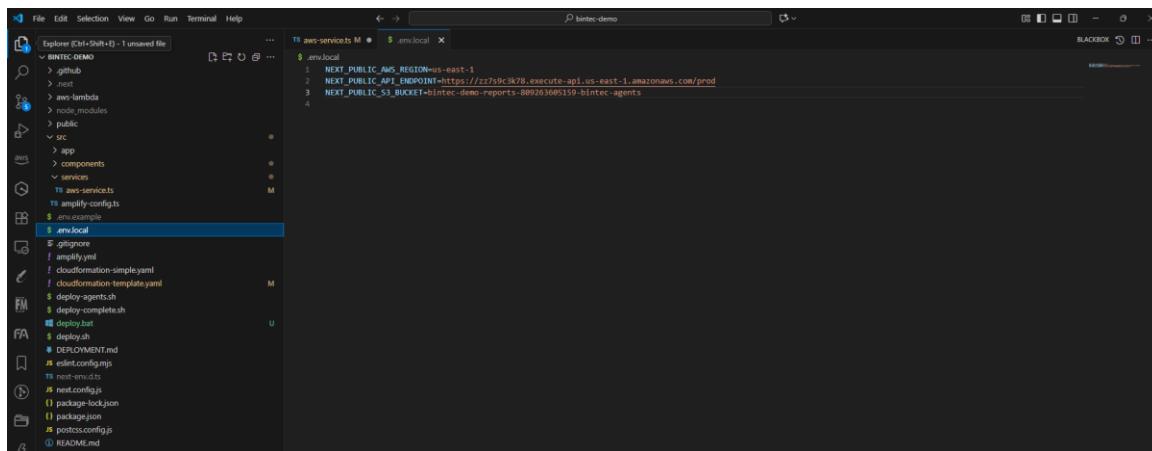


### 2. Crear archivo .env.local:

- **NEXT\_PUBLIC\_API\_ENDPOINT=https://[TU-API-ID].execute-api.us-east-1.amazonaws.com/prod**  
*(Usar la URL del paso 2.3)*
- **NEXT\_PUBLIC\_AWS\_REGION=us-east-1**
- **NEXT\_PUBLIC\_S3\_BUCKET=bintec-demo-reports-891377180652-bintec-demo**

### 3. Instalar dependencias:

**npm install**



```

$ aws-service.ts M • $ env.local
$ env.local
1 NEXT_PUBLIC_AWS_REGION=us-east-1
2 NEXT_PUBLIC_API_ENDPOINT=https://zz7s9c3k78.execute-api.us-east-1.amazonaws.com/prod
3 NEXT_PUBLIC_S3_BUCKET=bintec-demo-reports-809263685159-bintec-agents
4

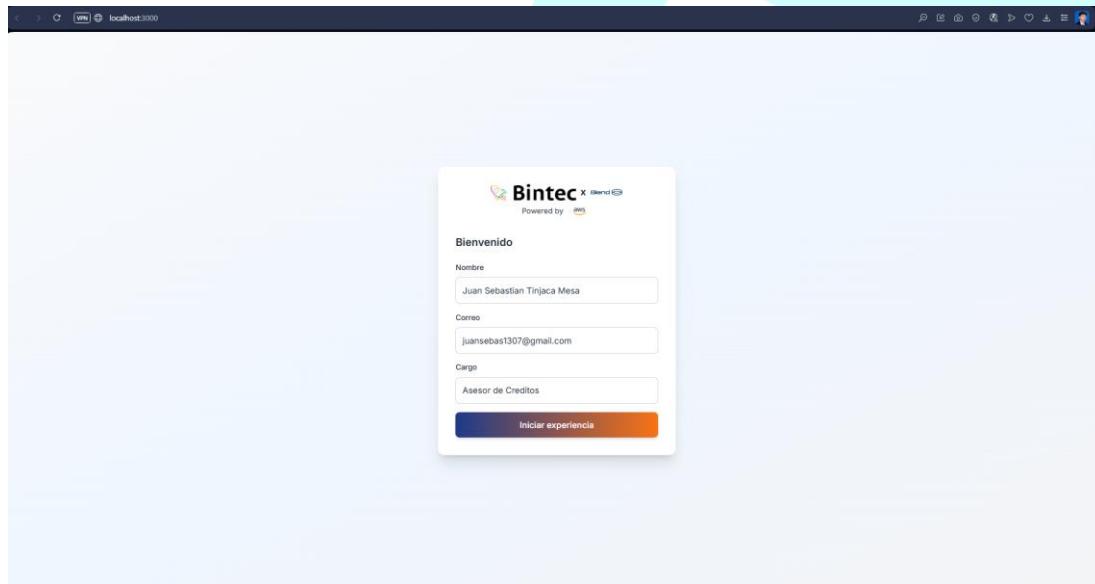
```

## Paso 7.2: Probar Localmente

**¿Qué vamos a hacer?** Verificar integración completa.

### Acciones:

1. **Ejecutar aplicación:** npm run dev
2. **Abrir:** <http://localhost:3000>
3. **Probar flujo completo:**
  - a. **Pregunta → Agentes IA**
  - b. **Email → SES**



## TAREA 8: Desplegar en AWS Amplify

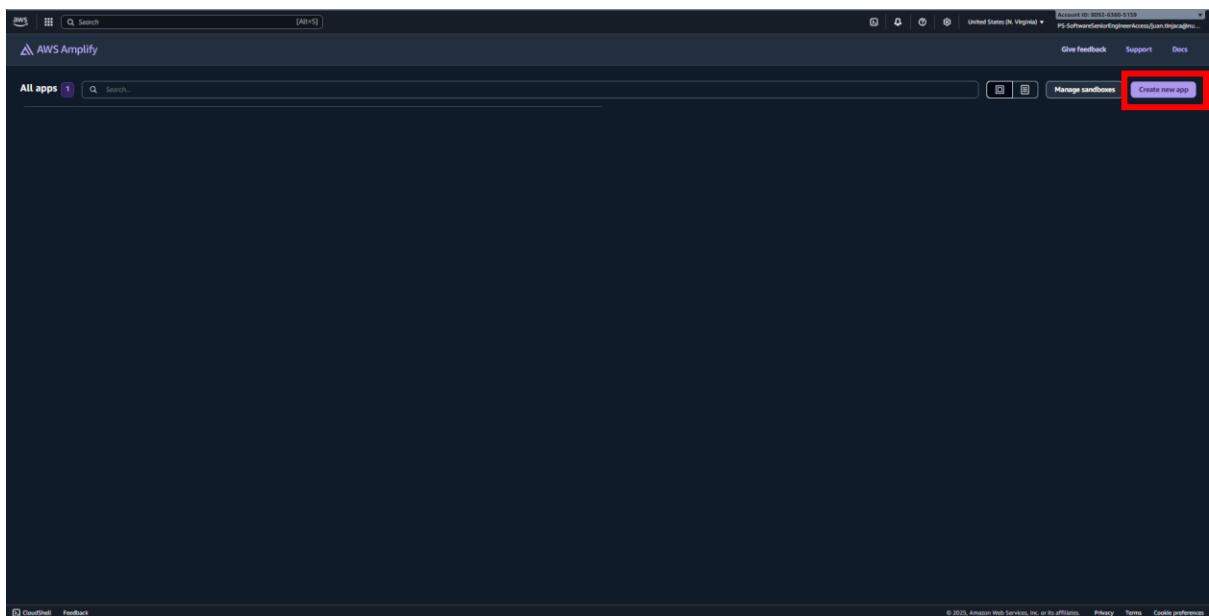
*Tiempo estimado: 12 minutos*

## Paso 8.1: Configurar Amplify Hosting

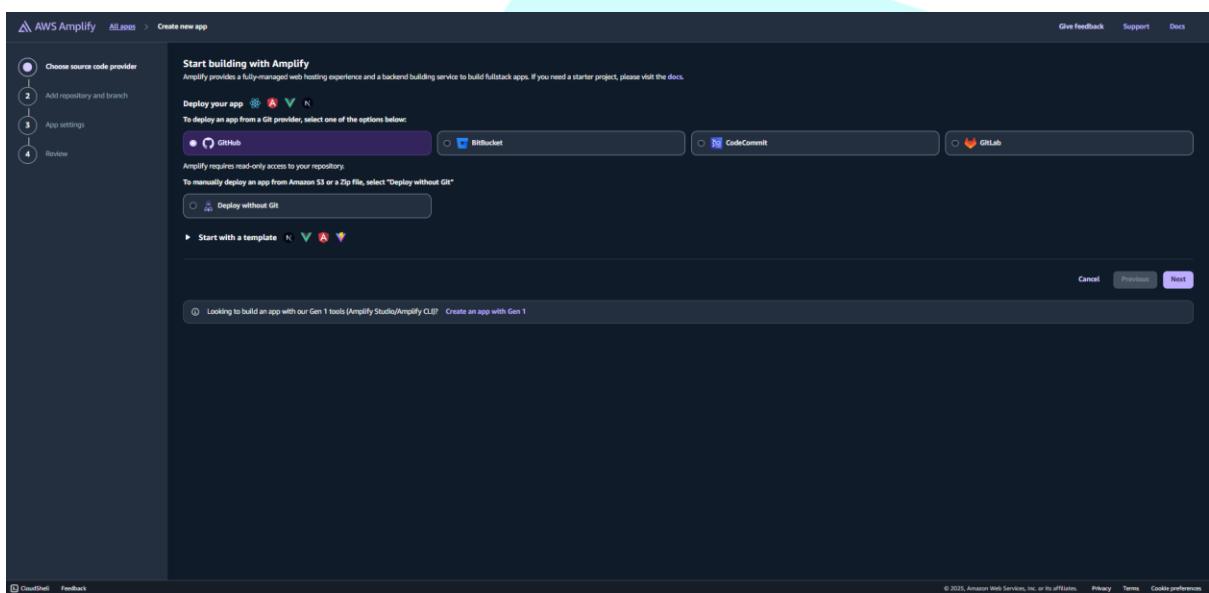
**¿Qué vamos a hacer?** Subir la aplicación a la nube.

**Acciones:**

1. Consola AWS → **AWS Amplify**
2. **New app** → **Host web app**



### 3. Deploy Github

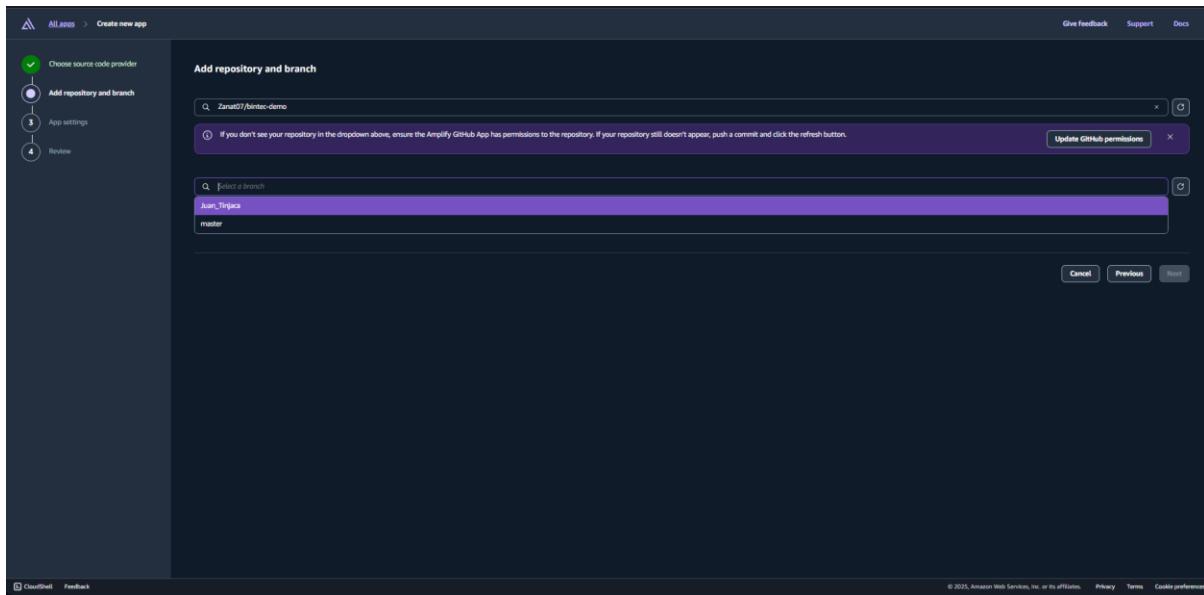


#### 4. App name:

**bintec-demo**

#### 5. Environment name:

**Prod**

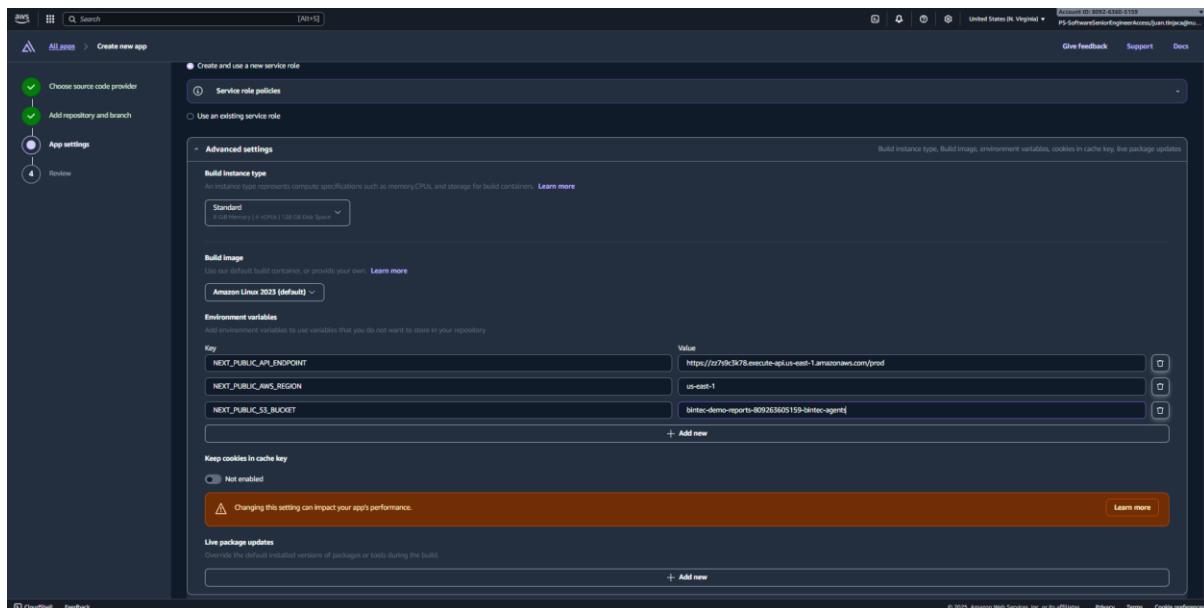


## Paso 8.2: Configurar Variables de Entorno en Amplify

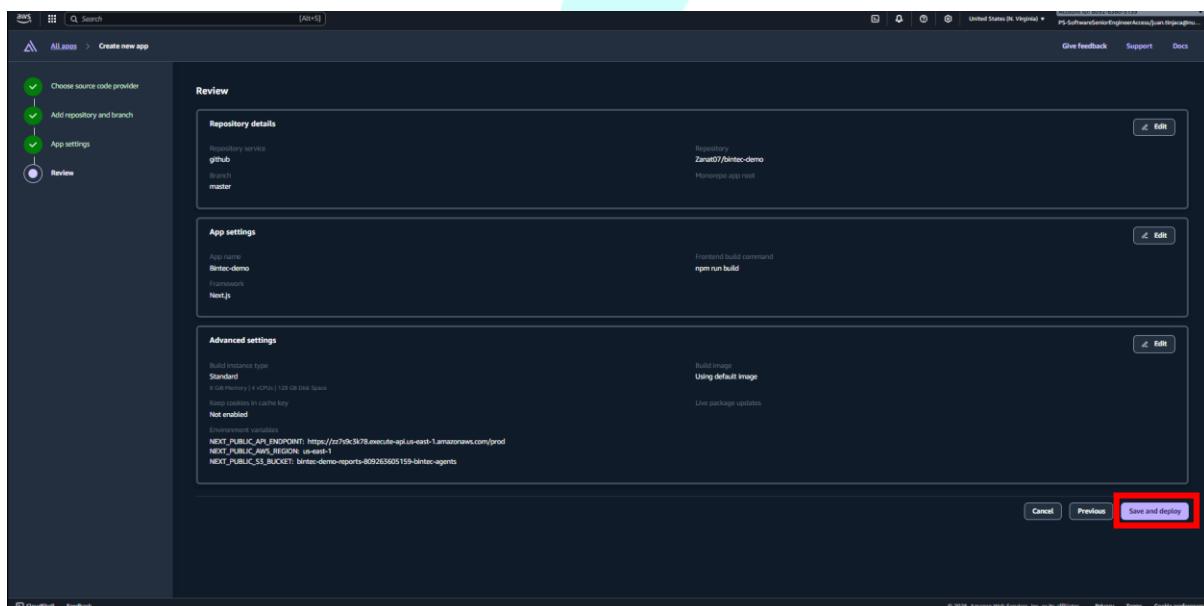
**¿Qué vamos a hacer?** Asegurar que las variables estén disponibles en producción.

### Acciones:

1. En la app Amplify → **Environment variables**
2. **Manage variables**
3. **Agregar:**
  - i. NEXT\_PUBLIC\_API\_ENDPOINT: [Tu API URL]
  - ii. NEXT\_PUBLIC\_AWS\_REGION: us-east-1
  - iii. S3\_BUCKET\_NAME=bintec-demo-reports--
  - iv.
4. **Save**



## 5. Deploy la aplicación

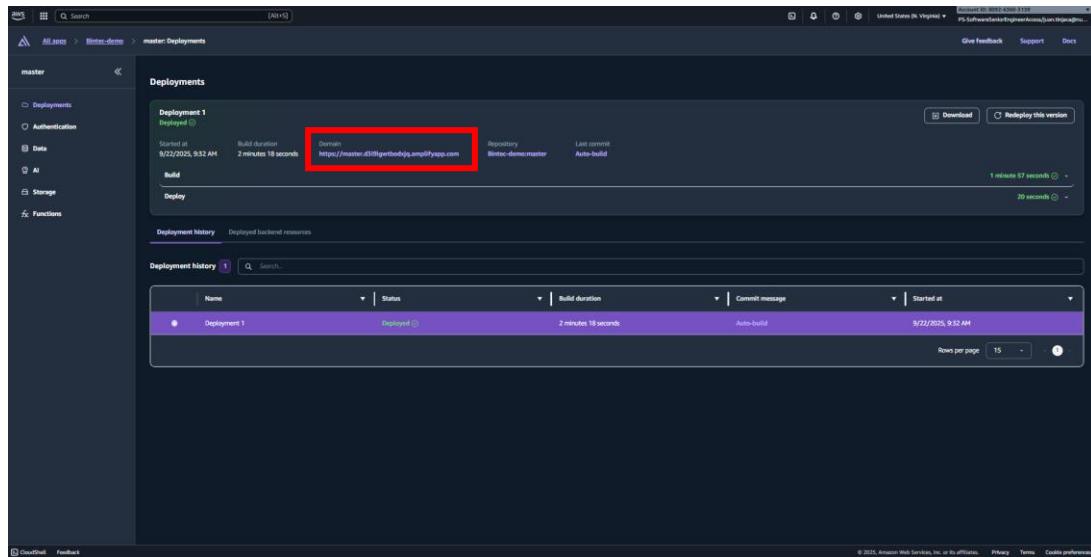


### Paso 8.3: Verificar Despliegue Final

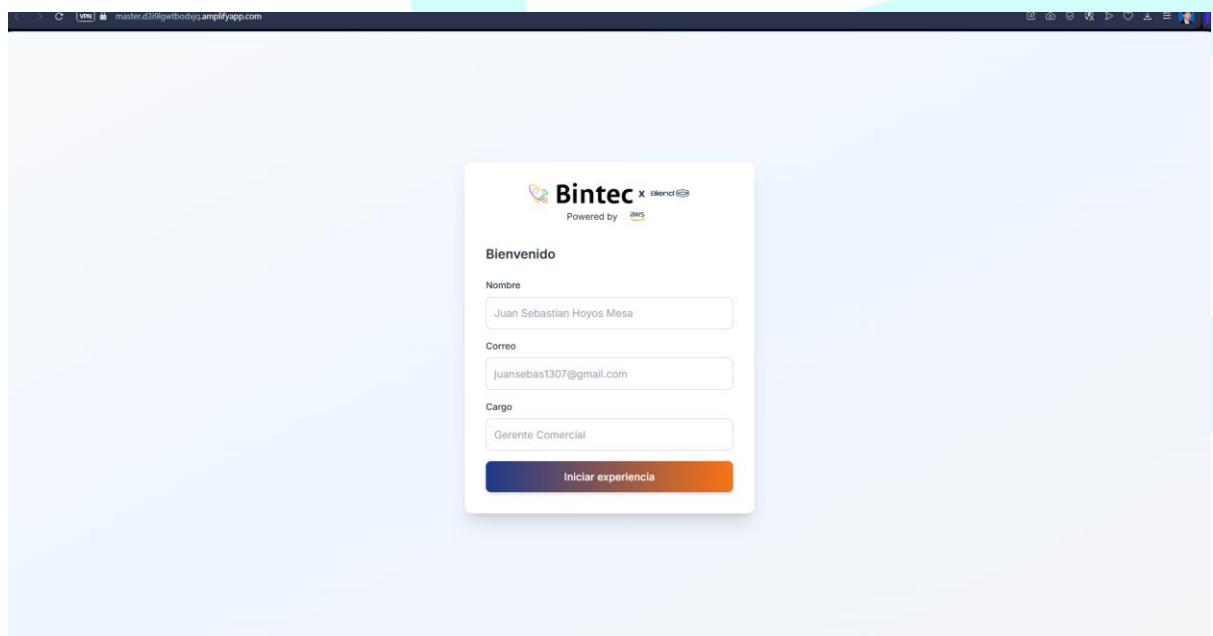
**¿Qué vamos a hacer?** Probar la aplicación en producción.

## Acciones:

1. Copiar **Domain URL** de Amplify



2. Abrir en navegador



3. Probar flujo completo end-to-end

4. Confirmar recepción de email.