

Informe del Laboratorio - Implementación de Infraestructura en AWS con CloudFormation

Este informe resume los pasos realizados para desplegar una infraestructura en AWS utilizando plantillas de CloudFormation. El objetivo fue crear y configurar diversos servicios de AWS, probar su funcionamiento e identificar y solucionar problemas encontrados durante el proceso.

Índice

1. [Configuración Inicial](#)
 2. [Despliegue de la VPC](#)
 3. [Despliegue de Instancias EC2](#)
 4. [Despliegue del Bucket S3](#)
 5. [Despliegue de DynamoDB](#)
 6. [Despliegue de SQS](#)
 7. [Despliegue de RDS](#)
 8. [Despliegue de Lambda y SNS](#)
 9. [Pruebas y Verificación](#)
 10. [Conclusiones](#)
-

Configuración Inicial

Antes de iniciar el despliegue, se realizaron las siguientes configuraciones:

1. **Configuración de Credenciales AWS:** Se configuraron las credenciales de AWS en el archivo `~/.aws/credentials`.
2. **Verificación de la Identidad del Usuario:**

```
aws sts get-caller-identity
```

Resultado:

```
{
  "UserId": "...",
  "Account": "...",
  "Arn": "arn:aws:sts::[ACCOUNT_ID]:assumed-
```

```
role/[ROLE_NAME]/[USER_NAME]"
}
```

3. **Configuración del Perfil AWS:** Se editó el archivo `~/.aws/config` para establecer la región y el formato de salida por defecto.

4. **Importación de la Clave SSH:**

- Se movió la clave privada descargada a la ubicación adecuada:

```
mv ~/Downloads/my-ec2-keypair.pem ~/.ssh/
```

- Se establecieron los permisos correctos para la clave privada:

```
chmod 400 ~/.ssh/my-ec2-keypair.pem
```

Despliegue de la VPC

Se desplegó la VPC utilizando la plantilla `vpc.yaml`:

```
aws cloudformation deploy \
  --template-file vpc.yaml \
  --stack-name MyVPCStack \
  --parameter-overrides EnvironmentName=MyEnvironment
```

Resultado:

- La VPC y sus recursos asociados se crearon exitosamente.
- Mensaje de confirmación:

```
Successfully created/updated stack - MyVPCStack
```

Despliegue de Instancias EC2

Se desplegaron las instancias EC2 utilizando la plantilla `ec2.yaml`:

```
aws cloudformation deploy \
  --template-file ec2.yaml \
  --stack-name MyEC2Stack \
```

```
--parameter-overrides \  
  KeyName=my-ec2-keypair \  
  VPCId=[VPC_ID] \  
  PublicSubnetId=[PUBLIC_SUBNET_ID] \  
  EnvironmentName=MyEnvironment
```

Notas:

- Reemplazar `[VPC_ID]` y `[PUBLIC_SUBNET_ID]` con los valores obtenidos de la salida del despliegue de la VPC.
- Se aseguró de utilizar el nombre correcto de la clave SSH (`my-ec2-keypair`).

Resultado:

- Las instancias EC2 se crearon exitosamente.
- Mensaje de confirmación:

```
Successfully created/updated stack - MyEC2Stack
```

Despliegue del Bucket S3

Se intentó desplegar el bucket S3 utilizando la plantilla `s3.yaml`:

```
aws cloudformation deploy \  
  --template-file s3.yaml \  
  --stack-name MyS3Stack \  
  --parameter-overrides EnvironmentName=MyEnvironment
```

Problema Encontrado:

- El despliegue falló debido a que el nombre del bucket contenía mayúsculas, lo cual no es permitido en nombres de buckets S3.
- **Mensaje de error:**

```
Resource handler returned message: "Bucket name should not contain  
uppercase characters"
```

Solución:

- Se modificó el parámetro `EnvironmentName` para que estuviera en minúsculas:

```
aws cloudformation delete-stack --stack-name MyS3Stack

aws cloudformation deploy \
  --template-file s3.yaml \
  --stack-name MyS3Stack \
  --parameter-overrides EnvironmentName=myenvironment
```

Resultado:

- El bucket S3 se creó exitosamente con el nombre en minúsculas.
- Mensaje de confirmación:

```
Successfully created/updated stack - MyS3Stack
```

Despliegue de DynamoDB

Se desplegó la tabla DynamoDB utilizando la plantilla `dynamodb.yaml`:

```
aws cloudformation deploy \
  --template-file dynamodb.yaml \
  --stack-name MyDynamoDBStack \
  --parameter-overrides EnvironmentName=MyEnvironment
```

Resultado:

- La tabla DynamoDB `sensor_data` se creó exitosamente.
- Mensaje de confirmación:

```
Successfully created/updated stack - MyDynamoDBStack
```

Despliegue de SQS

Se desplegó la cola SQS utilizando la plantilla `sqs.yaml`:

```
aws cloudformation deploy \
  --template-file sqs.yaml \
```

```
--stack-name MySQLStack \  
--parameter-overrides EnvironmentName=MyEnvironment
```

Resultado:

- La cola SQS `MyEnvironment-DataQueue` se creó exitosamente.
- Mensaje de confirmación:

```
Successfully created/updated stack - MySQLStack
```

Despliegue de RDS

Se intentó desplegar la base de datos RDS utilizando la plantilla `rds.yaml`:

```
aws cloudformation deploy \  
  --template-file rds.yaml \  
  --stack-name MyRDSSStack \  
  --parameter-overrides \  
    VPCId=[VPC_ID] \  
    PrivateSubnetIds=[PRIVATE_SUBNET_IDS] \  
    EC2SecurityGroup=[EC2_SECURITY_GROUP_ID] \  
    EnvironmentName=MyEnvironment \  
    DBUsername=admin \  
    DBPassword=MySecurePass123!
```

Notas:

- Reemplazar `[VPC_ID]`, `[PRIVATE_SUBNET_IDS]` y `[EC2_SECURITY_GROUP_ID]` con los valores correspondientes.
- Se utilizó una contraseña segura para el parámetro `DBPassword`.

Problemas Encontrados y Soluciones:

1. Error de Parámetro `KeyName`:

- **Mensaje de error:**

```
Parameter validation failed: parameter value for parameter name  
KeyName does not exist.
```

- **Solución:** Se eliminó la referencia al parámetro `KeyName` en la plantilla `rds.yaml`, ya que no era necesario para RDS.

2. Error de Compatibilidad de Versiones:

- **Mensaje de error:**

```
RDS does not support creating a DB instance with the following combination: DBInstanceClass=db.t2.micro, Engine=mysql, EngineVersion=8.0.35
```

- **Solución:** Se actualizó el `DBInstanceClass` a `db.t3.micro` y se ajustó la `EngineVersion` a una versión compatible.

3. Error de Versión No Encontrada:

- **Mensaje de error:**

```
Cannot find version 8.0.28 for mysql
```

- **Solución:** Se cambió la `EngineVersion` a una versión disponible, como `8.0.30`.

Resultado Final:

- Después de ajustar los parámetros y corregir los errores, la base de datos RDS se creó exitosamente.
- Mensaje de confirmación:

```
Successfully created/updated stack – MyRDSSStack
```

Despliegue de Lambda y SNS

Se desplegó la función Lambda y el tema SNS utilizando la plantilla `lambda_sns.yaml`:

```
aws cloudformation deploy \
  --template-file lambda_sns.yaml \
  --stack-name MyLambdaSNSStack \
  --capabilities CAPABILITY_NAMED_IAM \
  --parameter-overrides \
    EnvironmentName=MyEnvironment \
    SNSNotificationEmail=tu_email@example.com \
    DataQueueArn=arn:aws:sqs:[REGION]:[ACCOUNT_ID]:MyEnvironment-DataQueueue \
    LambdaRoleArn=arn:aws:iam::[ACCOUNT_ID]:role/LabRole
```

Notas:

- Reemplazar `tu_email@example.com` con la dirección de correo electrónico para recibir notificaciones.
- Reemplazar `[REGION]` y `[ACCOUNT_ID]` con los valores correspondientes.
- Se utilizó el rol existente `LabRole` en lugar de crear uno nuevo.

Problemas Encontrados y Soluciones:

1. Error de Dependencias No Resueltas:

- **Mensaje de error:**

```
Template error: instance of Fn::GetAtt references undefined
resource DataQueue
```

- **Solución:** Se aseguró que la cola SQS `MyEnvironment-DataQueue` existiera y se proporcionó su ARN correcto en los parámetros.

2. Error de Capacidades Insuficientes:

- **Mensaje de error:**

```
Requires capabilities : [CAPABILITY_NAMED_IAM]
```

- **Solución:** Se agregó el parámetro `--capabilities CAPABILITY_NAMED_IAM` al comando de despliegue.

3. Error de Estado del Stack:

- **Mensaje de error:**

```
Stack is in ROLLBACK_COMPLETE state and cannot be updated.
```

- **Solución:** Se eliminó el stack existente y se intentó desplegar nuevamente.

Resultado:

- La función Lambda y el tema SNS se crearon exitosamente.
- Mensaje de confirmación:

```
Successfully created/updated stack - MyLambdaSNSStack
```

Pruebas y Verificación

Se realizaron pruebas para verificar el funcionamiento de la infraestructura desplegada.

Envío de Mensajes a la Cola SQS

Se enviaron mensajes a la cola SQS utilizando la AWS CLI:

```
aws sqs get-queue-url --queue-name MyEnvironment-DataQueue
```

Resultado:

```
{
  "QueueUrl": "https://sqs.
[REGION].amazonaws.com/[ACCOUNT_ID]/MyEnvironment-DataQueue"
}
```

Se enviaron mensajes de prueba:

```
aws sqs send-message \
  --queue-url "https://sqs.
[REGION].amazonaws.com/[ACCOUNT_ID]/MyEnvironment-DataQueue" \
  --message-body '{"sensor_id": "sensor_1", "timestamp": "2024-10-
12T16:00:00Z", "value": 150}'
```

Verificación en DynamoDB

- Se accedió a la consola de DynamoDB y se verificó que los datos se almacenaron correctamente en la tabla `sensor_data`.

Verificación de Notificaciones SNS

- Se confirmó la recepción de correos electrónicos con alertas desde SNS, indicando que la función Lambda procesó los mensajes y envió notificaciones cuando el valor del sensor superaba el umbral definido.

Revisión de Logs en CloudWatch

- Se revisaron los logs de la función Lambda en CloudWatch para asegurar que la función se ejecutó correctamente y para identificar posibles errores.

Conclusiones

- Se logró desplegar y configurar una infraestructura en AWS utilizando CloudFormation, incluyendo VPC, EC2, S3, DynamoDB, SQS, RDS, Lambda y SNS.
 - Se enfrentaron y resolvieron varios problemas relacionados con nombres de recursos, versiones de motores de base de datos y permisos IAM.
 - Las pruebas realizadas confirmaron que los componentes funcionan de manera integrada, permitiendo el procesamiento de datos de sensores y el envío de notificaciones.
 - La experiencia adquirida en la resolución de problemas y ajuste de parámetros es valiosa para futuros proyectos en AWS.
-

Recomendaciones

- **Gestión de Nombres de Recursos:** Evitar el uso de mayúsculas en los nombres de recursos que tienen restricciones específicas, como los buckets de S3.
 - **Compatibilidad de Versiones:** Consultar la documentación oficial de AWS para conocer las versiones compatibles de servicios como RDS.
 - **Permisos IAM:** Planificar y definir correctamente las políticas de IAM para evitar problemas de permisos durante el despliegue.
 - **Uso de Parámetros y Outputs:** Aprovechar los parámetros y outputs de CloudFormation para pasar información entre stacks y facilitar la configuración.
-

Nota: Se han omitido detalles sensibles, como identificadores de cuenta y direcciones de correo electrónico, para mantener la confidencialidad de la información.