Morris & Opazo

aws partner network

Advanced
Consulting
Partner

- Data & Analytics Competency
- Solution Provider
- Public Sector Partner
- Immersion Day Partner
- AWS Lambda

Reducción de costos de Machine Learning con Endpoint Serverless

We Love the Cloud

morrisopazo.com

contacto@morrisopazo.com

Resumen

Morris & Opazo

Al momento de incorporar un modelo de aprendizaje automático dentro de tu organización, un concepto a definir es el tipo de consumo de esta solución. De acuerdo al contexto del negocio, una alternativa es desplegar una API (EndPoint) que de respuesta en tiempo real, lo que requiere habilitar una instancia 24/7 en el ambiente de producción, que por defecto, se traduce en un cobro por todo el tiempo que esté habilitado.

A continuación exploramos la solución desarrollada por AWS a ésta problemática, gracias a el despliegue de un End-Point Serverless. Con este tipo de puntos de enlace (EndPoint), podemos reducir los costos relacionados a las inferencias, donde el aprovisionamiento se hará en el momento que esté sea llamado, es decir, bajo demanda, y totalmente administrado por AWS.

DIAGRAMA DE ARQUITECTURA

Morris & Opazo

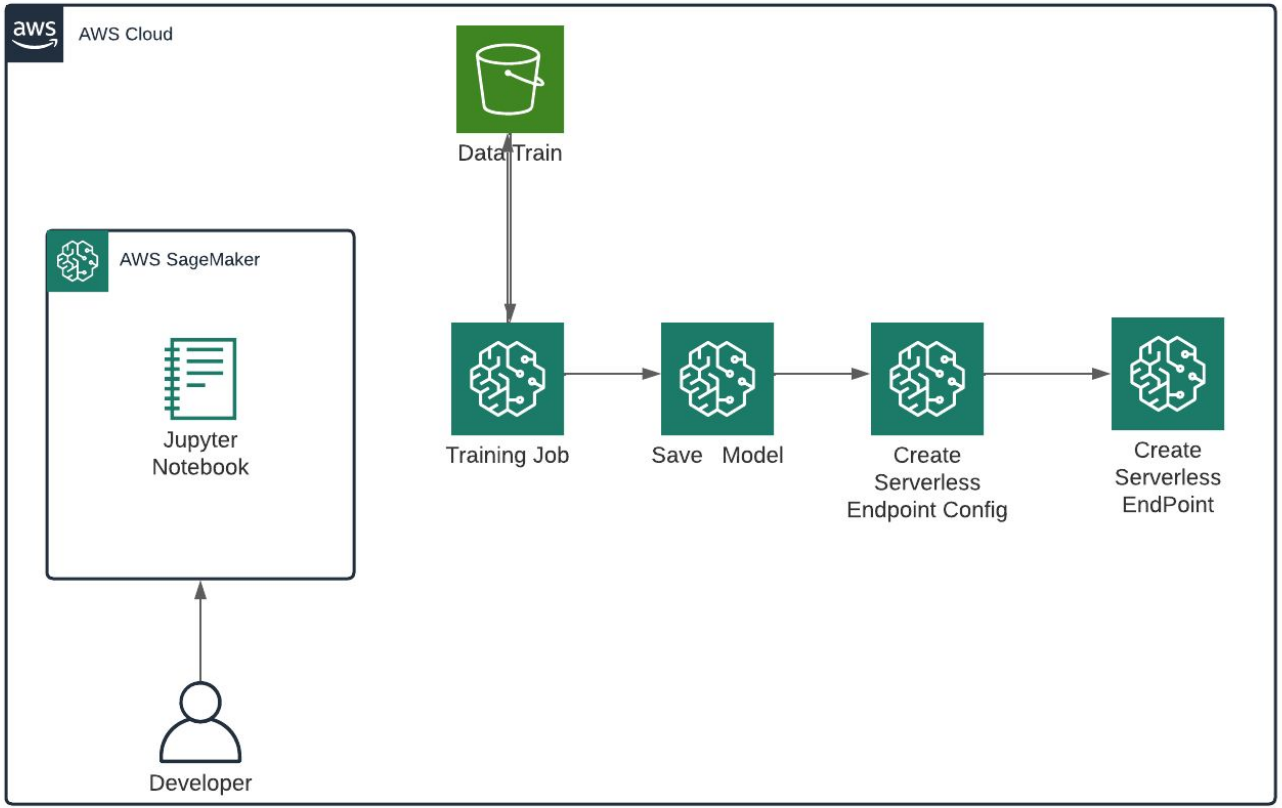


DIAGRAMA DE ARQUITECTURA

Morris & Opazo

Servicios de AWS se utilizan para qué fines en esta arquitectura:

- **Sagemaker Studio**: Desde aquí se escribirá el código relacionado a todo el despliegue de la solución mostrada, todo será escrito desde una Jupyter Notebook

- **SageMaker(TrainingJob)**: El encargado de levantar la instancia de entrenamiento, el resultado del entrenamiento nos entregará nuestro artefacto.

DIAGRAMA DE ARQUITECTURA

Morris & Opazo

aws

- **SageMaker(ModelRegistry):** El encargado de registrar nuestro modelo dentro de la nube, para posteriormente utilizarlo en levantamiento de inferencias.

- **SageMaker(Serverless Endpoint):** Aquí levantaremos nuestro Endpoint que no necesitará aprovisionamiento de servidor

- **Simple Storage Service:** Desde aquí obtendremos nuestra data de entrenamiento.

IMPLEMENTACIÓN

Morris & Opazo

1- **Preparación:** Nuestro primer paso será preparar el ambiente a utilizar, en éste caso usaremos las Jupyter Notebook proporcionadas por SageMaker como nuestro entorno de desarrollo, por lo cual lo primero será importar librerías a usar, además del rol de ejecución para realizar correctamente el levantamiento de tareas.

```
import sagemaker
```

```
from sagemaker.xgboost.estimator import XGBoost
```

```
import pandas as pd
```

```
import numpy as np
```

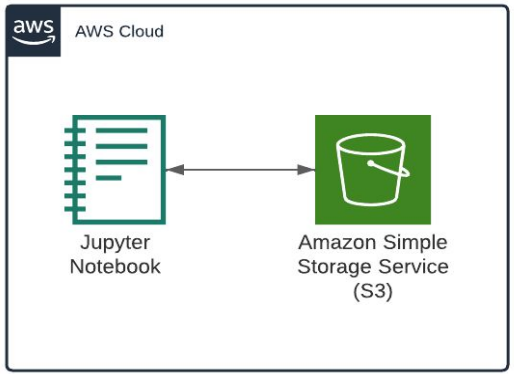
```
import boto3
#from sagemaker import get_execution_role
iam = boto3.client('iam')
role =
```

```
import json
import pathlib
import pickle
import tarfile
import joblib
import numpy as np
import pandas as pd
import xgboost
import argparse
import os
```

IMPLEMENTACIÓN

Morris & Opazo

2- **Data:** Para empezar a prepararnos para nuestro entrenamiento, necesitaremos un set de datos que esté alojado en S3 para la ejecución de este. Puedes fácilmente leer un set de datos en tu Jupyter Notebook, prepararlo y luego dejarlo en un bucket, incluso leer desde un bucket para luego editar y subir a un destino diferente. Siempre teniendo en cuenta que el resultado final debe ser un bucket.



IMPLEMENTACIÓN

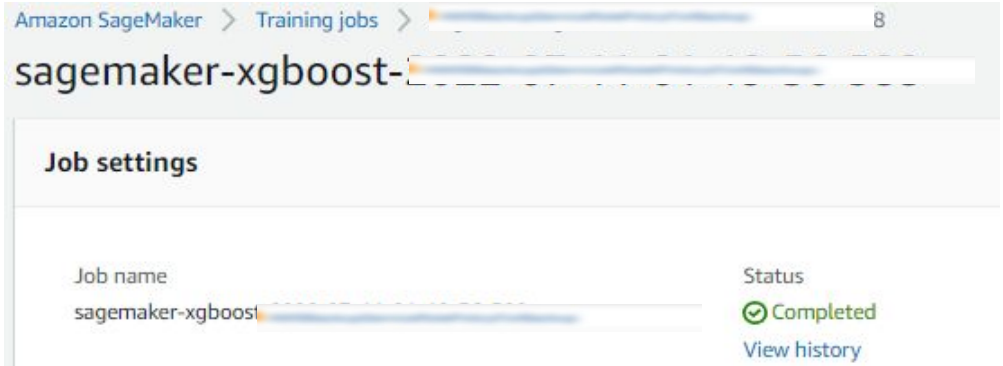
Morris & Opazo

3- **Entrenamiento:** Una vez que nuestros pasos anteriores fueron realizados, pasaremos a realizar nuestro entrenamiento con los datos anteriormente almacenados en S3. El resultado de éste entrenamiento será nuestro artefacto el cual deberemos trabajar en el paso siguiente.

```
train_input = TrainingInput("
validation_input = TrainingInp

estimator.fit({'train': train_input, 'validation': validation_input})
```

```
2022-07-11 01:19:56 Starting - Starting the training job...ProfilerReport-1657502396: InProgress
...
```



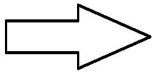
IMPLEMENTACIÓN

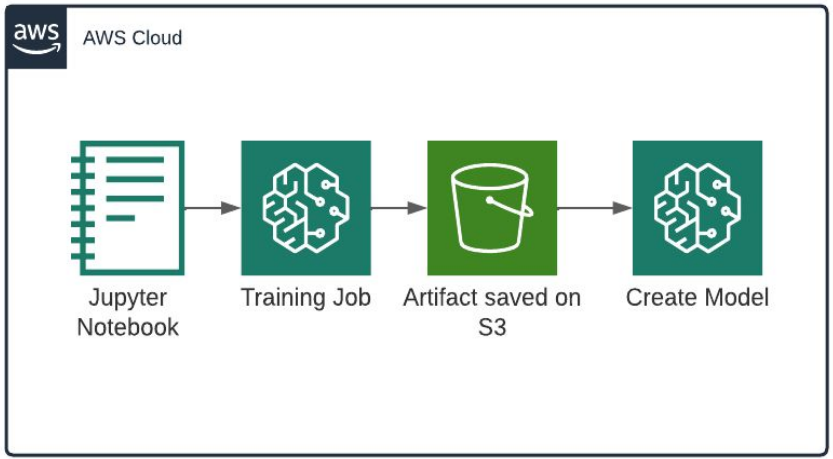
Morris & Opazo

4- **Crear modelo:** El resultado de nuestro entrenamiento nos arroja un artefacto que es almacenado en S3, para poder utilizarlo en los pasos posteriores; deberemos registrar éste artefacto en el directorio de modelos de AWS.

```
sagemaker_session = sagemaker.Session()
```

```
model = sagemaker_session.create_model_from_job(
        name="
        training_job_name=training_job_1_info["Tr
        role=role,
        image_uri=training_job_1_info[
    )
```



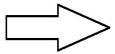


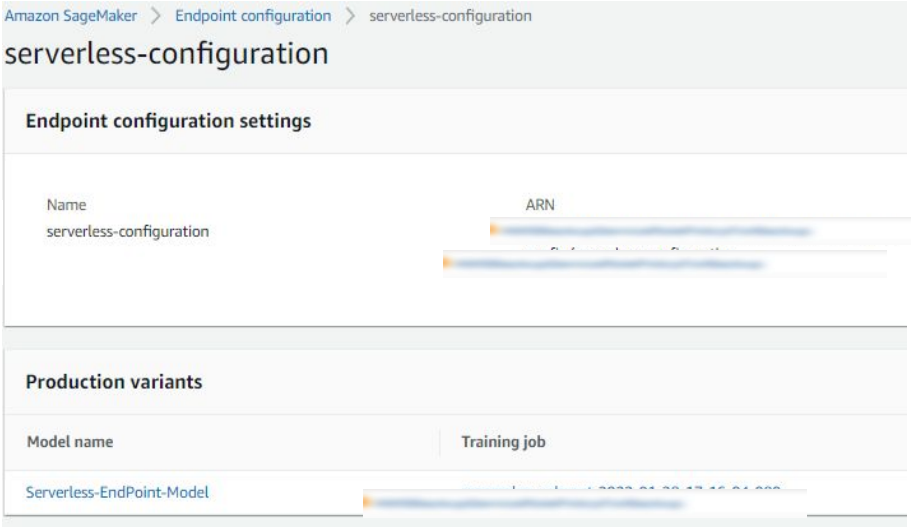
IMPLEMENTACIÓN

Morris & Opazo

5- **Crear configuración endpoint:** Una vez que nuestro modelo fue registrado, podemos pasar a crear nuestra configuración de EndPoint. Aquí le podemos definir si consumirá alguna instancia en específico o será Serverless. En éste caso estamos lanzando una configuración Serverless.

```
response = sagemaker_boto_client.create_endpoint_config(
    EndpointConfigName="serverless-configuration",
    ProductionVariants=[
        {
            "ModelName": [redacted]
            "VariantName": [redacted]
            "ServerlessConfig": {
                "MemorySizeInMB": 2048,
                "MaxConcurrency": 20
            }
        }
    ]
)
```

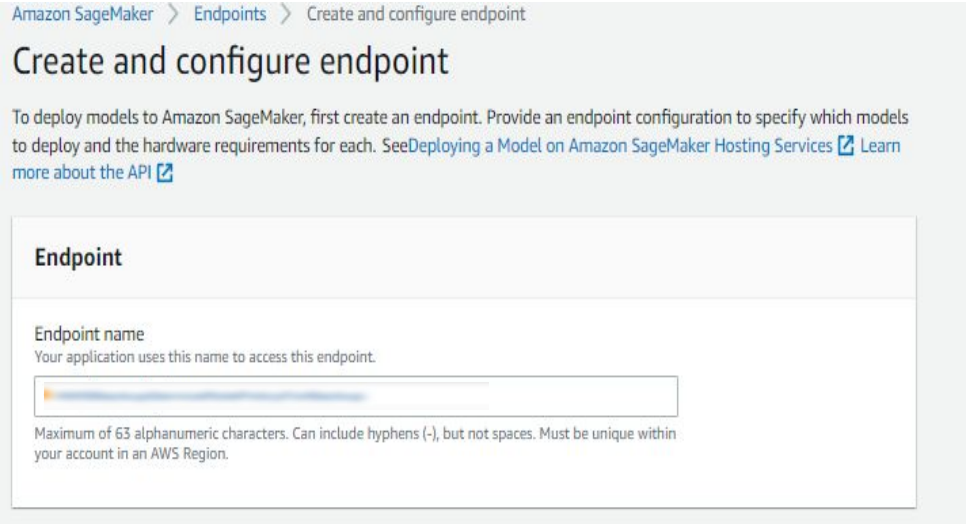


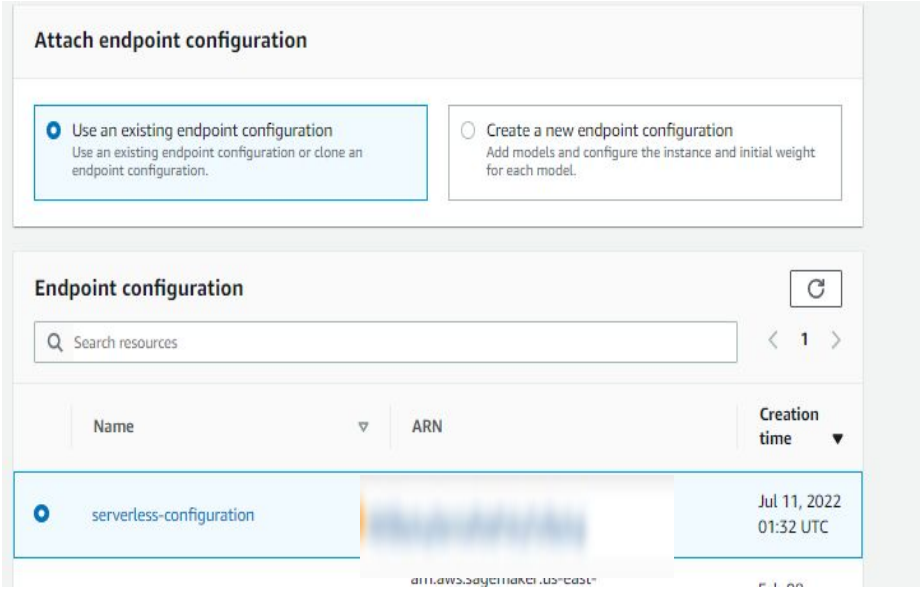


IMPLEMENTACIÓN

Morris & Opazo

6- **Levantamiento de endpoint:** Una vez que la configuración ya está arriba, pasaremos a levantar el enlace. Esto se puede realizar fácilmente tanto desde la consola, como desde nuestra Jupyter Notebook.

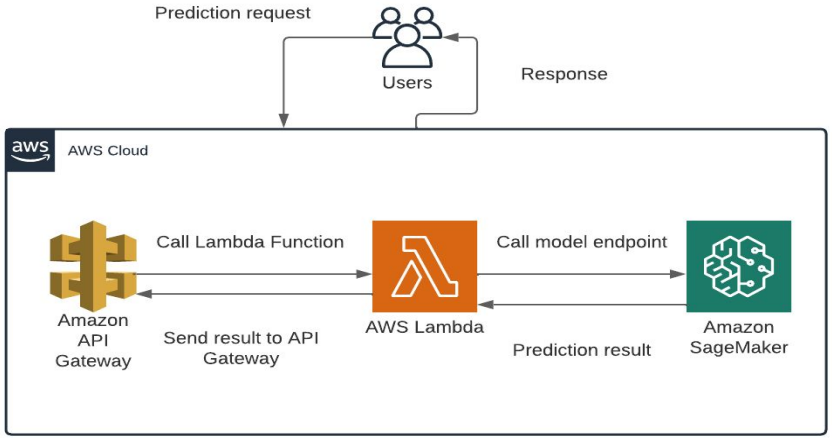




IMPLEMENTACIÓN

Morris & Opazo

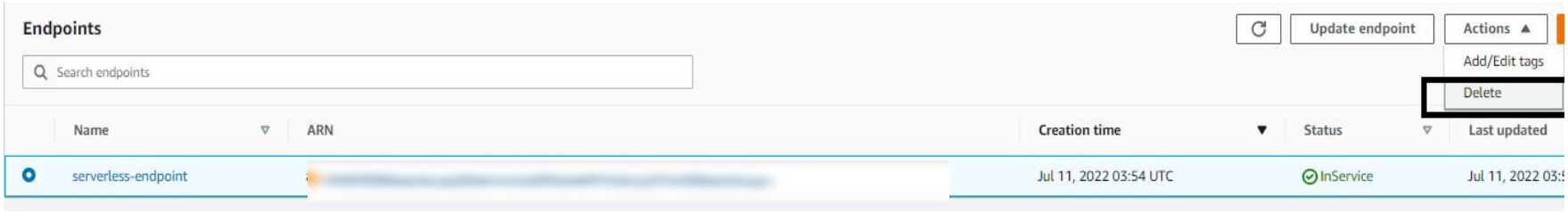
7- **Invocar endpoint:** Luego de haber levantado el endpoint llega el momento de consumir éste mismo. Puedes consumir directamente desde SageMaker o si ya confías totalmente en tu solución, puedes pasarla a producción. A continuación mostraremos un caso de uso a la hora de consumir EndPoints.



IMPLEMENTACIÓN

Morris & Opazo

6- **Borrar endpoint:** Recuerda que un endpoint es un recurso que está siendo levantado y queda prendido a menos que no le indiques su finalización. Evita siempre incurrir en gastos no esperados y recuerda eliminar tu endpoint. A continuación se mostrará dos maneras.




```
response = client.delete_endpoint(
    EndpointName="
)
```

Conclusion

Morris & Opazo

En este ejercicio, se logró levantar un Endpoint serverless, demostrando el paso a paso, desde preparar el ambiente hasta consumir dicho endpoint. Se ocupó una máquina para la notebook y el entrenamiento, pero a diferencia de un Endpoint en real-time, ésta vez nuestra inferencia no le especificamos máquina, haciendo de éste método una alternativa muy amigable a la hora de reducir costos en el ámbito de Machine Learning.

Gracias a la flexibilidad entregada por AWS, no se debe hacer mayores cambios a la arquitectura si actualmente estás usando EndPoint real-time, únicamente deberás apegarte a tu regla de negocio. Puesto que si necesitas una capacidad totalmente inmediata y de gran potencia, real-time deberá ser la opción. Si tus predicciones pueden esperar un tiempo de “en frío” y quieres reducir gastos de operación, ServerLess Endpoint es la alternativa.

LINKS RELACIONADOS

Morris & Opazo

- https://docs.aws.amazon.com/sagemaker/latest/dg/serverless-endpoints.html
- https://docs.aws.amazon.com/sagemaker/latest/dg/serverless-endpoints-create-invoke-update-delete.html
- https://docs.aws.amazon.com/sagemaker/latest/dg/serverless-endpoints-monitoring.html
- https://docs.aws.amazon.com/sagemaker/latest/dg/serverless-endpoints-troubleshooting.html
- https://docs.aws.amazon.com/sagemaker/latest/dg/deploy-model.html

Morris & Opazo

aws partner network

Advanced
Consulting Partner

- Data & Analytics Competency
- Solution Provider
- Public Sector Partner
- Immersion Day Partner
- AWS Lambda

Expertos en Tecnología de la Nube

We Love the Cloud

Morrisopazo.com

contacto@Morrisopazo.com