



Amazon FSx for NetApp ONTAP (FSx ONTAP) para MLOps

NetApp Solutions

NetApp
December 19, 2024

Tabla de contenidos

- Amazon FSx for NetApp ONTAP (FSx ONTAP) para MLOps 1
 - Amazon FSx for NetApp ONTAP (FSx ONTAP) para MLOps 1
 - 1 parte: Integración de Amazon FSx para NetApp ONTAP (FSx ONTAP) como bloque de S3 privado en AWS SageMaker 1
 - Parte 2: Aprovechamiento de AWS Amazon FSx para NetApp ONTAP (FSx ONTAP) como fuente de datos para el entrenamiento de modelos en SageMaker 15
 - Parte 3: Creación de Una canalización simplificada de MLOps (CI/CT/CD) 24
 - 2. Scheduling a job to shutdown the notebook to save the cost 27

Amazon FSx for NetApp ONTAP (FSx ONTAP) para MLOps

Amazon FSx for NetApp ONTAP (FSx ONTAP) para MLOps

En esta sección se profundiza en la aplicación práctica del desarrollo de la infraestructura de IA, ofreciendo un tutorial integral sobre la construcción de una canalización de MLOps con FSx ONTAP. Compuesto por tres ejemplos completos, te guía para satisfacer tus necesidades de MLOps a través de esta potente plataforma de gestión de datos.

Autor(es):

Jian Jian (KEN), científico sénior de datos y aplicado, NetApp

Estos artículos se centran en:

1. ["1 parte: Integración de Amazon FSx para NetApp ONTAP \(FSx ONTAP\) como bloque de S3 privado en AWS SageMaker"](#)
2. ["Parte 2: Aprovechamiento de Amazon FSx para NetApp ONTAP \(FSx ONTAP\) como fuente de datos para el entrenamiento de modelos en SageMaker"](#)
3. ["Parte 3: Creación de Una canalización simplificada de MLOps \(CI/CT/CD\)"](#)

Al final de esta sección, habrás obtenido una sólida comprensión de cómo usar FSx ONTAP para optimizar los procesos de MLOps.

1 parte: Integración de Amazon FSx para NetApp ONTAP (FSx ONTAP) como bloque de S3 privado en AWS SageMaker

Esta sección proporciona una guía sobre la configuración de FSx ONTAP como un bucket S3 privado usando AWS SageMaker.

Autor(es):

Jian Jian (KEN), científico sénior de datos y aplicado, NetApp

Introducción

Usando SageMaker como ejemplo, esta página proporciona una guía para configurar FSx ONTAP como un bucket privado de S3.

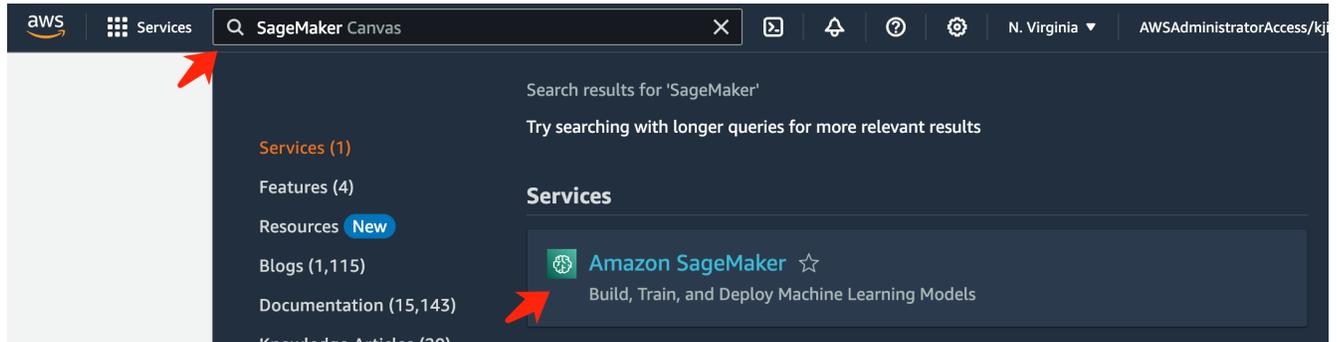
Si quiere más información sobre FSx ONTAP, eche un vistazo a esta presentación (["Enlace de vídeo"](#))

Guía del usuario

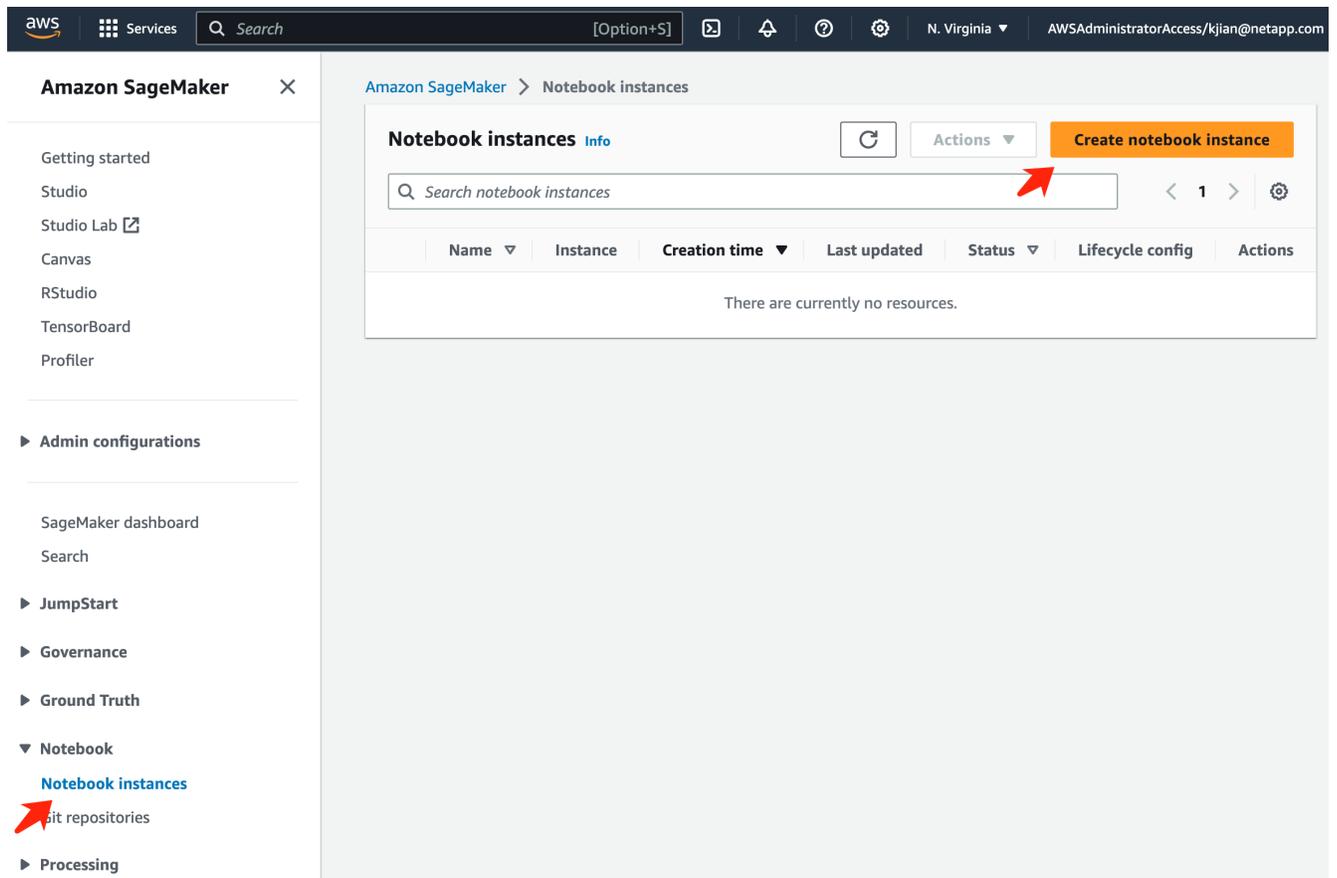
Creación de servidores

Cree una instancia de SageMaker Notebook

1. Abra la consola de AWS. En el panel de búsqueda, busque en SageMaker y haga clic en el servicio **Amazon SageMaker**.



2. Abra las instancias del bloc de notas * en la pestaña del bloc de notas, haga clic en el botón naranja * Crear instancia del bloc de notas *.



3. En la página de creación, ingrese el **Nombre de la instancia del bloc de notas** Expande el panel **Red** Deje otras entradas predeterminadas y seleccione un **VPC**, **Subred** y **Grupo(s) de seguridad**. (Este **vPC** y **subred** se utilizarán para crear el sistema de archivos FSX ONTAP más adelante) Haga clic en el botón naranja **Crear instancia de bloc de notas** en la parte inferior derecha.

Amazon SageMaker > Notebook instances > Create notebook instance

Create notebook instance

Amazon SageMaker provides pre-built fully managed notebook instances that run Jupyter notebooks. The notebook instances include example code for common model training and hosting exercises. [Learn more](#)

Notebook instance settings

Notebook instance name
fsxn-demo

Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.

Notebook instance type
ml.t3.medium

Elastic Inference [Learn more](#)
none

Platform identifier [Learn more](#)
Amazon Linux 2, Jupyter Lab 3

▶ Additional configuration

Permissions and encryption

IAM role
Notebook instances require permissions to call other services including SageMaker and S3. Choose a role or let us create a role with the [AmazonSageMakerFullAccess](#) IAM policy attached.
AmazonSageMakerServiceCatalogProductsUseRole

Create role using the role creation wizard

Root access - optional
 Enable - Give users root access to the notebook
 Disable - Don't give users root access to the notebook
Lifecycle configurations always have root access

Encryption key - optional
Encrypt your notebook data. Choose an existing KMS key or enter a key's ARN.
 No Custom Encryption

Network - optional

VPC - optional
Default vpc-0df3956ab1fca2ec9 (172.31.0.0/16)

Subnet
Choose a subnet in an availability zone supported by Amazon SageMaker.
 subnet-00060df0d0f562672 (172.31.16.0/20) | us-east-1a

Security group(s)
sg-0a39b3985770e9256 (default) X

Direct internet access
 Enable — Access the internet directly through Amazon SageMaker
 Disable — Access the internet through a VPC
To train or host models from a notebook, you need internet access. To enable internet access, make sure that your VPC has a NAT gateway and your security group allows outbound connections. [Learn more](#)

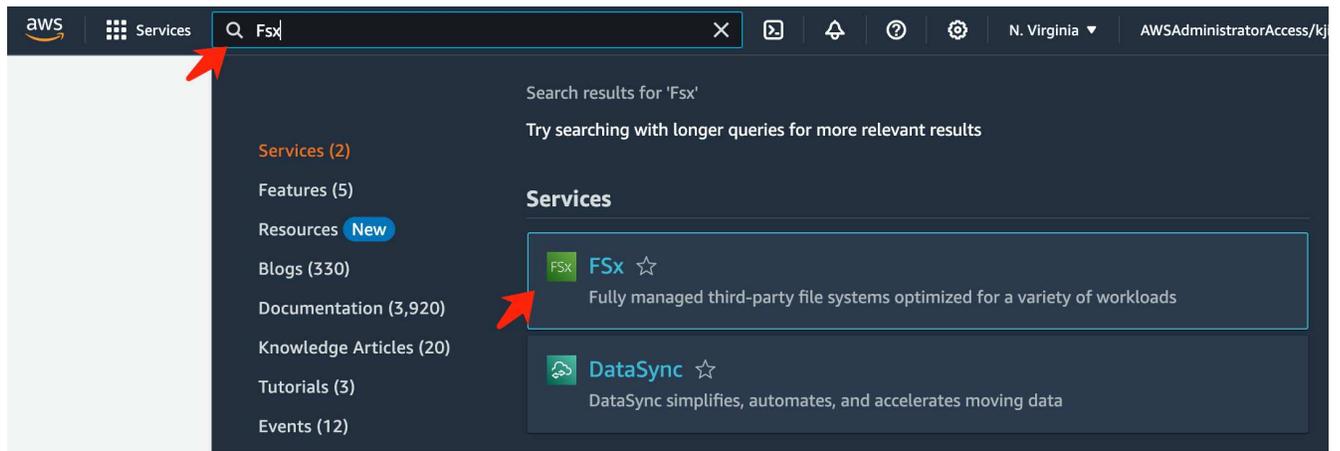
▶ Git repositories - optional

▶ Tags - optional

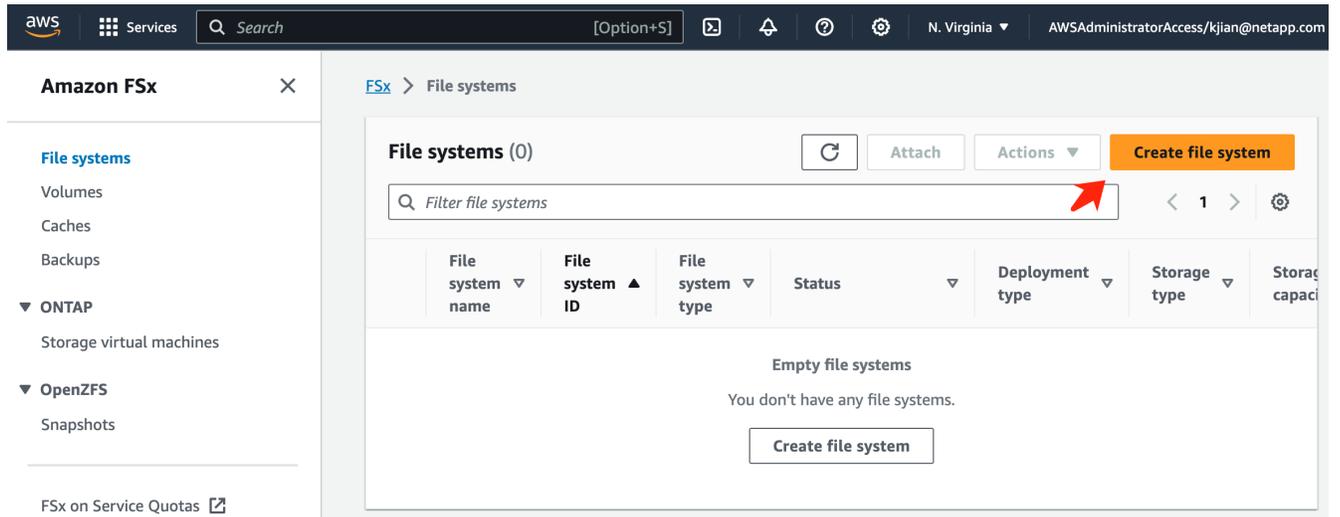
Cancel Create notebook instance

Crear un sistema de archivos FSX ONTAP

1. Abra la consola de AWS. En el panel de búsqueda, busca FSX y haz clic en el servicio **FSX**.



2. Haga clic en **Crear sistema de archivos**.



3. Seleccione la primera tarjeta **FSX ONTAP** y haga clic en **Siguiente**.

aws Services Search [Option+S] N. Virginia AWSAdministratorAccess/kjian@netapp

FSx > File systems > Create file system

Step 1
Select file system type

Step 2
Specify file system details

Step 3
Review and create

Select file system type

File system options

- Amazon FSx for NetApp ONTAP
- Amazon FSx for OpenZFS
- Amazon FSx for Windows File Server
- Amazon FSx for Lustre

Amazon FSx for NetApp ONTAP

Amazon FSx for NetApp ONTAP provides feature-rich, high-performance, and highly-reliable storage built on NetApp's popular ONTAP file system and fully managed by AWS.

- Broadly accessible from Linux, Windows, and macOS compute instances and containers (running on AWS or on-premises) via industry-standard NFS, SMB, and iSCSI protocols.
- Provides ONTAP's popular data management capabilities like Snapshots, SnapMirror (for data replication), FlexClone (for data cloning), and data compression / deduplication.
- Delivers hundreds of thousands of IOPS with consistent sub-millisecond latencies, and up to 3 GB/s of throughput.
- Offers highly-available and highly-durable single-AZ and multi-AZ deployment options, SSD storage with support for cross-region replication, and built-in, fully managed backups.
- Supports dynamic scaling of your file system to fit your storage capacity and throughput needs.
- Automatically tiers infrequently-accessed data to capacity pool storage, a fully elastic storage tier that can scale to petabytes in size and is cost-optimized for infrequently-accessed data.
- Integrates with Microsoft Active Directory (AD) to support Windows-based environments and enterprises.

Cancel Next

4. En la página de configuración de detalles.
 - a. Seleccione la opción **Standard create**.

aws Services Search [Option+S] N. Virginia AWSAdministratorAccess/kjian@netapp

FSx > File systems > Create file system

Step 1
[Select file system type](#)

Step 2
Specify file system details

Step 3
Review and create

Specify file system details

Creation method

- Quick create
Use recommended best-practice configurations. Most configuration options can be changed after the file system is created.
- Standard create
You set all of the configuration options, including specifying performance, networking, security, backups, and maintenance.

- b. Introduzca el **Nombre del sistema de archivos** y la **Capacidad de almacenamiento SSD**.

File system details

File system name - optional [Info](#)

fsxn-demo

Maximum of 256 Unicode letters, whitespace, and numbers, plus + - = . _ : /

Deployment type [Info](#)

- Multi-AZ
- Single-AZ

SSD storage capacity [Info](#)

1024

GiB

Minimum 1024 GiB; Maximum 192 TiB.

Provisioned SSD IOPS

Amazon FSx provides 3 IOPS per GiB of storage capacity. You can also provision additional SSD IOPS as needed.

- Automatic (3 IOPS per GiB of SSD storage)
- User-provisioned

Throughput capacity [Info](#)

The sustained speed at which the file server hosting your file system can serve data. The file server can also burst to higher speeds for periods of time.

- Recommended throughput capacity
128 MB/s
- Specify throughput capacity

c. Asegúrese de usar **VPC** y **subnet** igual a la instancia **SageMaker Notebook**.

Network & security

Virtual Private Cloud (VPC) | [Info](#)
Specify the VPC from which your file system is accessible.

vpc-0df3956ab1fca2ec9 (CIDR: 172.31.0.0/16) ▼

VPC Security Groups | [Info](#)
Specify VPC Security Groups to associate with your file system's network interfaces.

Choose VPC security group(s) ▼

sg-0a39b3985770e9256 (default) ✕

Preferred subnet | [Info](#)
Specify the preferred subnet for your file system.

subnet-00060df0d0f562672 (us-east-1a | use1-az4) ▼

Standby subnet

subnet-02b029f24d03a4af2 (us-east-1b | use1-az6) ▼

VPC route tables | [Info](#)
Specify the VPC route tables to associate with your file system.

VPC's main route table

Select one or more VPC route tables

Endpoint IP address range | [Info](#)
Specify the IP address range in which the endpoints to access your file system will be created

Unallocated IP address range from your VPC
Simplest option for access from other AWS services or peered / on-premises networks

Floating IP address range outside your VPC

Enter an IP address range

d. Introduzca el nombre de la máquina virtual **Storage** y **especifique una contraseña** para su SVM (máquina virtual de almacenamiento).

Default storage virtual machine configuration

Storage virtual machine name [Info](#)

fsxn-svm-demo

SVM administrative password
Password for this SVM's "vsadmin" user, which you can use to access the ONTAP CLI or REST API. You can provide a password later if you don't provide one now.

Don't specify a password

Specify a password

Password

.....

Confirm password

.....

Volume security style
The security style of the volume determines whether preference is given to NTFS or UNIX ACLs for multi-protocol access. The MIXED mode is not required for multi-protocol access and is only recommended for advanced users.

Unix (Linux) ▼

Active Directory
Joining an Active Directory enables access from Windows and MacOS clients over the SMB protocol.

Do not join an Active Directory

Join an Active Directory

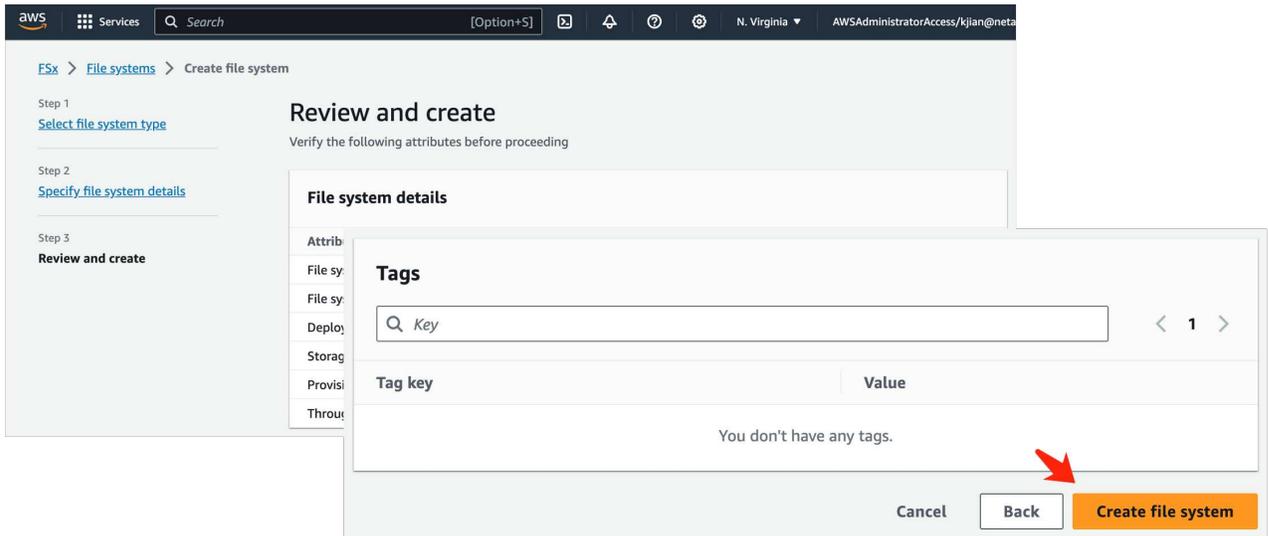
e. Deja otras entradas predeterminadas y haz clic en el botón naranja **Siguiente** en la parte inferior derecha.

► **Backup and maintenance - optional**

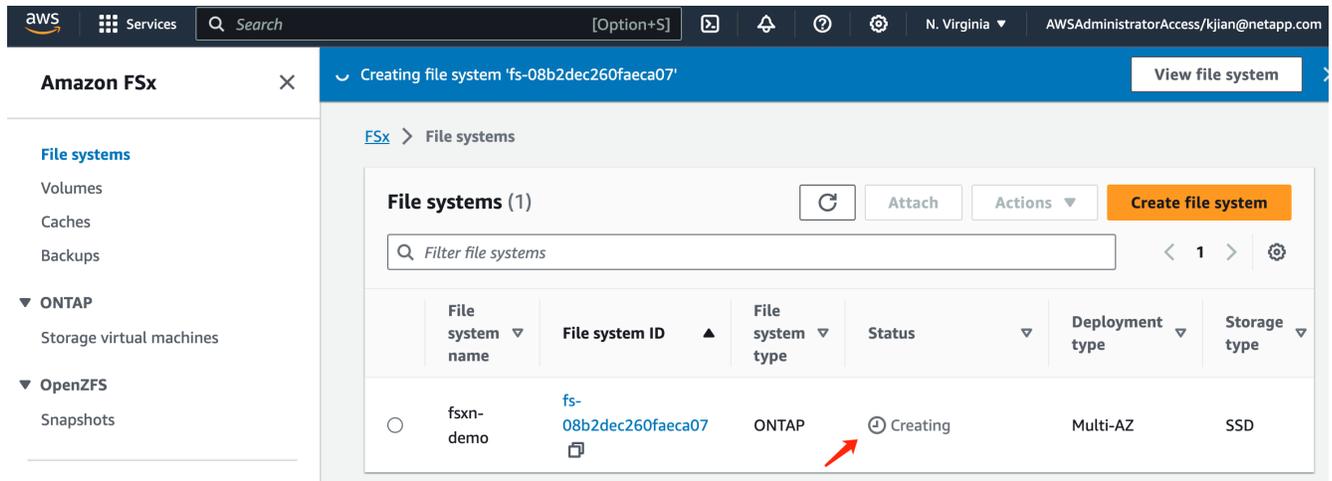
► **Tags - optional**

Cancel **Back** **Next**

f. Haga clic en el botón naranja **Crear sistema de archivos** en la parte inferior derecha de la página de revisión.



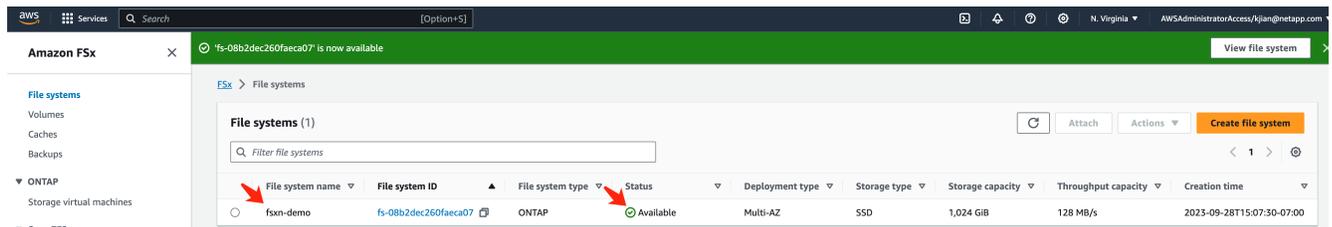
5. Puede tardar unos **20-40 minutos** en activar el sistema de archivos FSX.



Configuración del servidor

Configuración de ONTAP

1. Abra el sistema de archivos FSX creado. Por favor, asegúrese de que el estado es **disponible**.



2. Seleccione la pestaña **Administración** y mantenga el **Punto final de administración - dirección IP** y el **Nombre de usuario del administrador de ONTAP**.

The screenshot shows the AWS Management Console for an Amazon FSx ONTAP file system named 'fsxn-demo (fs-08b2dec260faeca07)'. The 'Administration' tab is active, showing the following details:

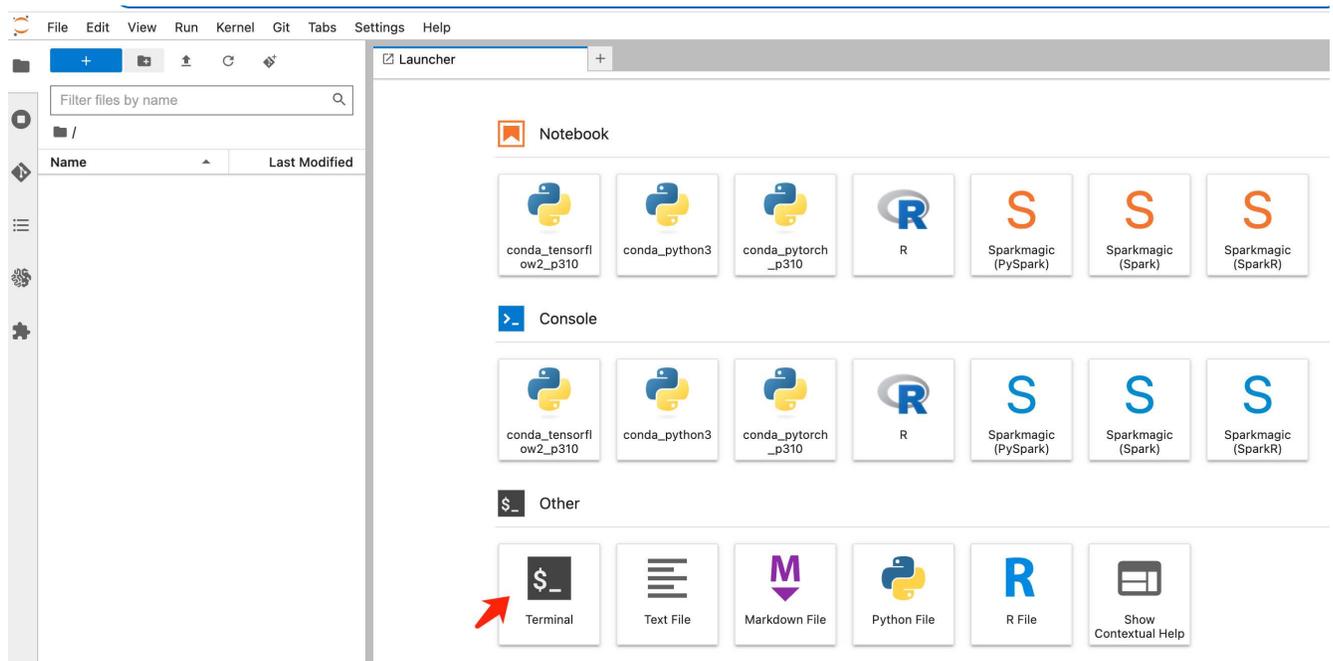
- Summary:**
 - File system ID: fs-08b2dec260faeca07
 - SSD storage capacity: 1024 GiB
 - Throughput capacity: 128 MB/s
 - Provisioned IOPS: 3072
 - Availability Zones: us-east-1a (Preferred), us-east-1b (Standby)
 - Creation time: 2023-09-28T14:50:07:00
 - File system type: ONTAP
 - Deployment type: Multi-AZ
 - Lifecycle state: Creating
- ONTAP administration:**
 - Management endpoint - DNS name: management.fs-08b2dec260faeca07.fsx.us-east-1.amazonaws.com
 - Management endpoint - IP address: 172.31.255.250
 - Inter-cluster endpoint - DNS name: intercluster.fs-08b2dec260faeca07.fsx.us-east-1.amazonaws.com
 - Inter-cluster endpoint - IP address: 172.31.32.38
 - ONTAP administrator username: fsxadmin
 - ONTAP administrator password: [Update]

3. Abra la instancia creada de **SageMaker Notebook** y haga clic en **Abrir JupyterLab**.

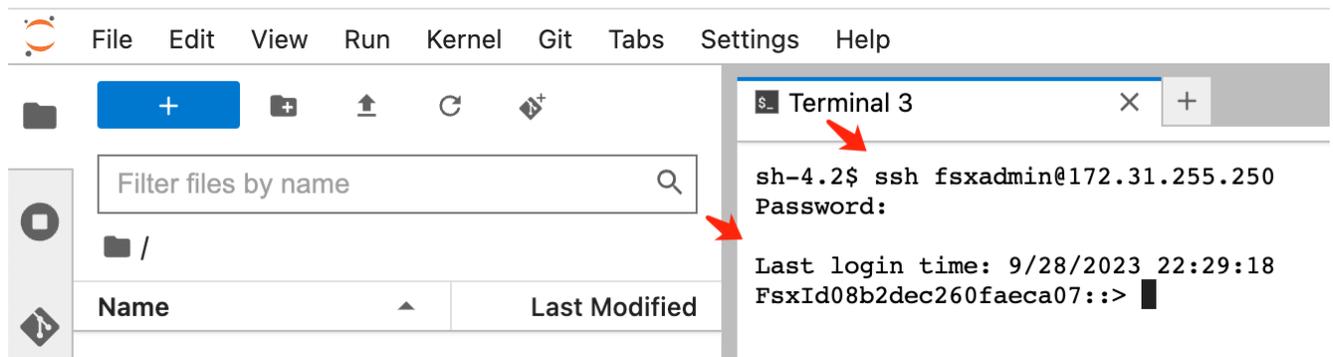
The screenshot shows the Amazon SageMaker console with the 'Notebook instances' page. A table lists the notebook instances:

Name	Instance	Creation time	Last updated	Status	Lifecycle config	Actions
fsxn-demo	ml.t3.medium	9/28/2023, 1:47:27 PM	9/28/2023, 1:50:28 PM	InService		Open Jupyter Open JupyterLab

4. En la página Jupyter Lab, abra un nuevo **Terminal**.



- Introduzca el comando `ssh ssh <nombre de usuario admin>@<IP de servidor ONTAP>` para iniciar sesión en el sistema de archivos FSx ONTAP. (El nombre de usuario y la dirección IP se recuperan del paso 2) Utilice la contraseña utilizada al crear la máquina virtual **Storage**.



- Ejecute los comandos en el siguiente orden. Utilizamos **fsxn-ONTAP** como nombre para el **FSX ONTAP private S3 bucket name**. Utilice el **nombre de máquina virtual de almacenamiento** para el argumento **-Vserver**.

```

vserver object-store-server create -vserver fsxn-svm-demo -object-store
-server fsx_s3 -is-http-enabled true -is-https-enabled false

vserver object-store-server user create -vserver fsxn-svm-demo -user
s3user

vserver object-store-server group create -name s3group -users s3user
-policies FullAccess

vserver object-store-server bucket create fsxn-ontap -vserver fsxn-svm-
demo -type nas -nas-path /vol1

```

```

sh-4.2$ ssh fsxadmin@172.31.255.250
Password:
Last login time: 9/28/2023 22:29:34
FxsId08b2dec260faeca07:~> vserver object-store-server create -vserver fsxn-svm-demo -object-store-server fsx_s3 -is-http-enabled true -is-https-enabled false
FxsId08b2dec260faeca07:~> vserver object-store-server user create -vserver fsxn-svm-demo -user s3user
FxsId08b2dec260faeca07:~> vserver object-store-server group create -name s3group -users s3user -policies FullAccess
FxsId08b2dec260faeca07:~> vserver object-store-server bucket create fsxn-ontap -vserver fsxn-svm-demo -type nas -nas-path /vol1
FxsId08b2dec260faeca07:~>

```

7. Ejecute los siguientes comandos para recuperar la IP de punto final y las credenciales para FSX ONTAP private S3.

```

network interface show -vserver fsxn-svm-demo -lif nfs_smb_management_1

set adv

vserver object-store-server user show

```

8. Conserve la IP del extremo y las credenciales para usarlo en el futuro.

```

sh-4.2$ ssh fsxadmin@172.31.255.250
Password:
Last login time: 9/28/2023 22:32:42
FxsId08b2dec260faeca07:~> network interface show -vserver fsxn-svm-demo -lif nfs_smb_management_1

Vserver Name: fsxn-svm-demo
Logical Interface Name: nfs_smb_management_1
Service Policy: default-data-files
Service List: data-core, data-nfs, data-cifs,
              management-ssh, management-https,
              data-s3-server, data-dns-server
(DEPRECATED)-Role: data
Data Protocol: nfs, cifs, s3
Network Address: Fsx IP Address
Netmask: 255.255.255.192
Bits in the Netmask: 26
Is VIP LIF: false
Subnet Name:
Home Node: FxsId08b2dec260faeca07-01
Home Port: e0e
Current Node: FxsId08b2dec260faeca07-01
Current Port: e0e
Operational Status: up
Extended Status: -
Is Home: true
Administrative Status: up
Failover Policy: system-defined
(DEPRECATED)-Firewall Policy: data
Auto Revert: true
Fully Qualified DNS Zone Name: none
DNS Query Listen Enable: false
Failover Group Name: Fsx
FCP WWPN: -
Address family: ipv4
Comment: -
IPspace of LIF: Default
Is Dynamic DNS Update Enabled?: true
Probe-port for Cloud Load Balancer: -
Broadcast Domain: Fsx
Vserver Type: data
Required RDMA offload protocols: -

FxsId08b2dec260faeca07:~> set adv

Warning: These advanced commands are potentially dangerous; use them only when directed to do so by NetApp personnel.
Do you want to continue? {y|n}: y

FxsId08b2dec260faeca07:~> vserver object-store-server user show
Vserver  User      ID      Access Key      Secret Key
-----  -
fsxn-svm-demo
  Comment: root User
fsxn-svm-demo
  s3user    1      AWS Access Key Id AWS Secret Access Key

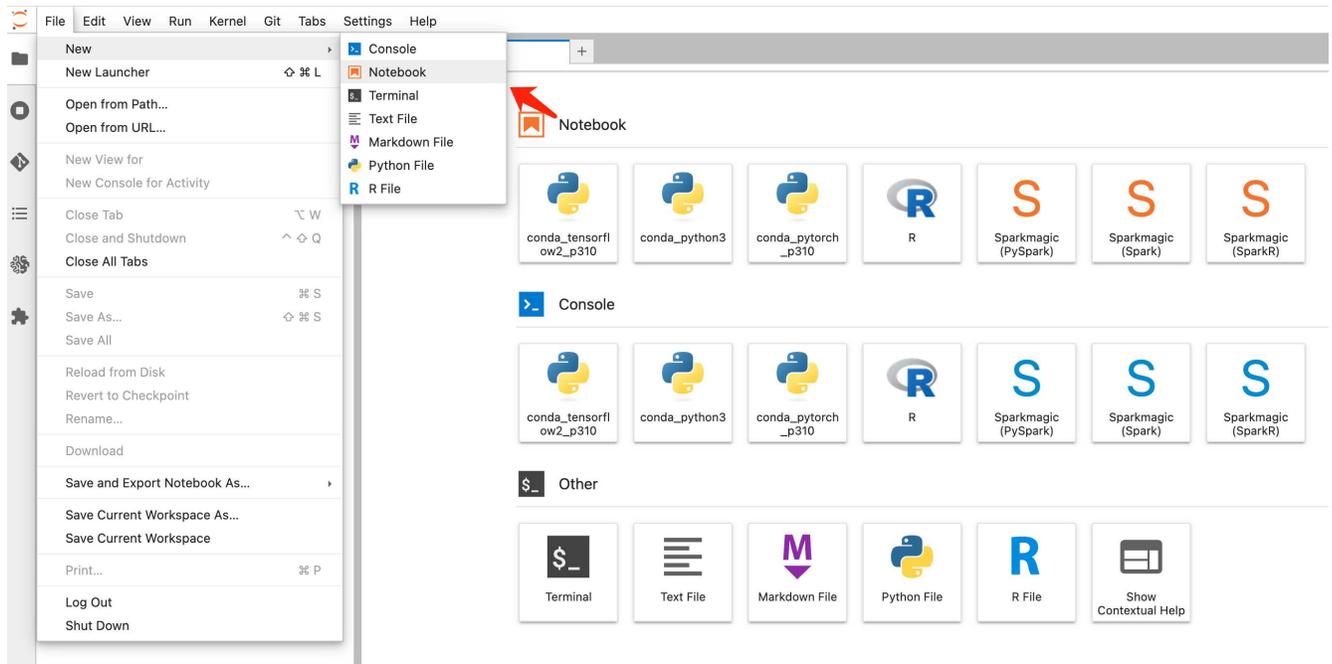
2 entries were displayed.

FxsId08b2dec260faeca07:~>

```

Configuración del cliente

1. En la instancia de SageMaker Notebook, cree un nuevo cuaderno Jupyter.



2. Utilice el siguiente código como solución de solución para cargar archivos en el cubo privado de S3 de FSx ONTAP. Para obtener un ejemplo de código completo, consulte este cuaderno. ["fsxn_demo.ipynb"](#)

```
# Setup configurations
# ----- Manual configurations -----
seed: int = 77 # Random
seed
bucket_name: str = 'fsxn-ontap' # The bucket
name in ONTAP
aws_access_key_id = '<Your ONTAP bucket key id>' # Please get
this credential from ONTAP
aws_secret_access_key = '<Your ONTAP bucket access key>' # Please get
this credential from ONTAP
fsx_endpoint_ip: str = '<Your FSx ONTAP IP address>' # Please get
this IP address from FSx ONTAP
# ----- Manual configurations -----

# Workaround
## Permission patch
!mkdir -p voll
!sudo mount -t nfs $fsx_endpoint_ip:/voll /home/ec2-user/SageMaker/voll
!sudo chmod 777 /home/ec2-user/SageMaker/voll

## Authentication for FSx ONTAP as a Private S3 Bucket
!aws configure set aws_access_key_id $aws_access_key_id
!aws configure set aws_secret_access_key $aws_secret_access_key
```

```

## Upload file to the FSx ONTAP Private S3 Bucket
%%capture
local_file_path: str = <Your local file path>

!aws s3 cp --endpoint-url http://$fsx_endpoint_ip /home/ec2-user
/SageMaker/$local_file_path s3://$bucket_name/$local_file_path

# Read data from FSx ONTAP Private S3 bucket
## Initialize a s3 resource client
import boto3

# Get session info
region_name = boto3.session.Session().region_name

# Initialize Fsx S3 bucket object
# --- Start integrating SageMaker with FSXN ---
# This is the only code change we need to incorporate SageMaker with
FSXN
s3_client: boto3.client = boto3.resource(
    's3',
    region_name=region_name,
    aws_access_key_id=aws_access_key_id,
    aws_secret_access_key=aws_secret_access_key,
    use_ssl=False,
    endpoint_url=f'http://{fsx_endpoint_ip}',
    config=boto3.session.Config(
        signature_version='s3v4',
        s3={'addressing_style': 'path'}
    )
)
# --- End integrating SageMaker with FSXN ---

## Read file byte content
bucket = s3_client.Bucket(bucket_name)

binary_data = bucket.Object(data.filename).get()['Body']

```

Esto concluye la integración entre FSX ONTAP y la instancia de SageMaker.

Lista de comprobación de depuración útil

- Asegúrese de que la instancia del bloc de notas de SageMaker y el sistema de archivos de FSX ONTAP estén en la misma VPC.
- Recuerde ejecutar el comando **set dev** en ONTAP para establecer el nivel de privilegio en **dev**.

Preguntas frecuentes (a partir del 27 de septiembre de 2023)

P: ¿Por qué recibo el error “**Se ha producido un error (NotImplemented) al llamar a la operación CreateMultipartUpload: El comando S3 que solicitó no está implementado**” al cargar archivos a FSX ONTAP?

R: Como depósito privado de S3, FSX ONTAP admite la carga de archivos de hasta 100MB GB. Cuando se utiliza el protocolo S3, los archivos de más de 100MB MB se dividen en 100MB fragmentos y se llama a la función 'CreateMultipartUpload'. Sin embargo, la implementación actual de FSX ONTAP PRIVATE S3 no admite esta función.

P: ¿Por qué recibo el error “**Se ha producido un error (ACCESSDENIED) al llamar a las operaciones PutObject: Acceso denegado**” al cargar archivos en FSX ONTAP?

R: Para acceder al bucket privado S3 de FSx ONTAP desde una instancia de bloc de notas de SageMaker, cambie las credenciales de AWS a las credenciales de FSx ONTAP. Sin embargo, otorgar permiso de escritura a la instancia requiere una solución provisional que implique montar el bucket y ejecutar el comando shell 'chmod' para cambiar los permisos.

P: ¿Cómo puedo integrar el cubo privado S3 de FSX ONTAP con otros servicios de SageMaker ML?

R: Desafortunadamente, el SDK de servicios de SageMaker no proporciona una forma de especificar el punto final para el cubo privado de S3. Como resultado, FSX ONTAP S3 no es compatible con los servicios de SageMaker como Sagemaker Data Wrangler, Sagemaker Clarify, Sagemaker Glue, Sagemaker Athena, Sagemaker AutoML y otros.

Parte 2: Aprovechamiento de AWS Amazon FSx para NetApp ONTAP (FSx ONTAP) como fuente de datos para el entrenamiento de modelos en SageMaker

Este artículo es un tutorial sobre el uso de Amazon FSx para NetApp ONTAP (FSX ONTAP) para el entrenamiento de los modelos de PyTorch en SageMaker, específicamente para un proyecto de clasificación de la calidad de los neumáticos.

Autor(es):

Jian Jian (KEN), científico sénior de datos y aplicado, NetApp

Introducción

Este tutorial ofrece un ejemplo práctico de un proyecto de clasificación de visión por computadora, proporcionando experiencia práctica en la construcción de modelos DE ML que utilizan FSX ONTAP como la fuente de datos dentro del entorno de SageMaker. El proyecto se centra en el uso de PyTorch, un marco de aprendizaje profundo, para clasificar la calidad de los neumáticos en función de las imágenes de los neumáticos. Hace hincapié en el desarrollo de modelos de aprendizaje automático utilizando FSx ONTAP como fuente de datos en Amazon SageMaker.

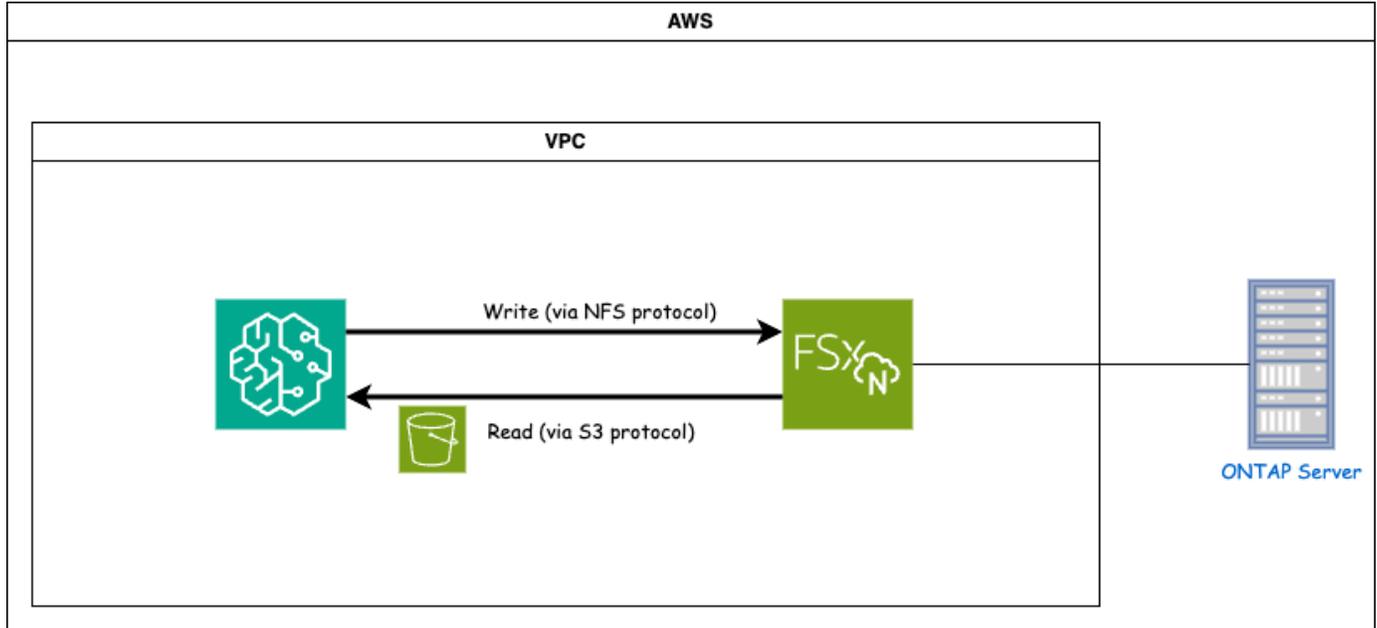
Qué es FSx ONTAP

Amazon FSx ONTAP es, de hecho, una solución de almacenamiento totalmente gestionada que ofrece AWS. Aprovecha el sistema de archivos ONTAP de NetApp para ofrecer un almacenamiento fiable y de alto rendimiento. Con su compatibilidad con protocolos como NFS, SMB e iSCSI, permite un acceso fluido desde diferentes instancias de computación y contenedores. El servicio está diseñado para ofrecer un rendimiento

excepcional, lo que garantiza operaciones de datos rápidas y eficaces. También ofrece alta disponibilidad y durabilidad, lo que garantiza que sus datos permanezcan accesibles y protegidos. Además, la capacidad de almacenamiento de Amazon FSx ONTAP es escalable, lo que te permite ajustarla fácilmente de acuerdo a tus necesidades.

Requisito previo

Entorno de red



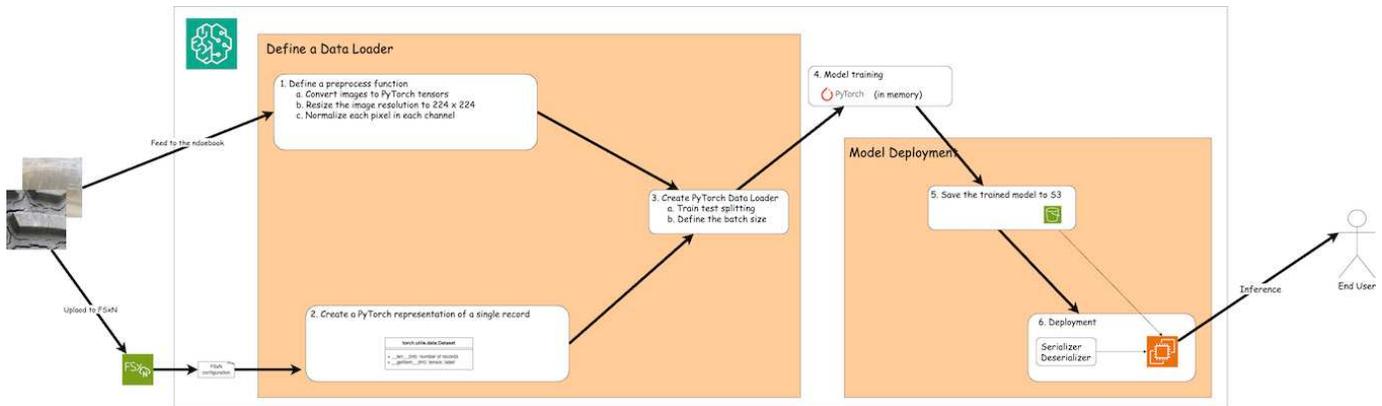
FSx ONTAP (Amazon FSx ONTAP) es un servicio de almacenamiento de AWS. Incluye un sistema de archivos que se ejecuta en el sistema NetApp ONTAP y una máquina virtual de sistema gestionado por AWS (SVM) que se conecta a él. En el diagrama proporcionado, el servidor NetApp ONTAP gestionado por AWS se encuentra fuera del VPC. El SVM sirve como intermediario entre SageMaker y el sistema NetApp ONTAP, al recibir solicitudes de operaciones de SageMaker y reenviarlas al almacenamiento subyacente. Para acceder a FSx ONTAP, SageMaker debe colocarse dentro de la misma VPC que la implementación de FSx ONTAP. Esta configuración garantiza la comunicación y el acceso a los datos entre SageMaker y FSx ONTAP.

Acceso a los datos

En escenarios del mundo real, los científicos de datos suelen utilizar los datos existentes almacenados en FSx ONTAP para construir sus modelos de aprendizaje automático. Sin embargo, para fines de demostración, dado que el sistema de archivos FSx ONTAP está inicialmente vacío después de la creación, es necesario cargar manualmente los datos de entrenamiento. Esto se puede lograr mediante el montaje de FSx ONTAP como un volumen para SageMaker. Una vez que el sistema de archivos se ha montado correctamente, puede cargar su conjunto de datos en la ubicación montada, lo que lo hace accesible para el entrenamiento de sus modelos dentro del entorno de SageMaker. Este enfoque te permite aprovechar la capacidad de almacenamiento y las funcionalidades de FSx ONTAP mientras trabajas con SageMaker para el desarrollo y entrenamiento de modelos.

El proceso de lectura de datos implica configurar FSx ONTAP como un bucket de S3 privado. Para obtener más información sobre las instrucciones de configuración detalladas, consulte "[Parte 1: Integrar Amazon FSx para NetApp ONTAP \(FSx ONTAP\) como bloque de S3 privado en AWS SageMaker](#)".

Visión General de la Integración



El flujo de trabajo de usar los datos de entrenamiento en FSx ONTAP para construir un modelo de aprendizaje profundo en SageMaker se puede resumir en tres pasos principales: Definición de Data Loader, entrenamiento de modelos y despliegue. En líneas generales, estos pasos forman la base de una canalización de MLOps. Sin embargo, cada paso implica varios subpasos detallados para una implementación integral. Estos subpasos abarcan diversas tareas, como el preprocesamiento de datos, la división de conjuntos de datos, la configuración del modelo, el ajuste de hiperparámetros, la evaluación de modelos, y la puesta en marcha de modelos. Estos pasos garantizan un proceso completo y efectivo para construir e implementar modelos de aprendizaje profundo utilizando datos de entrenamiento de FSx ONTAP dentro del entorno de SageMaker.

Integración paso a paso

Cargador de datos

Para entrenar una red de aprendizaje profundo de PyTorch con datos, se crea un cargador de datos para facilitar la alimentación de datos. El cargador de datos no sólo define el tamaño del lote, sino que también determina el procedimiento para leer y preprocesar cada registro del lote. Al configurar el cargador de datos, podemos manejar el procesamiento de datos en lotes, lo que permite el entrenamiento de la red de aprendizaje profundo.

El cargador de datos consta de 3 partes.

Función de preprocesamiento

```
from torchvision import transforms

preprocess = transforms.Compose([
    transforms.ToTensor(),
    transforms.Resize((224, 224)),
    transforms.Normalize(
        mean=[0.485, 0.456, 0.406],
        std=[0.229, 0.224, 0.225]
    )
])
```

El fragmento de código anterior demuestra la definición de las transformaciones de preprocesamiento de imágenes utilizando el módulo `torchvision.transform`. En este tutorial, se crea el objeto de preproceso para

aplicar una serie de transformaciones. En primer lugar, la transformación **ToTensor()** convierte la imagen en una representación tensora. Posteriormente, la transformación **Resize 224.224** cambia el tamaño de la imagen a un tamaño fijo de 224x224 píxeles. Finalmente, la transformación **Normalize()** normaliza los valores del tensor restando la media y dividiendo por la desviación estándar a lo largo de cada canal. Los valores de desviación media y estándar utilizados para la normalización se emplean comúnmente en modelos de redes neuronales pre-entrenados. En general, este código prepara los datos de la imagen para su posterior procesamiento o entrada en un modelo preentrenado convirtiéndolo en un tensor, ajustándolo y normalizando los valores de píxeles.

La clase de conjunto de datos de PyTorch

```
import torch
from io import BytesIO
from PIL import Image

class FSxNImageDataset(torch.utils.data.Dataset):
    def __init__(self, bucket, prefix='', preprocess=None):
        self.image_keys = [
            s3_obj.key
            for s3_obj in list(bucket.objects.filter(Prefix=prefix).all())
        ]
        self.preprocess = preprocess

    def __len__(self):
        return len(self.image_keys)

    def __getitem__(self, index):
        key = self.image_keys[index]
        response = bucket.Object(key)

        label = 1 if key[13:].startswith('defective') else 0

        image_bytes = response.get()['Body'].read()
        image = Image.open(BytesIO(image_bytes))
        if image.mode == 'L':
            image = image.convert('RGB')

        if self.preprocess is not None:
            image = self.preprocess(image)
        return image, label
```

Esta clase proporciona funcionalidad para obtener el número total de registros en el conjunto de datos y define el método para leer datos para cada registro. Dentro de la función **getitem**, el código utiliza el objeto `bucket boto3 S3` para recuperar los datos binarios de FSX ONTAP. El estilo de código para acceder a los datos de FSx ONTAP es similar a la lectura de datos de Amazon S3. La explicación subsiguiente profundiza en el proceso de creación del objeto privado S3 **bucket**.

FSX ONTAP como repositorio S3 privado

```
seed = 77 # Random seed
bucket_name = '<Your ONTAP bucket name>' # The bucket
name in ONTAP
aws_access_key_id = '<Your ONTAP bucket key id>' # Please get
this credential from ONTAP
aws_secret_access_key = '<Your ONTAP bucket access key>' # Please get
this credential from ONTAP
fsx_endpoint_ip = '<Your FSx ONTAP IP address>' # Please
get this IP address from FSXN
```

```
import boto3

# Get session info
region_name = boto3.session.Session().region_name

# Initialize FsxN S3 bucket object
# --- Start integrating SageMaker with FSXN ---
# This is the only code change we need to incorporate SageMaker with FSXN
s3_client: boto3.client = boto3.resource(
    's3',
    region_name=region_name,
    aws_access_key_id=aws_access_key_id,
    aws_secret_access_key=aws_secret_access_key,
    use_ssl=False,
    endpoint_url=f'http://{fsx_endpoint_ip}',
    config=boto3.session.Config(
        signature_version='s3v4',
        s3={'addressing_style': 'path'}
    )
)
# s3_client = boto3.resource('s3')
bucket = s3_client.Bucket(bucket_name)
# --- End integrating SageMaker with FSXN ---
```

Para leer datos de FSx ONTAP en SageMaker, se crea un manejador que señala al almacenamiento FSx ONTAP mediante el protocolo S3. Esto permite tratar el FSX ONTAP como un bucket privado de S3. La configuración del controlador incluye especificar la dirección IP de la SVM de FSx ONTAP, el nombre del bloque y las credenciales necesarias. Para obtener una explicación completa sobre la obtención de estos elementos de configuración, consulte el documento en ["1 parte: Integración de Amazon FSx para NetApp ONTAP \(FSx ONTAP\) como bloque de S3 privado en AWS SageMaker"](#).

En el ejemplo mencionado anteriormente, el objeto bucket se utiliza para instanciar el objeto de conjunto de datos PyTorch. El objeto del conjunto de datos se explicará con más detalle en la sección siguiente.

El cargador de datos de PyTorch

```
from torch.utils.data import DataLoader
torch.manual_seed(seed)

# 1. Hyperparameters
batch_size = 64

# 2. Preparing for the dataset
dataset = FSxNImageDataset(bucket, 'dataset/tyre', preprocess=preprocess)

train, test = torch.utils.data.random_split(dataset, [1500, 356])

data_loader = DataLoader(dataset, batch_size=batch_size, shuffle=True)
```

En el ejemplo proporcionado, se especifica un tamaño de lote de 64, lo que indica que cada lote contendrá 64 registros. Al combinar la clase PyTorch **Dataset**, la función de preprocesamiento y el tamaño de lote de entrenamiento, obtenemos el cargador de datos para el entrenamiento. Este cargador de datos facilita el proceso de iteración por el conjunto de datos en lotes durante la fase de entrenamiento.

Entrenamiento de modelos

```
from torch import nn

class TyreQualityClassifier(nn.Module):
    def __init__(self):
        super().__init__()
        self.model = nn.Sequential(
            nn.Conv2d(3, 32, (3, 3)),
            nn.ReLU(),
            nn.Conv2d(32, 32, (3, 3)),
            nn.ReLU(),
            nn.Conv2d(32, 64, (3, 3)),
            nn.ReLU(),
            nn.Flatten(),
            nn.Linear(64 * (224 - 6) * (224 - 6), 2)
        )
    def forward(self, x):
        return self.model(x)
```

```

import datetime

num_epochs = 2
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

model = TyreQualityClassifier()
fn_loss = torch.nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=1e-3)

model.to(device)
for epoch in range(num_epochs):
    for idx, (X, y) in enumerate(data_loader):
        X = X.to(device)
        y = y.to(device)

        y_hat = model(X)

        loss = fn_loss(y_hat, y)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
        current_time = datetime.datetime.now().strftime("%Y-%m-%d
%H:%M:%S")
        print(f"Current Time: {current_time} - Epoch [{epoch+1}/
{num_epochs}]- Batch [{idx + 1}] - Loss: {loss}", end='\r')

```

Este código implementa un proceso de entrenamiento estándar de PyTorch. Define un modelo de red neuronal llamado **TyreQualityClassifier** usando capas convolucionales y una capa lineal para clasificar la calidad de los neumáticos. El bucle de entrenamiento itera sobre los lotes de datos, calcula la pérdida y actualiza los parámetros del modelo mediante retropropagación y optimización. Además, imprime la hora actual, la época, el lote y la pérdida con fines de monitorización.

Puesta en marcha de modelos

Puesta en marcha

```

import io
import os
import tarfile
import sagemaker

# 1. Save the PyTorch model to memory
buffer_model = io.BytesIO()
traced_model = torch.jit.script(model)
torch.jit.save(traced_model, buffer_model)

# 2. Upload to AWS S3
sagemaker_session = sagemaker.Session()
bucket_name_default = sagemaker_session.default_bucket()
model_name = f'tyre_quality_classifier.pth'

# 2.1. Zip PyTorch model into tar.gz file
buffer_zip = io.BytesIO()
with tarfile.open(fileobj=buffer_zip, mode="w:gz") as tar:
    # Add PyTorch pt file
    file_name = os.path.basename(model_name)
    file_name_with_extension = os.path.splitext(file_name)[-1]
    tarinfo = tarfile.TarInfo(file_name_with_extension)
    tarinfo.size = len(buffer_model.getbuffer())
    buffer_model.seek(0)
    tar.addfile(tarinfo, buffer_model)

# 2.2. Upload the tar.gz file to S3 bucket
buffer_zip.seek(0)
boto3.resource('s3') \
    .Bucket(bucket_name_default) \
    .Object(f'pytorch/{model_name}.tar.gz') \
    .put(Body=buffer_zip.getvalue())

```

El código guarda el modelo de PyTorch en **Amazon S3** porque SageMaker requiere que el modelo se almacene en S3 para su implementación. Al subir el modelo a **Amazon S3**, se vuelve accesible para SageMaker, lo que permite la implementación e inferencia en el modelo desplegado.

```

import time
from sagemaker.pytorch import PyTorchModel
from sagemaker.predictor import Predictor
from sagemaker.serializers import IdentitySerializer
from sagemaker.deserializers import JSONDeserializer

class TyreQualitySerializer(IdentitySerializer):

```

```

CONTENT_TYPE = 'application/x-torch'

def serialize(self, data):
    transformed_image = preprocess(data)
    tensor_image = torch.Tensor(transformed_image)

    serialized_data = io.BytesIO()
    torch.save(tensor_image, serialized_data)
    serialized_data.seek(0)
    serialized_data = serialized_data.read()

    return serialized_data

class TyreQualityPredictor(Predictor):
    def __init__(self, endpoint_name, sagemaker_session):
        super().__init__(
            endpoint_name,
            sagemaker_session=sagemaker_session,
            serializer=TyreQualitySerializer(),
            deserializer=JSONDeserializer(),
        )

sagemaker_model = PyTorchModel(
    model_data=f's3://{bucket_name_default}/pytorch/{model_name}.tar.gz',
    role=sagemaker.get_execution_role(),
    framework_version='2.0.1',
    py_version='py310',
    predictor_cls=TyreQualityPredictor,
    entry_point='inference.py',
    source_dir='code',
)

timestamp = int(time.time())
pytorch_endpoint_name = '{}-{}-{}'.format('tyre-quality-classifier', 'pt',
timestamp)
sagemaker_predictor = sagemaker_model.deploy(
    initial_instance_count=1,
    instance_type='ml.p3.2xlarge',
    endpoint_name=pytorch_endpoint_name
)

```

Este código facilita el despliegue de un modelo PyTorch en SageMaker. Define un serializador personalizado, **TyreQualitySerializer**, que preprocesa y serializa los datos de entrada como un tensor PyTorch. La clase **TyreQualityPredictor** es un predictor personalizado que utiliza el serializador definido y un **JSONDeserializer**. El código también crea un objeto **PyTorchModel** para especificar la ubicación S3 del modelo, el rol IAM, la versión del marco y el punto de entrada para la inferencia. El código genera una marca

de tiempo y construye un nombre de punto final basado en el modelo y la marca de tiempo. Por último, el modelo se despliega mediante el método de despliegue, especificando el recuento de instancias, el tipo de instancia y el nombre de punto final generado. Esto permite que el modelo de PyTorch se despliegue y sea accesible para la inferencia en SageMaker.

Inferencia

```
image_object = list(bucket.objects.filter('dataset/tyre'))[0].get()
image_bytes = image_object['Body'].read()

with Image.open(with Image.open(BytesIO(image_bytes)) as image:
    predicted_classes = sagemaker_predictor.predict(image)

print(predicted_classes)
```

Este es el ejemplo de utilizar el punto final desplegado para llevar a cabo la inferencia.

Parte 3: Creación de Una canalización simplificada de MLOps (CI/CT/CD)

Este artículo proporciona una guía para crear una canalización de MLOps con servicios de AWS, centrándose en el reciclaje automatizado de modelos, la implementación y la optimización de costos.

Autor(es):

Jian Jian (KEN), científico sénior de datos y aplicado, NetApp

== Introducción

En este tutorial, aprenderá cómo aprovechar varios servicios de AWS para construir una canalización simple de MLOps que abarque la integración continua (CI), el entrenamiento continuo (CT) y la implementación continua (CD). A diferencia de las canalizaciones tradicionales de DevOps, MLOps requiere consideraciones adicionales para completar el ciclo operativo. Al seguir este tutorial, obtendrá información sobre la incorporación de CT en el bucle de MLOps, lo que permite el entrenamiento continuo de sus modelos y la implementación sin problemas para la inferencia. El tutorial le guiará a través del proceso de uso de los servicios de AWS para establecer este pipeline de MLOps de extremo a extremo.

== Manifiesto

```

|===
| Funcionalidad | Nombre | Comentar

| Almacenamiento de datos | ONTAP FSX de AWS | Consulte
xref:{relative_path}./mlops_fsxn_s3_integration.html["1 parte: Integración
de Amazon FSx para NetApp ONTAP (FSx ONTAP) como bloque de S3 privado en
AWS SageMaker"].

| IDE de ciencia de datos | SageMaker de AWS | Este tutorial se basa en el
cuaderno Jupyter presentado en
xref:{relative_path}./mlops_fsxn_sagemaker_integration_training.html["Part
e 2: Aprovechamiento de Amazon FSx para NetApp ONTAP (FSx ONTAP) como
fuente de datos para el entrenamiento de modelos en SageMaker"].

| Función para activar el pipeline de MLOps | Función AWS Lambda | -

| Disparador de trabajo CRON | EventBridge de AWS | -

| Marco de aprendizaje profundo | PyTorch | -

| SDK de AWS Python | boto3 | -

| Lenguaje de programación | Python | v3,10
|===

```

== Requisito previo

- * Un sistema de archivos FSX ONTAP preconfigurado. Este tutorial utiliza datos almacenados en FSx ONTAP para el proceso de entrenamiento.
- * Una instancia de bloc de notas *SageMaker* que está configurada para compartir la misma VPC que el sistema de archivos FSX ONTAP mencionado anteriormente.
- * Antes de activar la función *AWS Lambda*, asegúrese de que la instancia *SageMaker Notebook* esté en estado *Detenido*.
- * El tipo de instancia *ML.g4dn.xlarge* es necesario para aprovechar la aceleración de GPU necesaria para los cálculos de redes neuronales profundas.

== Arquitectura

image:mlops_fsxn_cictcd_0.png["Arquitectura"]

Esta canalización de MLOps es una implementación práctica que utiliza un trabajo cron para activar una función sin servidor, que a su vez ejecuta un servicio de AWS registrado con una función de devolución de llamada de ciclo de vida. El *AWS EventBridge* actúa como el trabajo cron. Invoca periódicamente una función *AWS Lambda* responsable de reciclar y reimplementar el modelo. Este proceso implica poner en marcha la instancia de *AWS SageMaker Notebook* para realizar las tareas necesarias.

== Configuración paso a paso

=== Configuraciones de ciclo de vida

Para configurar la función de devolución de llamada de ciclo de vida para la instancia de AWS SageMaker Notebook, utilizaría *Configuraciones de ciclo de vida*. Este servicio le permite definir las acciones necesarias que se deben realizar durante el giro de la instancia del bloc de notas. Específicamente, se puede implementar un script de shell dentro de las configuraciones de ciclo de vida *para cerrar automáticamente la instancia de notebook una vez que se completen los procesos de entrenamiento e implementación*. Esta es una configuración necesaria, ya que el coste es uno de los principales factores que hay que tener en cuenta en MLOps.

Es importante tener en cuenta que la configuración de *configuraciones de ciclo de vida* debe configurarse con antelación. Por lo tanto, se recomienda priorizar la configuración de este aspecto antes de continuar con la otra configuración de pipeline de MLOps.

. Para configurar una configuración de ciclo de vida, abra el panel *Sagemaker* y vaya a *Configuraciones de ciclo de vida* en la sección *Configuraciones de administración*.

+

image:mlops_fsxn_cictcd_1.png["Panel SageMaker"]

. Seleccione la pestaña *Instancia de bloc de notas* y haga clic en el botón *Crear configuración*

+

image:mlops_fsxn_cictcd_2.png["Página de bienvenida de configuración del ciclo de vida"]

. Pegue el siguiente código en el área de entrada.

+

```
[source, bash]
```

```
#!/bin/bash
```

```
set -e
```

```
sudo -u ec2-user -i <<'EOF'
```

```
# 1. Retraining and redeploying the model
```

```
NOTEBOOK_FILE=/home/ec2-user/SageMaker/tyre_quality_classification_local_training.ipynb
```

```
echo "Activating conda env"
```

```
source /home/ec2-user/anaconda3/bin/activate pytorch_p310
```

```
nohup jupyter nbconvert "$NOTEBOOK_FILE" --ExecutePreprocessor.kernel_name=python --execute --to notebook &
```

```
nbconvert_pid=$!
```

```
conda deactivate
```

2. Scheduling a job to shutdown the notebook to save the cost

```
PYTHON_DIR='/home/ec2-user/anaconda3/envs/JupyterSystemEnv/bin/python3.10'
```

```
echo "Starting the autostop script in cron"
```

```
(crontab -l 2>/dev/null; echo "*/5 * * * * bash -c 'if ps -p $nbconvert_pid > /dev/null; then echo \"Notebook is still running.\" >> /var/log/jupyter.log; else echo \"Notebook execution completed.\" >> /var/log/jupyter.log;
```

```
$PYTHON_DIR -c \"import
```

```
boto3;boto3.client('sagemaker').stop_notebook_instance(NotebookInstanceName=get_notebook_name())\"
```

```
>> /var/log/jupyter.log; fi\"") | crontab -
```

```
EOF
```

. Este script ejecuta el Jupyter Notebook, que se encarga del reciclaje y el redespigie del modelo para la inferencia. Una vez finalizada la ejecución, el bloc de notas se apagará automáticamente en 5 minutos. Para obtener más información sobre la declaración del problema y la implementación del código, consulte

xref:{relative_path}./mlops_fsxn_sagemaker_integration_training.html["Part e 2: Aprovechamiento de Amazon FSx para NetApp ONTAP (FSx ONTAP) como fuente de datos para el entrenamiento de modelos en SageMaker"].

+

image:mlops_fsxn_cictcd_3.png["Cree la configuración del ciclo de vida"]

. Después de la creación, navegue a Instancias de bloc de notas, seleccione la instancia de destino y haga clic en *Actualizar configuración* en el menú desplegable Acciones.

+

image:mlops_fsxn_cictcd_4.png["Lista desplegable Actualizar

```
configuración"]
```

. Seleccione la **Configuración de ciclo de vida** creada y haga clic en **Actualizar instancia de bloc de notas**.

+

```
image:mlops_fsxn_cictcd_5.png["Actualizar la configuración del ciclo de vida del portátil"]
```

=== Función sin servidor de AWS Lambda

Como se mencionó anteriormente, la función **AWS Lambda** es responsable de poner en funcionamiento la instancia **AWS SageMaker Notebook**.

. Para crear una función **AWS Lambda**, navegue hasta el panel correspondiente, cambie a la pestaña **Funciones** y haga clic en **Crear función**.

+

```
image:mlops_fsxn_cictcd_6.png["Página de bienvenida de la función AWS lambda"]
```

. Por favor, archiva todas las entradas requeridas en la página y recuerda cambiar el tiempo de ejecución a **Python 3,10**.

+

```
image:mlops_fsxn_cictcd_7.png["Cree una función lambda de AWS"]
```

. Verifique que el rol designado tiene el permiso requerido **AmazonSageMakerFullAccess** y haga clic en el botón **Crear función**.

+

```
image:mlops_fsxn_cictcd_8.png["Seleccione el rol de ejecución"]
```

. Seleccione la función Lambda creada. En la pestaña de código, copie y pegue el siguiente código en el área de texto. Este código inicia la instancia de notebook llamada **fsxn-ontap**.

+

```
[source, python]
```

```
import boto3
import logging
```

```
def lambda_handler(event, context):
    client = boto3.client('sagemaker')
    logging.info('Invoking SageMaker')
    client.start_notebook_instance(NotebookInstanceName='fsxn-ontap')
```

```
return {
'statusCode': 200,
'body': f'Starting notebook instance: {notebook_instance_name}'
}
```

. Haga clic en el botón *Desplegar* para aplicar este cambio de código.
+
image:mlops_fsxn_cictcd_9.png["Puesta en marcha"]

. Para especificar cómo activar esta función de AWS Lambda, haga clic en el botón Agregar Disparador.

+
image:mlops_fsxn_cictcd_10.png["Agregar disparador de función AWS"]

. Seleccione EventBridge en el menú desplegable y, a continuación, haga clic en el botón de opción con la etiqueta Crear una nueva regla. En el campo de expresión de programación, introduzca `rate(1 day)`, Y haga clic en el botón Agregar para crear y aplicar esta nueva regla de trabajo cron a la función AWS Lambda.

+
image:mlops_fsxn_cictcd_11.png["Finalizar disparador"]

Después de completar la configuración en dos pasos, diariamente, la función *AWS Lambda* iniciará el *SageMaker Notebook*, realizará el reciclaje del modelo utilizando los datos del repositorio *FSX ONTAP*, volverá a desplegar el modelo actualizado en el entorno de producción y cerrará automáticamente la instancia *SageMaker Notebook* para optimizar los costos. Esto garantiza que el modelo permanezca actualizado.

Esto concluye el tutorial para desarrollar un pipeline de MLOps.

:leveloffset: -1

:leveloffset: -1

<<<

Información de copyright

Copyright © 2024 NetApp, Inc. Todos los derechos reservados. Imprimido en EE. UU. No se puede reproducir este documento protegido por copyright ni parte del mismo de ninguna forma ni por ningún medio (gráfico, electrónico o mecánico, incluidas fotocopias, grabaciones o almacenamiento en un

sistema de recuperación electrónico) sin la autorización previa y por escrito del propietario del copyright.

El software derivado del material de NetApp con copyright está sujeto a la siguiente licencia y exención de responsabilidad:

ESTE SOFTWARE LO PROPORCIONA NETAPP «TAL CUAL» Y SIN NINGUNA GARANTÍA EXPRESA O IMPLÍCITA, INCLUYENDO, SIN LIMITAR, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN CONCRETO, CUYA RESPONSABILIDAD QUEDA EXIMIDA POR EL PRESENTE DOCUMENTO. EN NINGÚN CASO NETAPP SERÁ RESPONSABLE DE NINGÚN DAÑO DIRECTO, INDIRECTO, ESPECIAL, EJEMPLAR O RESULTANTE (INCLUYENDO, ENTRE OTROS, LA OBTENCIÓN DE BIENES O SERVICIOS SUSTITUTIVOS, PÉRDIDA DE USO, DE DATOS O DE BENEFICIOS, O INTERRUPCIÓN DE LA ACTIVIDAD EMPRESARIAL) CUALQUIERA SEA EL MODO EN EL QUE SE PRODUJERON Y LA TEORÍA DE RESPONSABILIDAD QUE SE APLIQUE, YA SEA EN CONTRATO, RESPONSABILIDAD OBJETIVA O AGRAVIO (INCLUIDA LA NEGLIGENCIA U OTRO TIPO), QUE SURJAN DE ALGÚN MODO DEL USO DE ESTE SOFTWARE, INCLUSO SI HUBIEREN SIDO ADVERTIDOS DE LA POSIBILIDAD DE TALES DAÑOS.

NetApp se reserva el derecho de modificar cualquiera de los productos aquí descritos en cualquier momento y sin aviso previo. NetApp no asume ningún tipo de responsabilidad que surja del uso de los productos aquí descritos, excepto aquello expresamente acordado por escrito por parte de NetApp. El uso o adquisición de este producto no lleva implícita ninguna licencia con derechos de patente, de marcas comerciales o cualquier otro derecho de propiedad intelectual de NetApp.

Es posible que el producto que se describe en este manual esté protegido por una o más patentes de EE. UU., patentes extranjeras o solicitudes pendientes.

LEYENDA DE DERECHOS LIMITADOS: el uso, la copia o la divulgación por parte del gobierno están sujetos a las restricciones establecidas en el subpárrafo (b) (3) de los derechos de datos técnicos y productos no comerciales de DFARS 252.227-7013 (FEB de 2014) y FAR 52.227-19 (DIC de 2007).

Los datos aquí contenidos pertenecen a un producto comercial o servicio comercial (como se define en FAR 2.101) y son propiedad de NetApp, Inc. Todos los datos técnicos y el software informático de NetApp que se proporcionan en este Acuerdo tienen una naturaleza comercial y se han desarrollado exclusivamente con fondos privados. El Gobierno de EE. UU. tiene una licencia limitada, irrevocable, no exclusiva, no transferible, no sublicenciable y de alcance mundial para utilizar los Datos en relación con el contrato del Gobierno de los Estados Unidos bajo el cual se proporcionaron los Datos. Excepto que aquí se disponga lo contrario, los

Datos no se pueden utilizar, desvelar, reproducir, modificar, interpretar o mostrar sin la previa aprobación por escrito de NetApp, Inc. Los derechos de licencia del Gobierno de los Estados Unidos de América y su Departamento de Defensa se limitan a los derechos identificados en la cláusula 252.227-7015(b) de la sección DFARS (FEB de 2014).

Información de la marca comercial

NETAPP, el logotipo de NETAPP y las marcas que constan en [link:http://www.netapp.com/TM](http://www.netapp.com/TM)^[http://www.netapp.com/TM^] son marcas comerciales de NetApp, Inc. El resto de nombres de empresa y de producto pueden ser marcas comerciales de sus respectivos propietarios.