Canalizaciones de datos, lagos de datos y gestión

NetApp Solutions

NetApp
April 24, 2024

This PDF was generated from https://docs.netapp.com/es-es/netapp-solutions/ai/mlops_fsxn_s3_integration.html on April 24, 2024. Always check docs.netapp.com for the latest.

Tabla de contenidos

Canalizaciones de datos, lagos de datos y gestión . . . 1
AWS FSx para NetApp ONTAP (FSxN) para MLOps (en inglés) . . . 1
MLOps de multinube híbrida con Domino Data Lab y NetApp . . . 38
NVIDIA AI Enterprise con NetApp y VMware . . . 53
TR-4851: Lago de datos de StorageGRID de NetApp para cargas de trabajo de conducción autónoma - Diseño de la solución . . . 63
Plano de control de IA de NetApp . . . 63
MLRun Pipeline con Iguazio . . . 121
TR-4915: Movimiento de datos con E-Series y BeeGFS para flujos de trabajo de análisis e IA . . . 148

Canalizaciones de datos, lagos de datos y gestión

AWS FSx para NetApp ONTAP (FSxN) para MLOps (en inglés)

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

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

Estos artículos se centran en:

1. "1 parte: Integración de AWS FSx para NetApp ONTAP (FSxN) como bloque de S3 privado en AWS SageMaker"
2. "Parte 2: Aprovechamiento de AWS FSx para NetApp ONTAP (FSxN) 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, habrá obtenido una sólida comprensión de cómo usar FSxN para agilizar los procesos de MLOps.

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

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

Introducción

Utilizando SageMaker como ejemplo, esta página proporciona una guía para configurar FSxN como un bucket privado de S3.

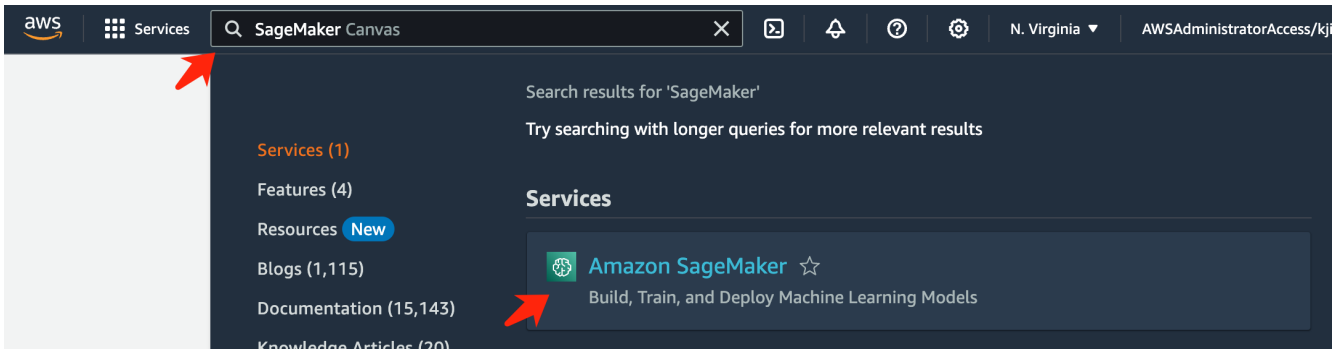
Para obtener más información sobre FSxN, por favor eche un vistazo a esta presentación ("Enlace de vídeo")

Guía del usuario

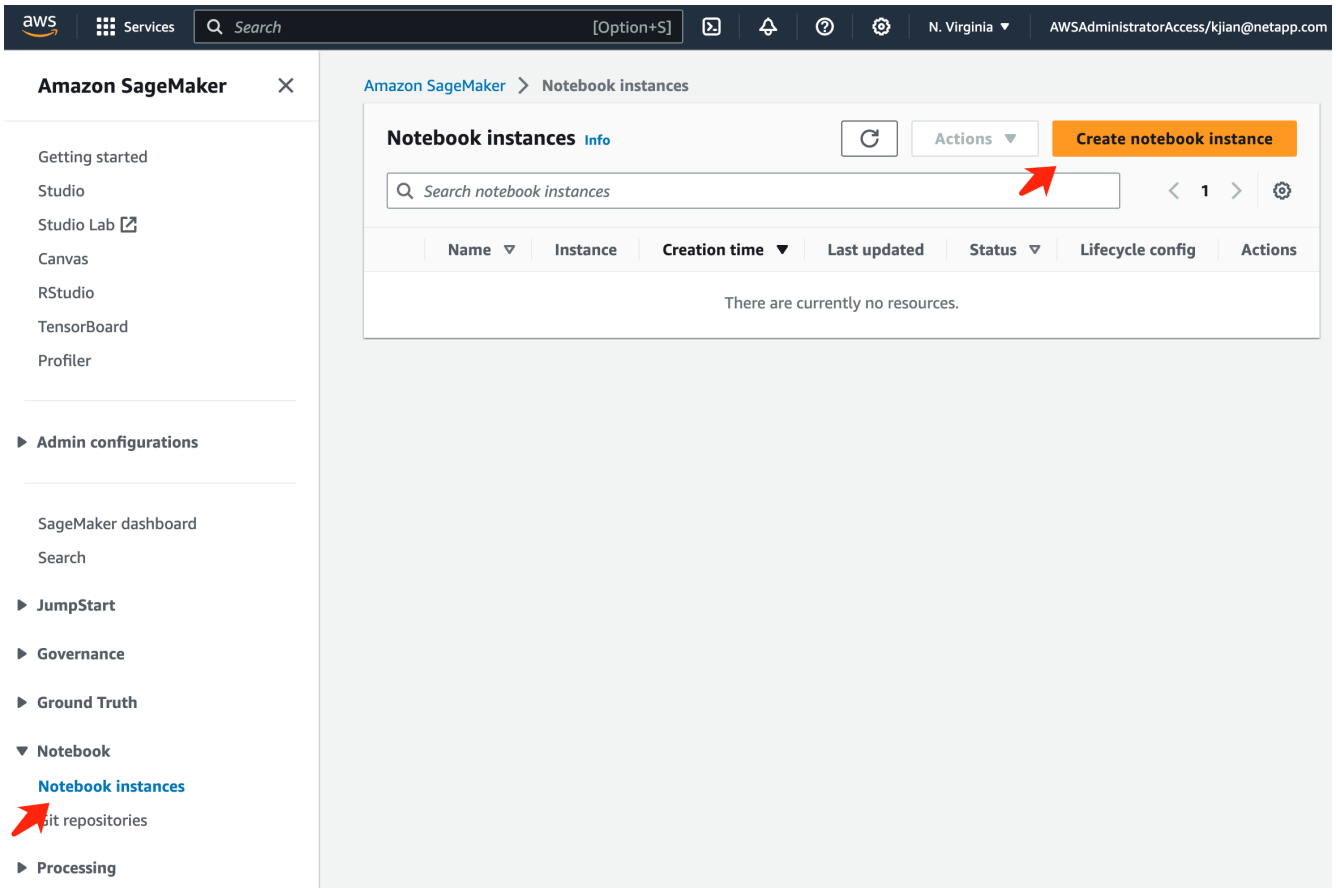
Creación de servidores

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,
   Introduzca el nombre de la instancia de **Notebook**
   Expanda el panel **Network**
   Deje otras entradas predeterminadas y seleccione un **VPC**, **Subnet** y **Grupo(s) de seguridad**. (Este **VPC** y **Subnet** se utilizarán para crear el sistema de archivos FSxN 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) ✕

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

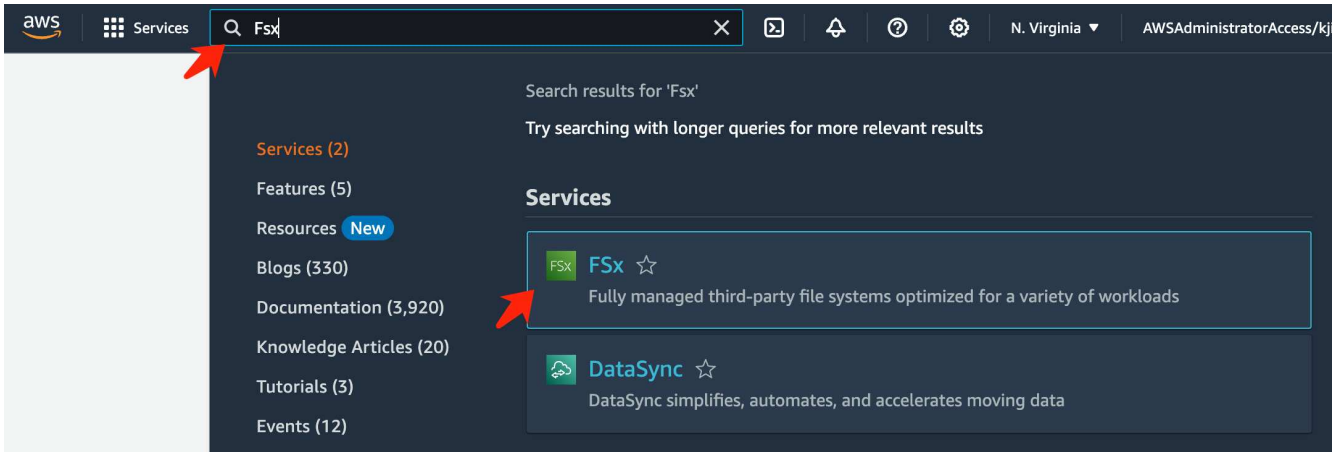
▶ Git repositories- *optional*

▶ Tags - *optional*

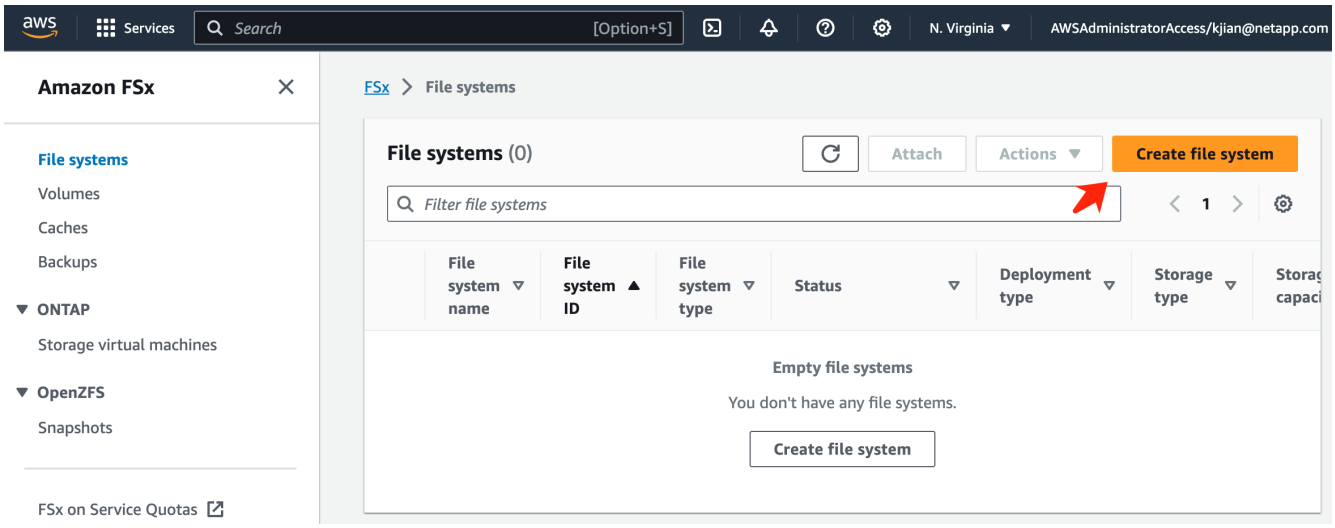
Cancel | Create notebook instance

Crear un sistema de archivos FSxN

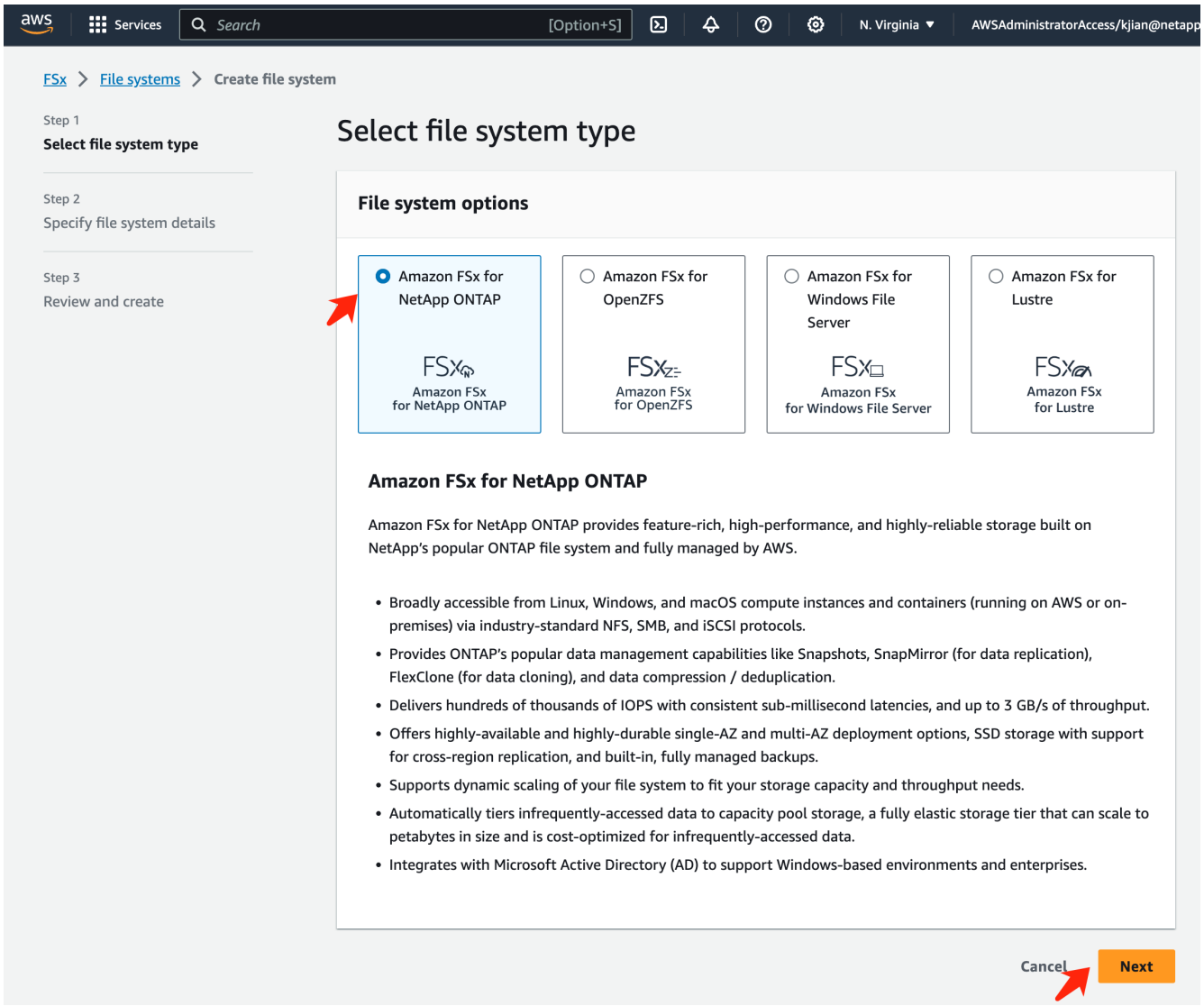
1. Abra la consola de AWS. En el panel de búsqueda, busca FSX y haz clic en el servicio **FSX**.



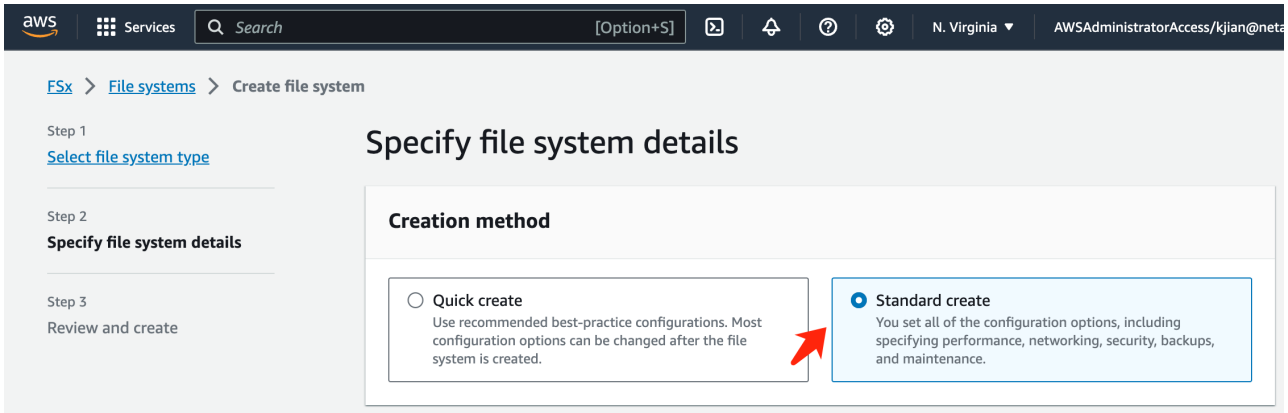
2. Haga clic en **Crear sistema de archivos**.



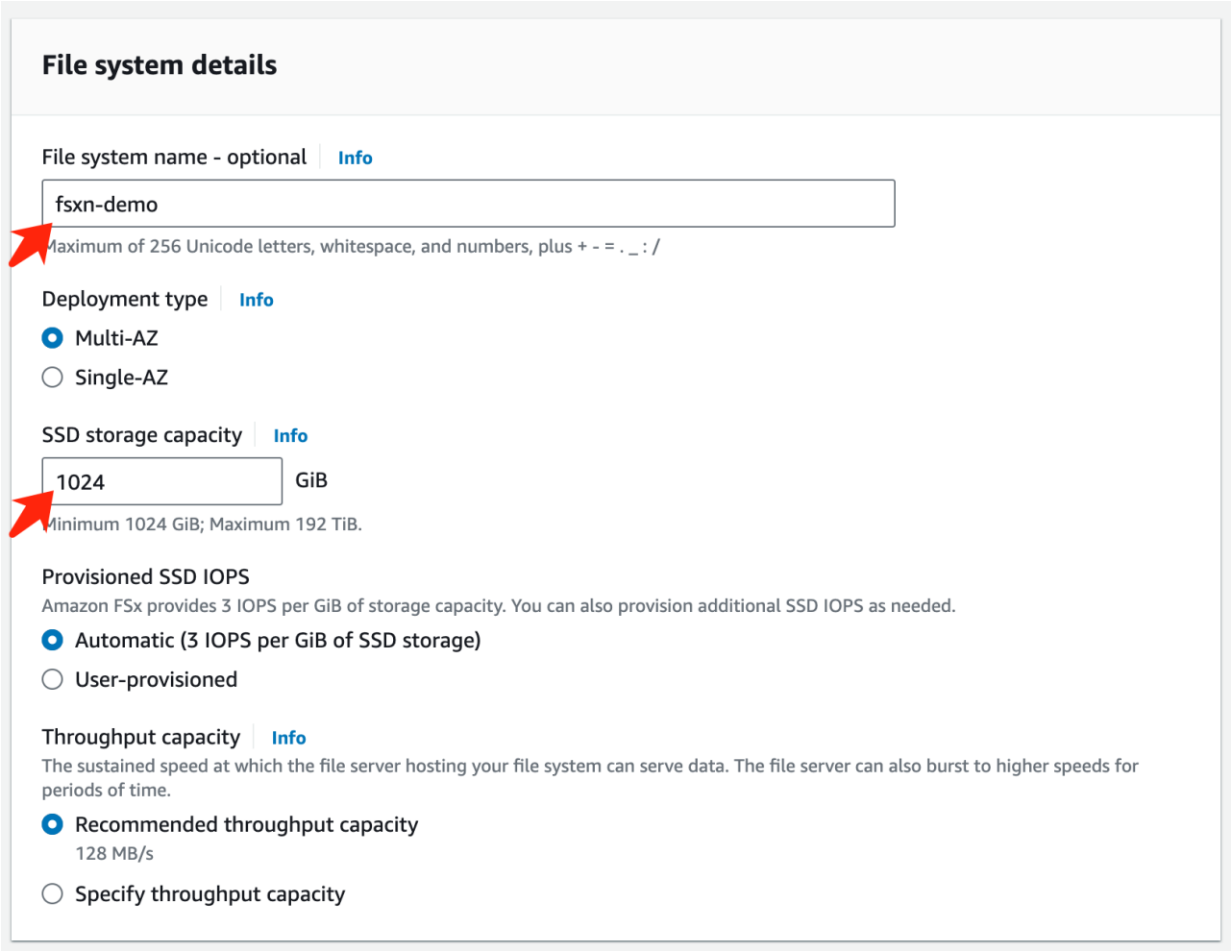
3. Seleccione la primera tarjeta **FSX for NetApp ONTAP** y haga clic en **Siguiente**.



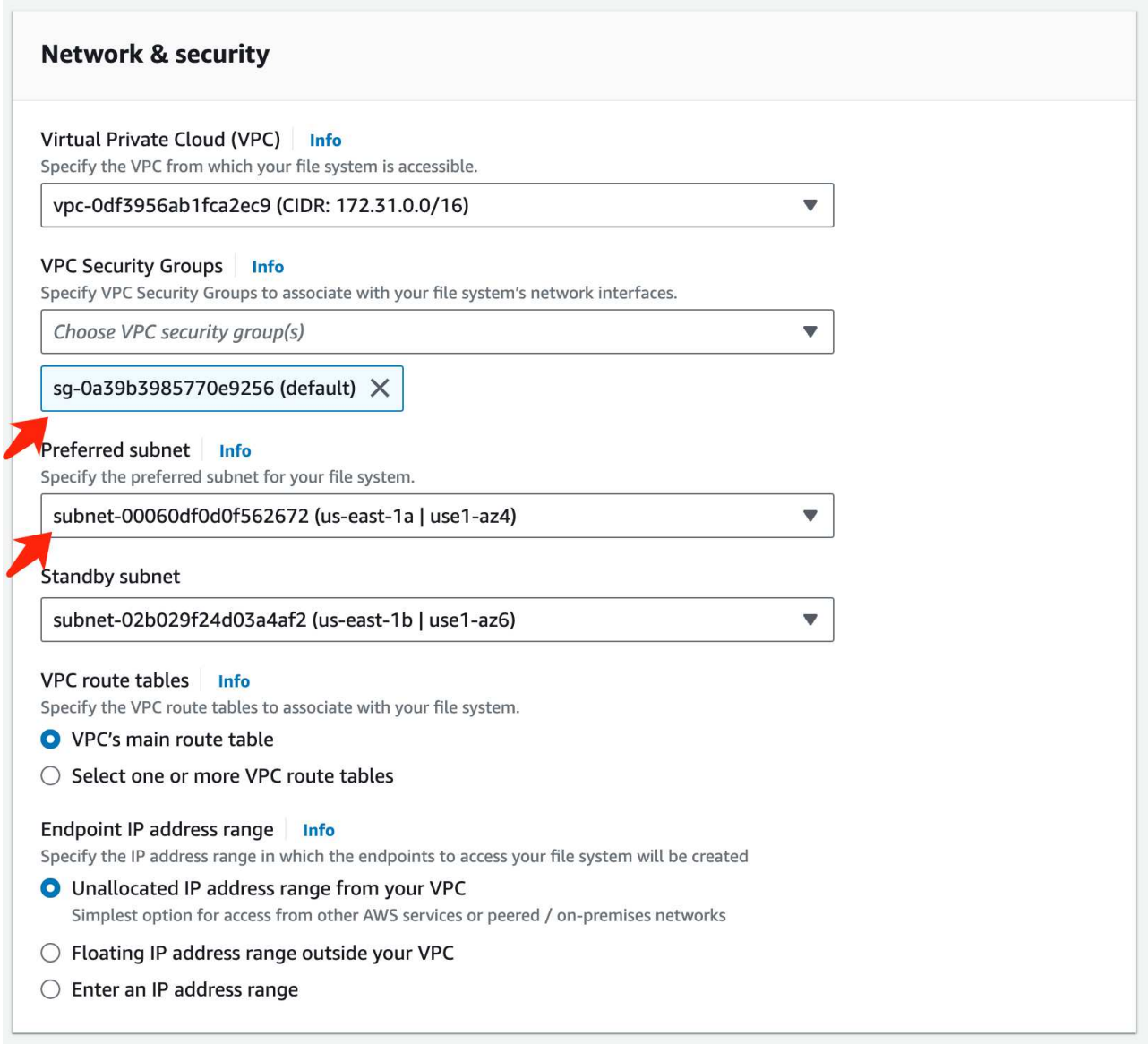
4. En la página de configuración de detalles.
   a. Seleccione la opción **Standard create**.



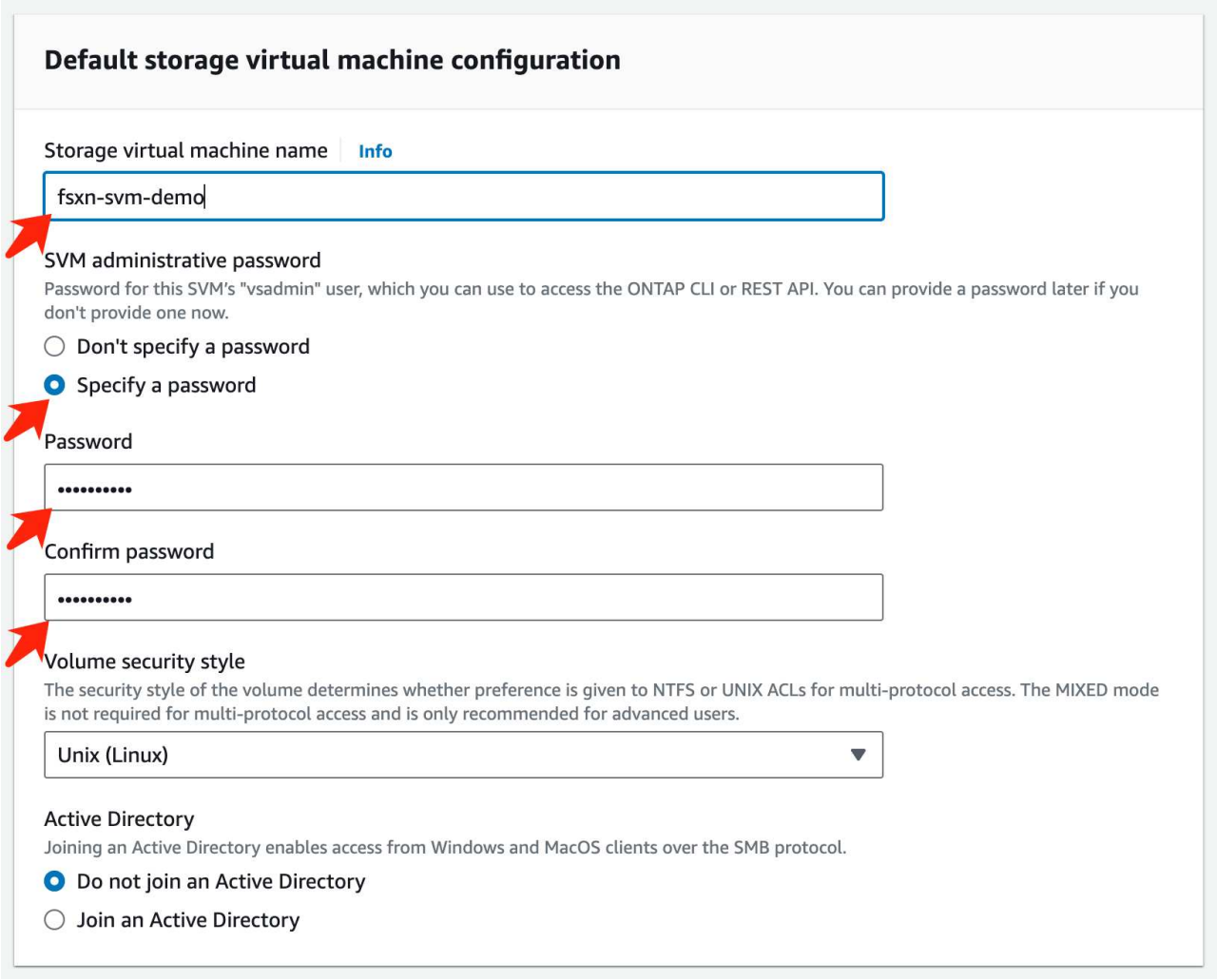
   b. Introduzca el **Nombre del sistema de archivos** y la **Capacidad de almacenamiento SSD**.



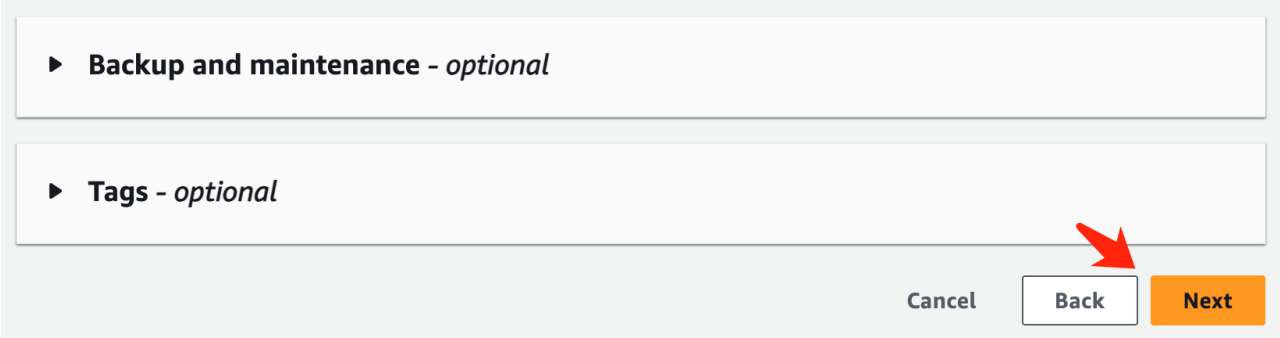
c. Asegúrese de usar **VPC** y **subnet** igual a la instancia **SageMaker Notebook**.



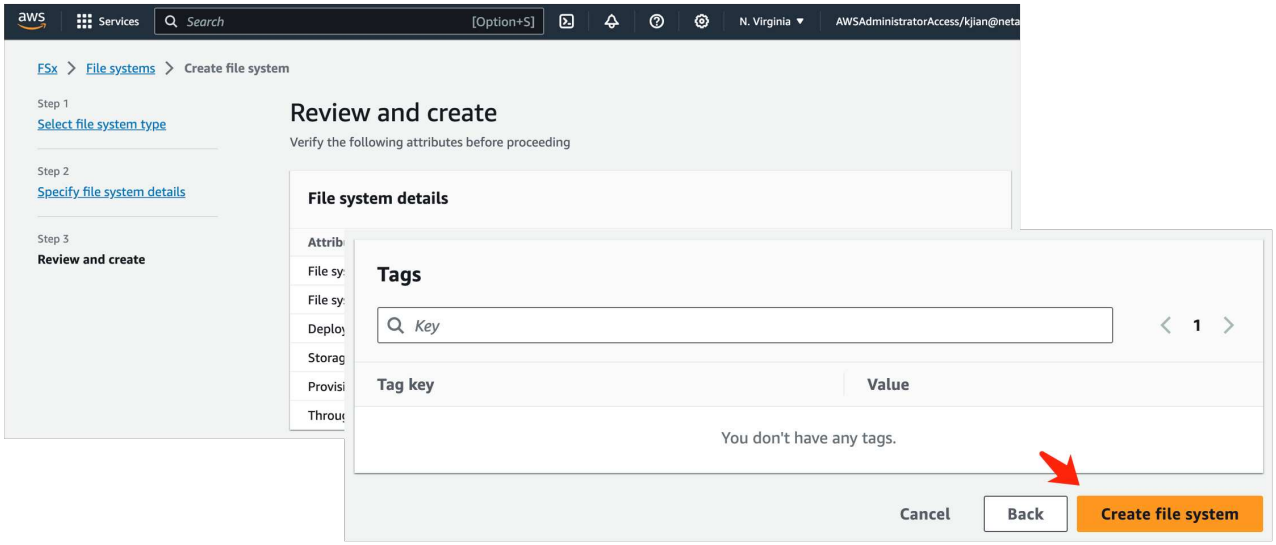
d. Introduzca el nombre de la máquina virtual **Storage** y **especifique una contraseña** para su SVM (máquina virtual de almacenamiento).



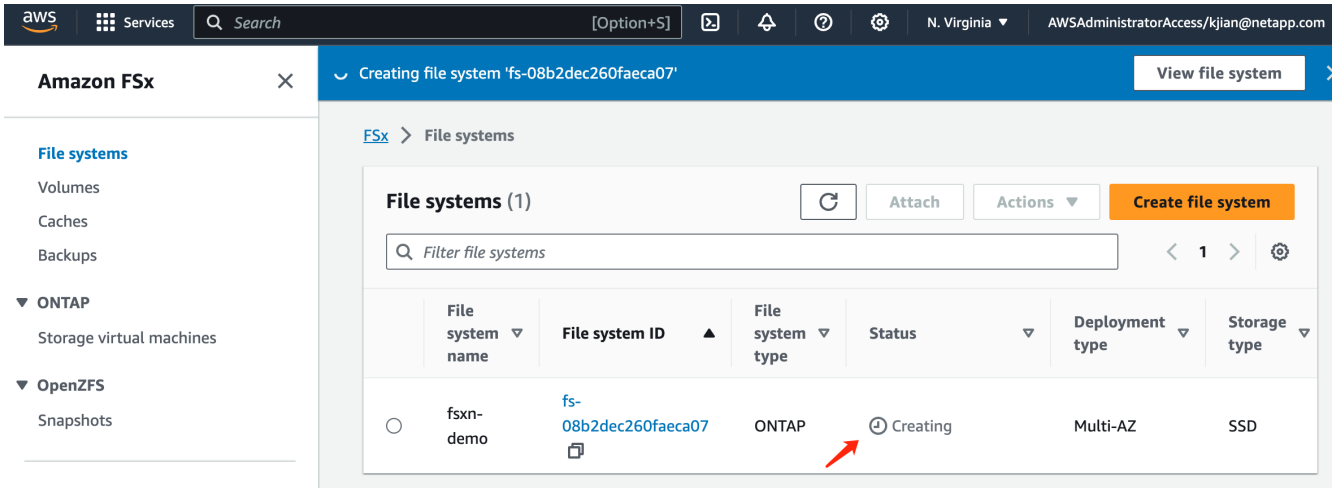
e. Deja otras entradas predeterminadas y haz clic en el botón naranja **Siguiente** en la parte inferior derecha.



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



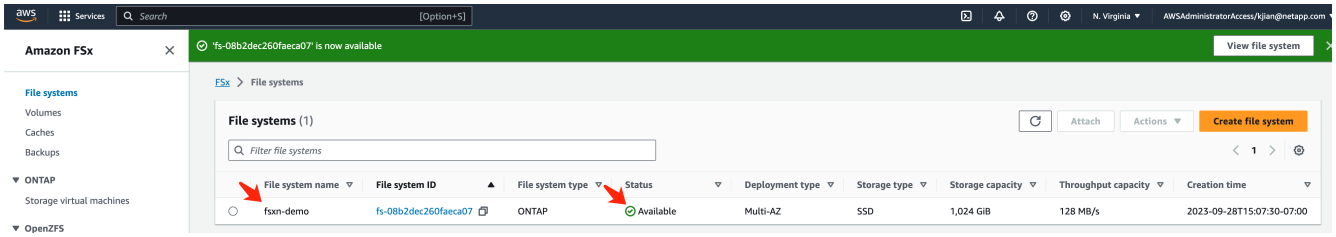
5. Puede tardar unos **20-40 minutos** en activar el sistema de archivos FSX.



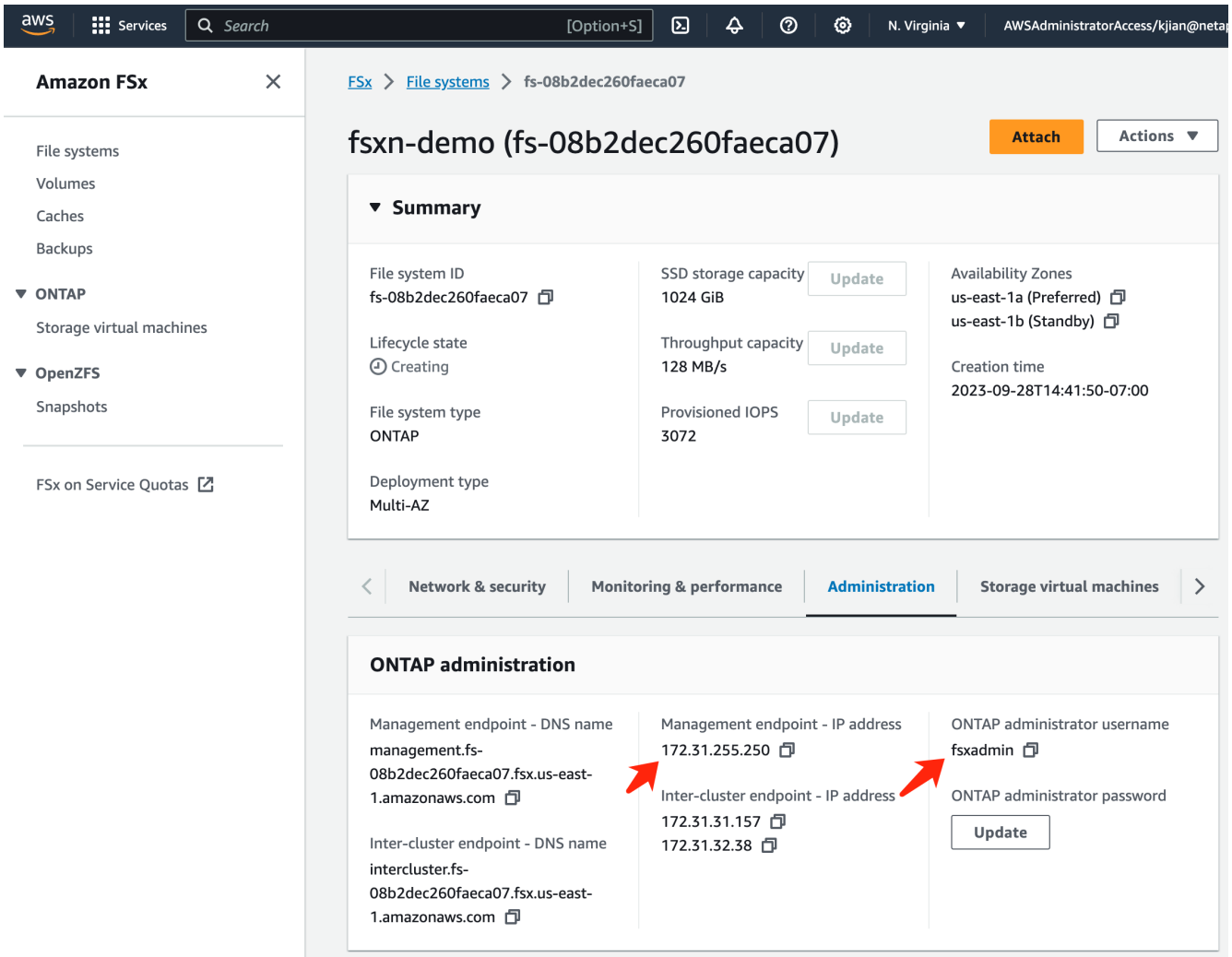
Configuración del servidor

Configuración de ONTAP

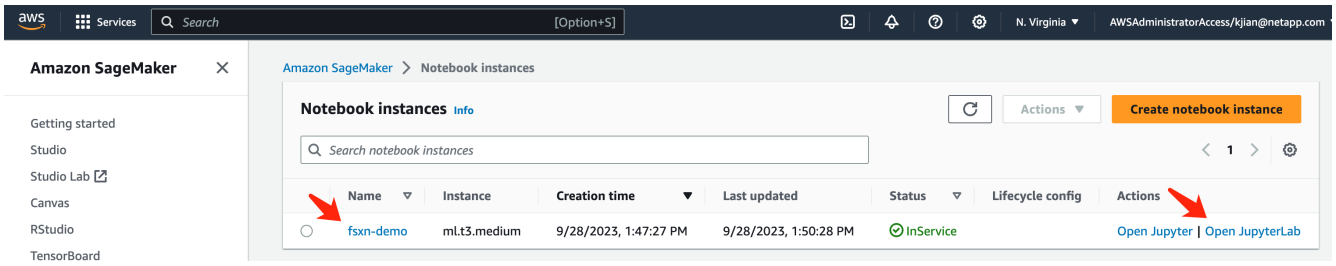
1. Abra el sistema de archivos FSX creado. Por favor, asegúrese de que el estado es **disponible**.



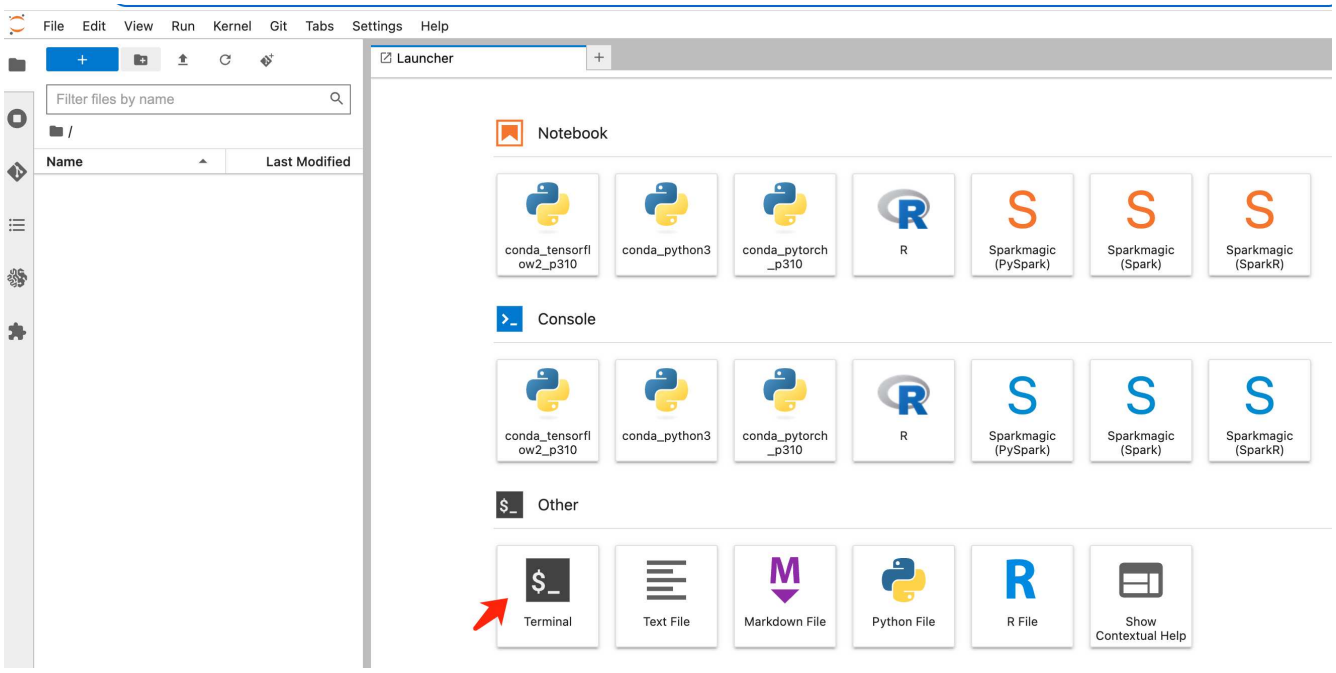
2. Seleccione la pestaña **Administración** y mantenga el **Punto final de administración - dirección IP** y el **Nombre de usuario del administrador de ONTAP**.



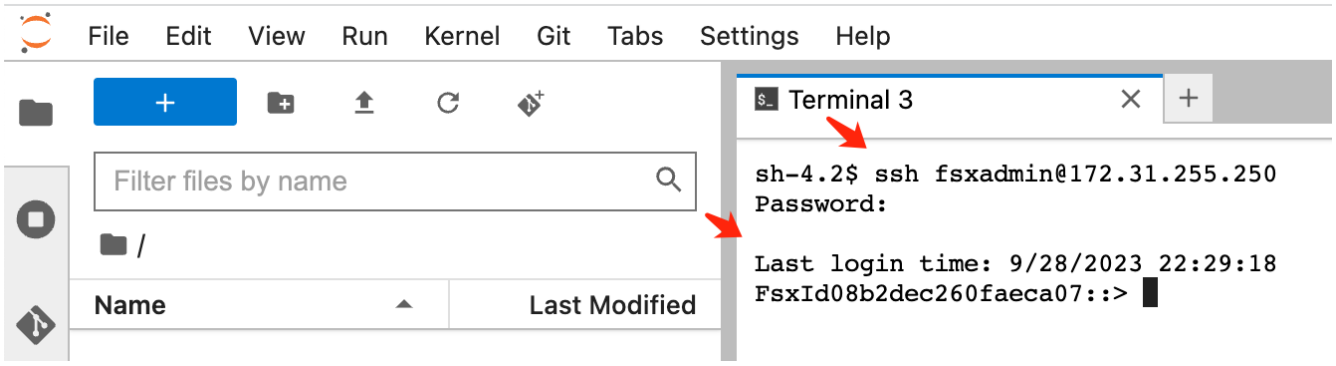
3. Abra la instancia creada de **SageMaker Notebook** y haga clic en **Abrir JupyterLab**.



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

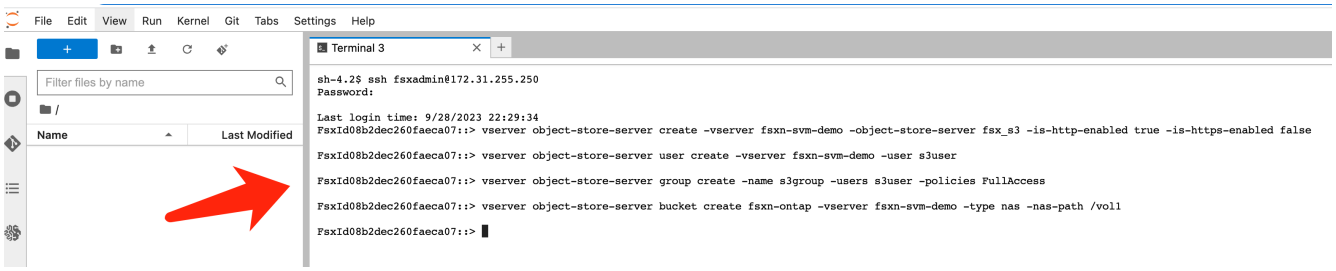


5. Introduzca el comando ssh ssh <nombre de usuario admin>@<IP del servidor de ONTAP> para iniciar sesión en el sistema de archivos ONTAP FSxN. (El nombre de usuario y la dirección IP se recuperan del paso 2)
   Utilice la contraseña utilizada al crear la **Storage virtual machine**.



6. Ejecute los comandos en el siguiente orden.
   Utilizamos **fsxn-ontap** como nombre para el **FSxN 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
```



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

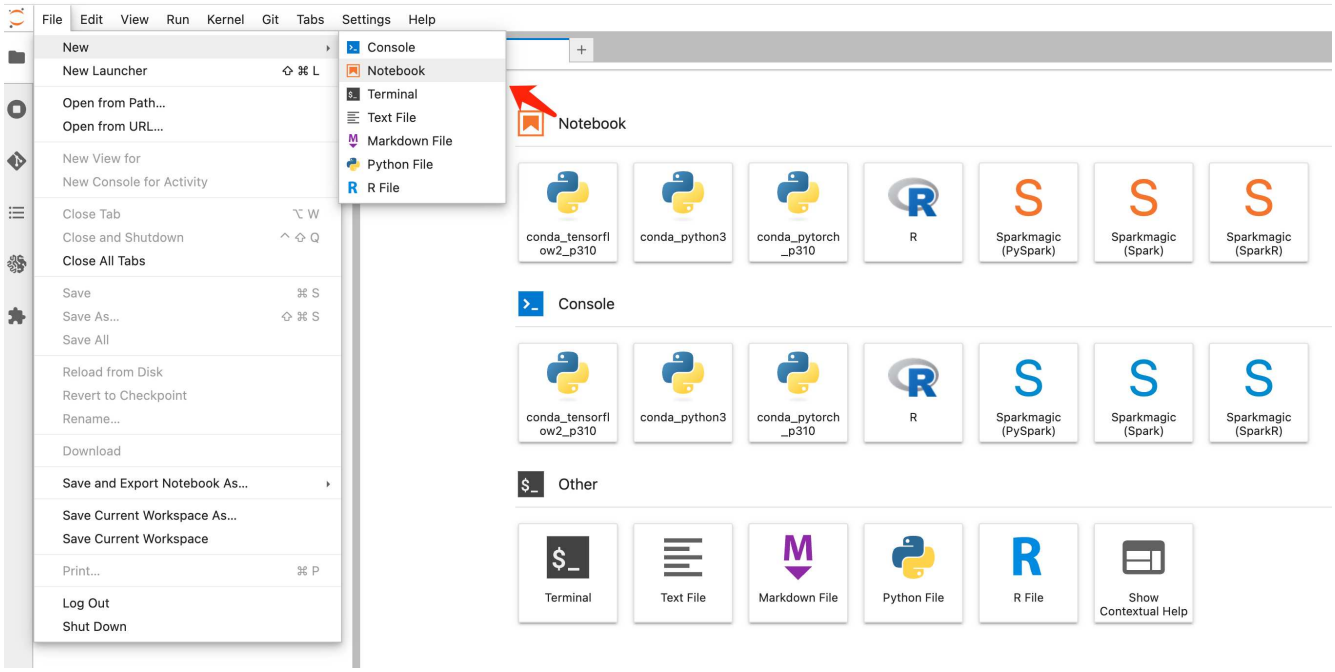
```
network interface show -vserver fsxn-svm-demo -lif nfs_smb_management_1

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
FsxId08b2dec260faeca07::> 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: [redacted]
                          Netmask: 255.255.255.192
              Bits in the Netmask: 26
                       Is VIP LIF: false
                      Subnet Name: -
                        Home Node: FsxId08b2dec260faeca07-01
                        Home Port: e0e
                     Current Node: FsxId08b2dec260faeca07-01
                     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: -

FsxId08b2dec260faeca07::> 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

FsxId08b2dec260faeca07::*> vserver object-store-server user show
Vserver     User            ID        Access Key           Secret Key
----------- --------------- --------- -------------------- --------------------
fsxn-svm-demo
            root            0         -                    -
   Comment: Root User
fsxn-svm-demo
            s3user          1         [redacted]           [redacted]

2 entries were displayed.

FsxId08b2dec260faeca07::*>
```

Configuración del cliente

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



2. Utilice el siguiente código como solución alternativa para cargar archivos en el cubo privado de FSxN S3. 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 FSxN IP address>'                 # Please get
this IP address from FSXN
# -------- Manual configurations --------

# Workaround
## Permission patch
!mkdir -p vol1
!sudo mount -t nfs $fsx_endpoint_ip:/vol1 /home/ec2-user/SageMaker/vol1
!sudo chmod 777 /home/ec2-user/SageMaker/vol1

## Authentication for FSxN 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 FSxN 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 FSxN Private S3 bucket
## Initialize a s3 resource client
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'}
    )
)
# --- 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 FSxN y la instancia de SageMaker.

Lista de comprobación de depuración útil

- Asegúrese de que la instancia de SageMaker Notebook y el sistema de archivos FSxN 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 FSxN?

R: Como depósito privado de S3, FSxN 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 FSxN PRIVATE S3 no soporta 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 a FSxN?

R: Para acceder al bucket S3 privado FSxN desde una instancia de Notebook de SageMaker, cambie las credenciales de AWS a las credenciales FSxN. 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 FSxN private S3 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, FSxN S3 no es compatible con los servicios de SageMaker tales como Sagemaker Data Wrangler, Sagemaker Clarify, Sagemaker Glue, Sagemaker Athena, Sagemaker

AutoML, Sagemaker AutoML, y otros.

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

Autor(es):

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

Introducción

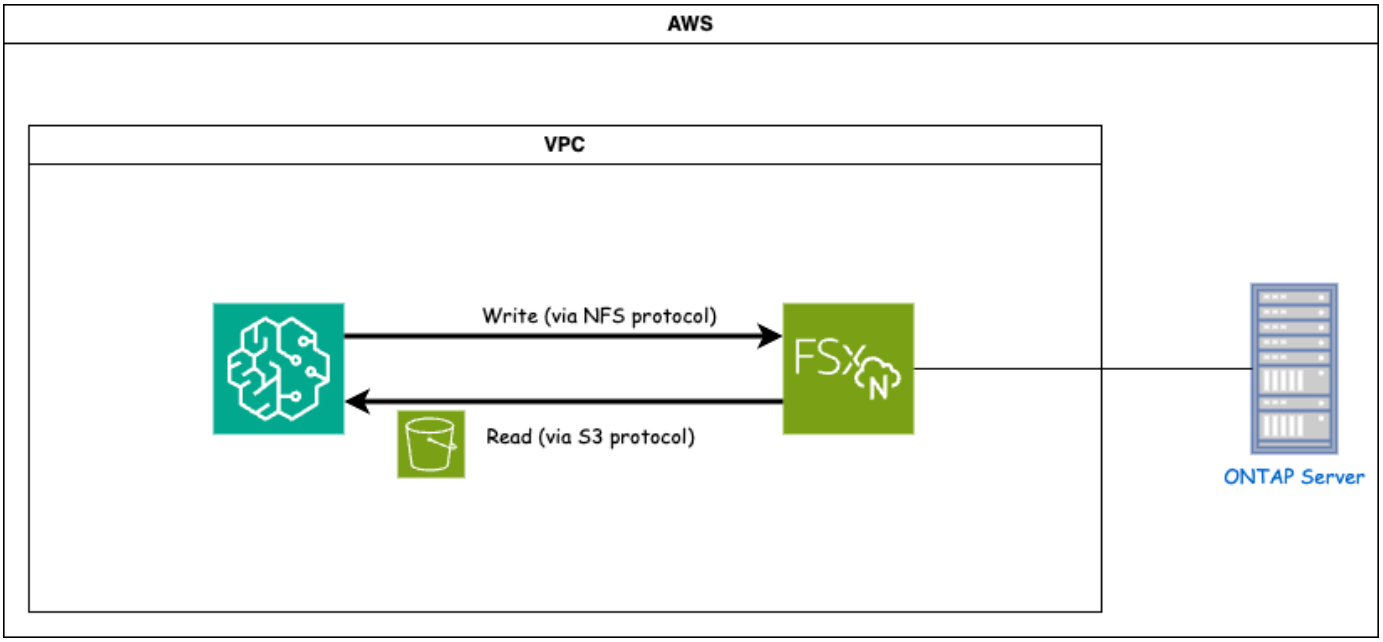
Este tutorial ofrece un ejemplo práctico de un proyecto de clasificación de visión por computadora, que proporciona experiencia práctica en la construcción de modelos ML que utilizan FSxN como fuente de datos dentro del entorno de SageMaker. El proyecto se centra en el uso de PyTorch, un marco de aprendizaje 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 FSxN como fuente de datos en Amazon SageMaker.

Qué es FSxN

Amazon FSx para NetApp 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 para NetApp ONTAP es escalable, lo que te permite ajustarla fácilmente según tus necesidades.

Requisito previo

Entorno de red



FSxN (Amazon FSx para NetApp 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

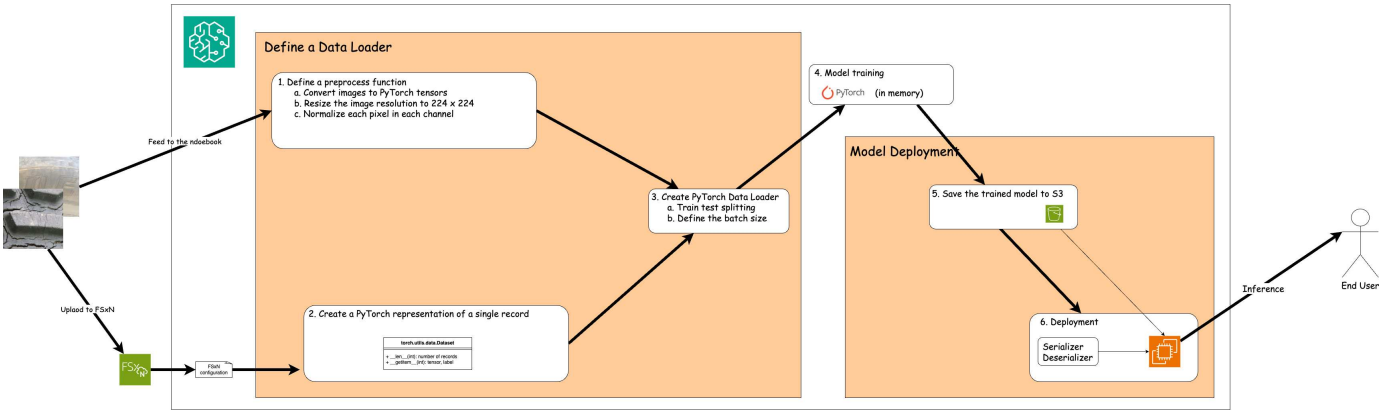
(SVM) que se conecta a él. En el diagrama proporcionado, el servidor NetApp ONTAP gestionado por AWS se encuentra fuera del VPC. El SVM sirve como intermediario entre SageMaker y el sistema NetApp ONTAP, al recibir solicitudes de operaciones de SageMaker y reenviarlas al almacenamiento subyacente. Para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

Acceso a los datos

En escenarios del mundo real, los científicos de datos suelen utilizar los datos existentes almacenados en FSxN para crear sus modelos de aprendizaje automático. Sin embargo, a efectos de demostración, dado que el sistema de archivos FSxN está inicialmente vacío después de la creación, es necesario cargar manualmente los datos de entrenamiento. Esto se puede lograr mediante el montaje de FSxN como un volumen a 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 le permite aprovechar la capacidad de almacenamiento y las capacidades de FSxN mientras trabaja con SageMaker para el desarrollo y entrenamiento de modelos.

El proceso de lectura de datos implica la configuración de FSxN como un bucket S3 privado. Para obtener más información sobre las instrucciones de configuración detalladas, consulte "1 parte: Integración de AWS FSx para NetApp ONTAP (FSxN) como bloque de S3 privado en AWS SageMaker"

Visión General de la Integración



El flujo de trabajo del uso de datos de entrenamiento en FSxN para construir un modelo de aprendizaje profundo en SageMaker se puede resumir en tres pasos principales: Definición de cargador de datos, entrenamiento de modelos e implementación. 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 eficaz para construir e implementar modelos de aprendizaje profundo utilizando datos de entrenamiento de FSxN 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 FSxN. El estilo de código para acceder a los datos de FSxN es similar a la lectura de datos de Amazon S3. La explicación subsiguiente profundiza en el proceso de creación del objeto privado S3 **bucket**.

FSxN 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 FSxN 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 FSxN en SageMaker, se crea un manejador que apunta al almacenamiento FSxN mediante el protocolo S3. Esto permite que FSxN se trate como un cubo privado de S3. La configuración del manejador incluye especificar la dirección IP de la SVM FSxN, el nombre del depósito 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 AWS FSx para NetApp ONTAP (FSxN) 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

```python
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.split(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.

```python
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)

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

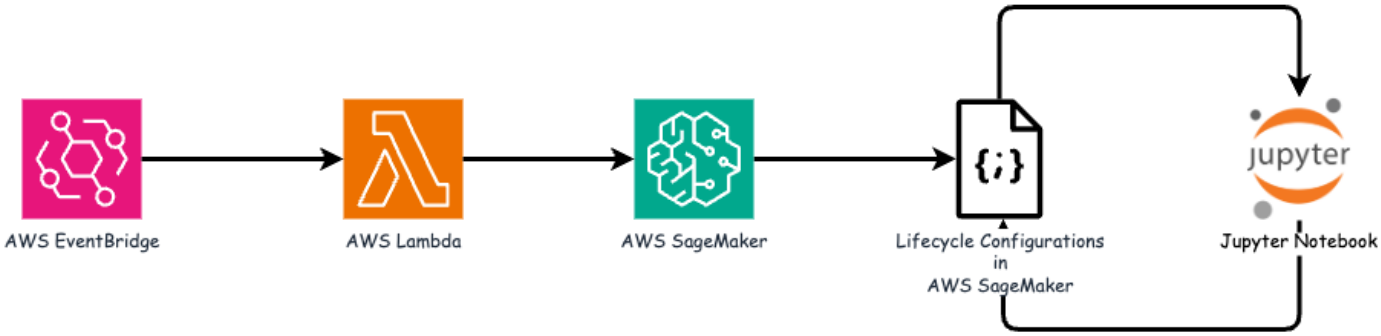
Funcionalidad	Nombre	Comentar
Almacenamiento de datos	FSxN de AWS	Consulte "1 parte: Integración de AWS FSx para NetApp ONTAP (FSxN) como bloque de S3 privado en AWS SageMaker".
IDE de ciencia de datos	SageMaker de AWS	Este tutorial se basa en el cuaderno Jupyter que se presenta en la "Parte 2: Aprovechamiento de AWS FSx para NetApp ONTAP (FSxN) como fuente de datos para el entrenamiento de modelos en SageMaker".
Función para activar el pipeline de MLOps	Función AWS Lambda	-

Funcionalidad	Nombre	Comentar
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 FSxN preconfigurado. Este tutorial utiliza los datos almacenados en FSxN para el proceso de entrenamiento.
- Una instancia **SageMaker Notebook** que está configurada para compartir la misma VPC que el sistema de archivos FSxN 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



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

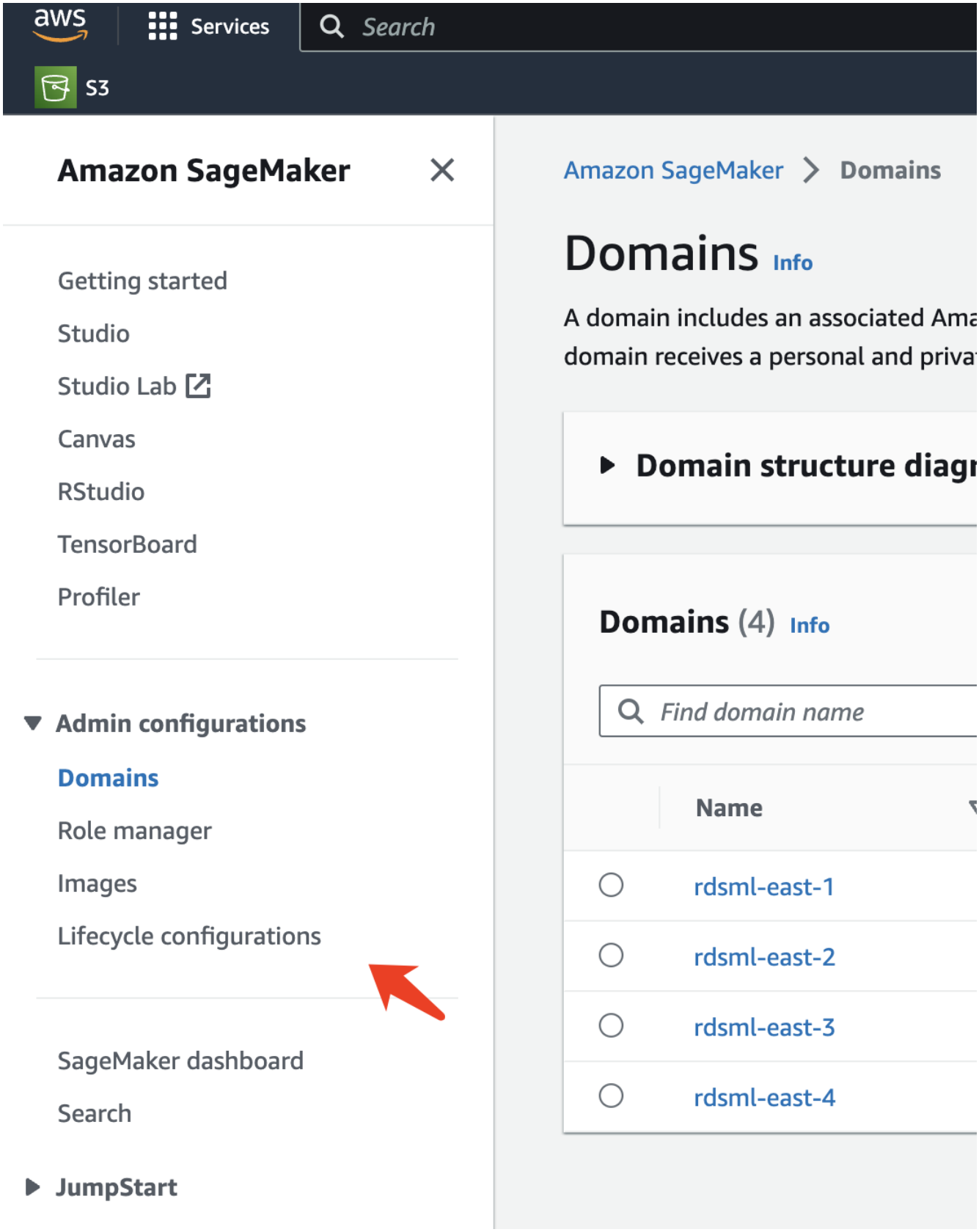
Configuraciones de ciclo de vida

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.

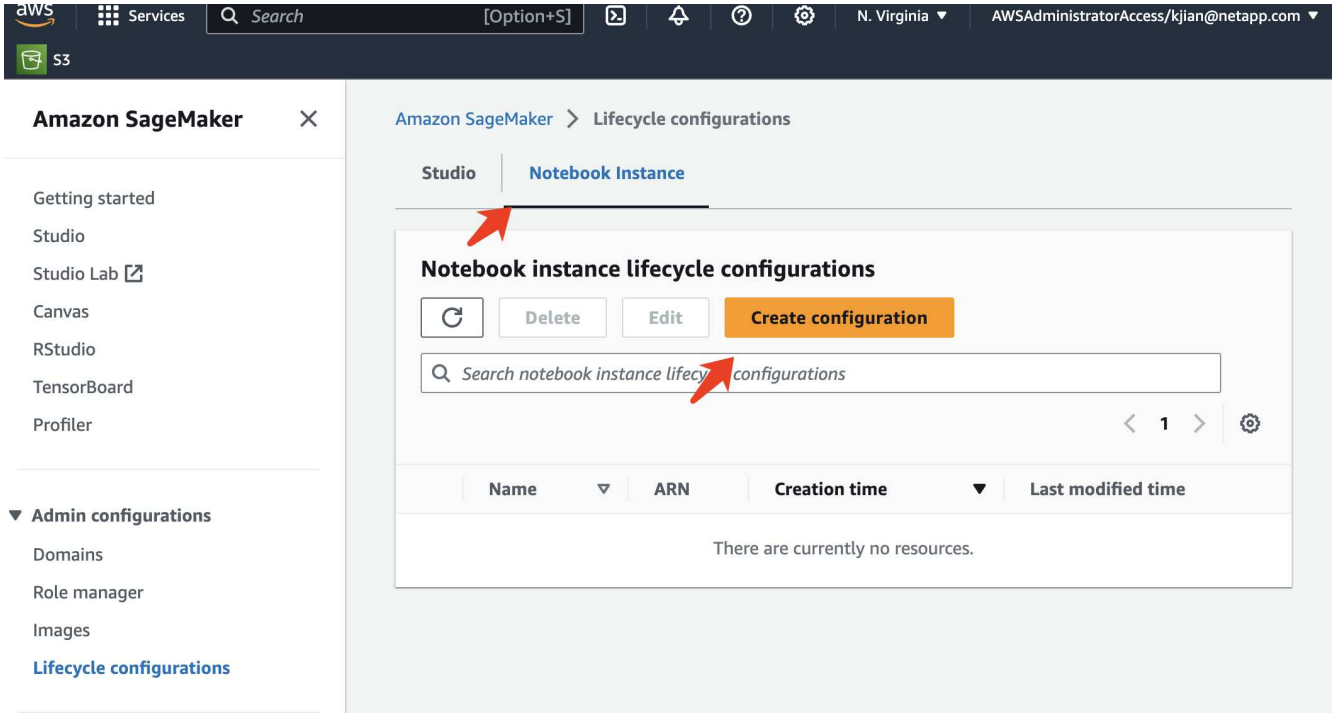
Es importante tener en cuenta que la configuración de **configuraciones de ciclo de vida** debe configurarse con antelación. Por lo tanto, se recomienda priorizar la configuración de este aspecto antes de continuar con

la otra configuración de pipeline de MLOps.

1. 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**.



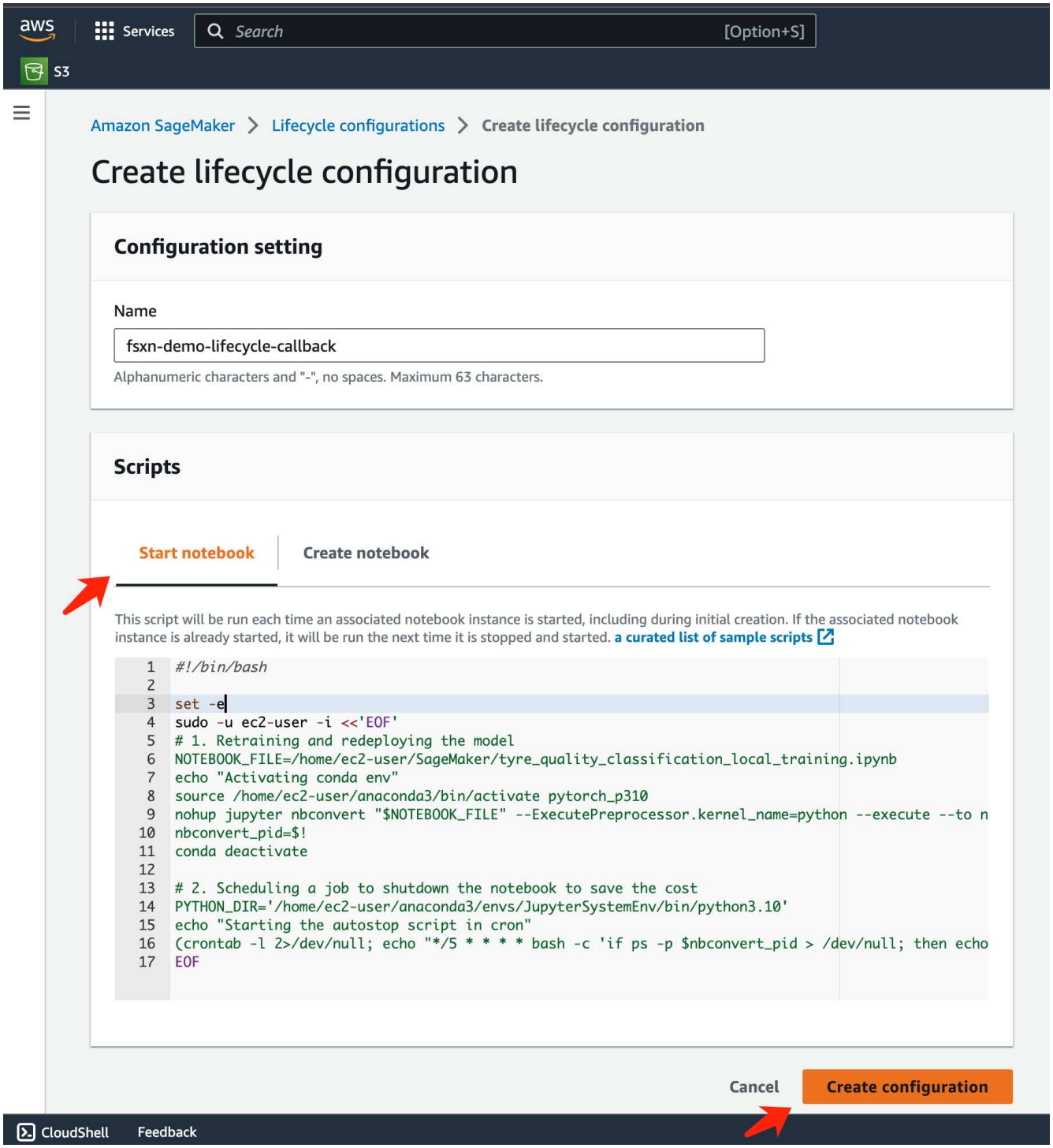
2. Seleccione la pestaña **Instancia de bloc de notas** y haga clic en el botón **Crear configuración**



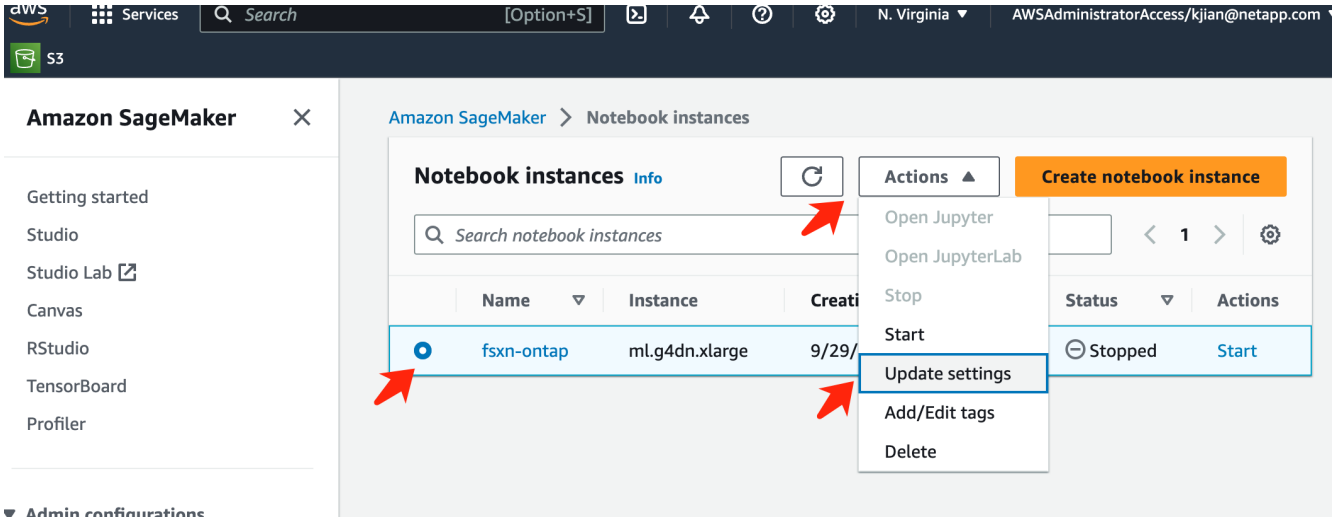
3. Pegue el siguiente código en el área de entrada.

```
#!/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
```

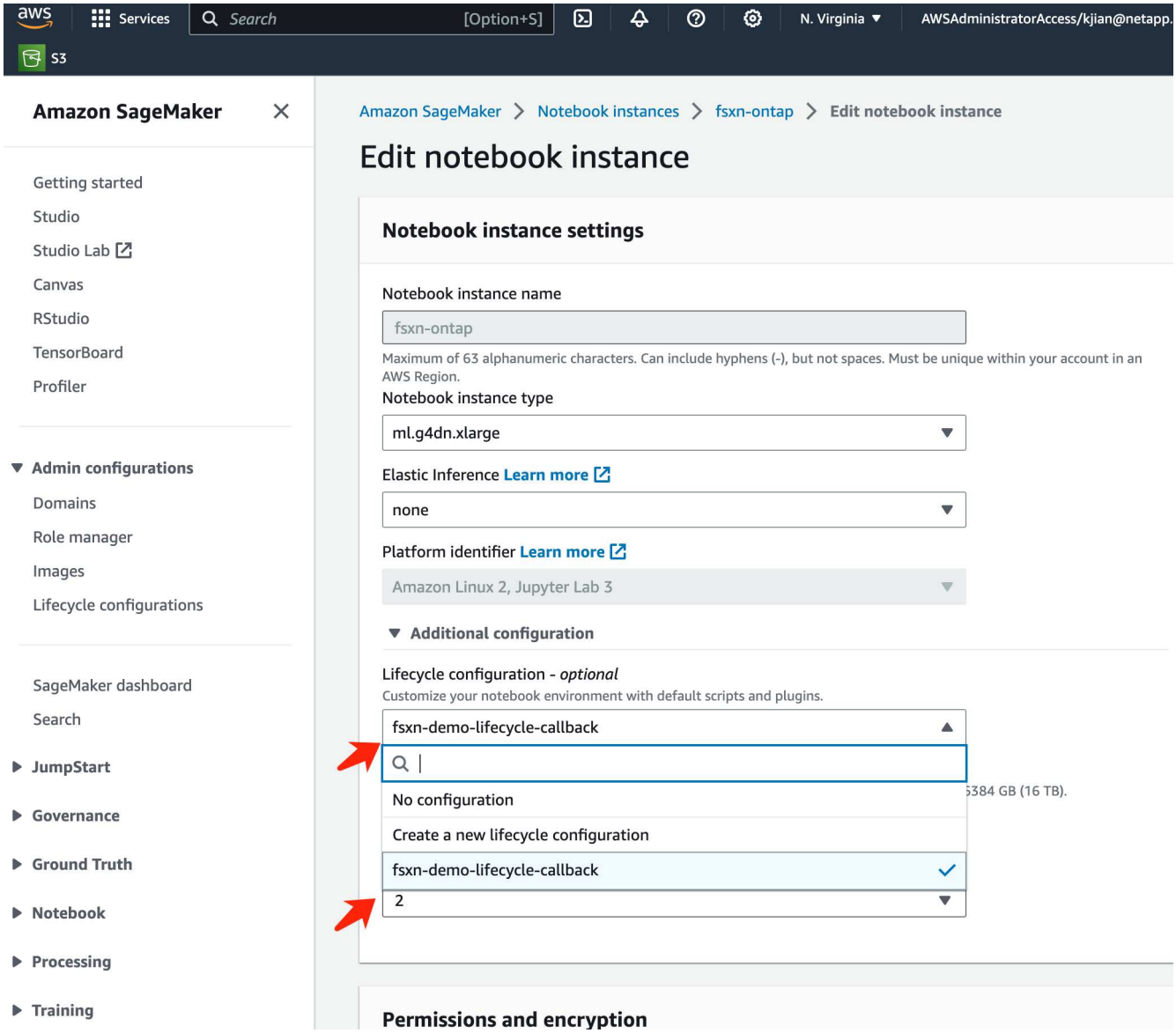
4. Este script ejecuta el Jupyter Notebook, que se encarga del reciclaje y el redespliegue del modelo para la inferencia. Una vez finalizada la ejecución, el bloc de notas se apagará automáticamente en 5 minutos. Para obtener más información sobre la declaración del problema y la implementación del código, consulte "Parte 2: Aprovechamiento de AWS FSx para NetApp ONTAP (FSxN) como fuente de datos para el entrenamiento de modelos en SageMaker".



5. 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.



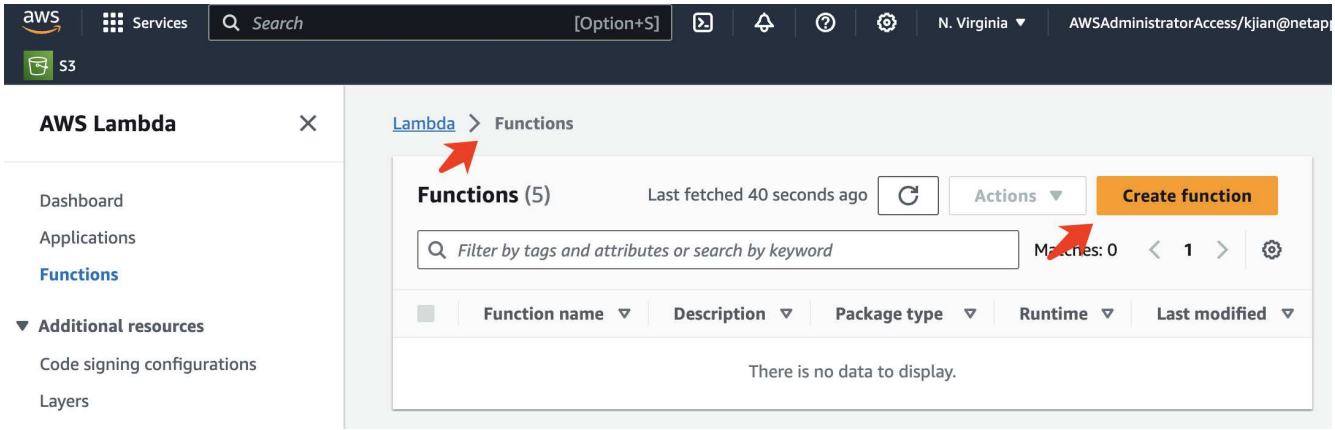
6. Seleccione la **Configuración de ciclo de vida** creada y haga clic en **Actualizar instancia de bloc de notas**.



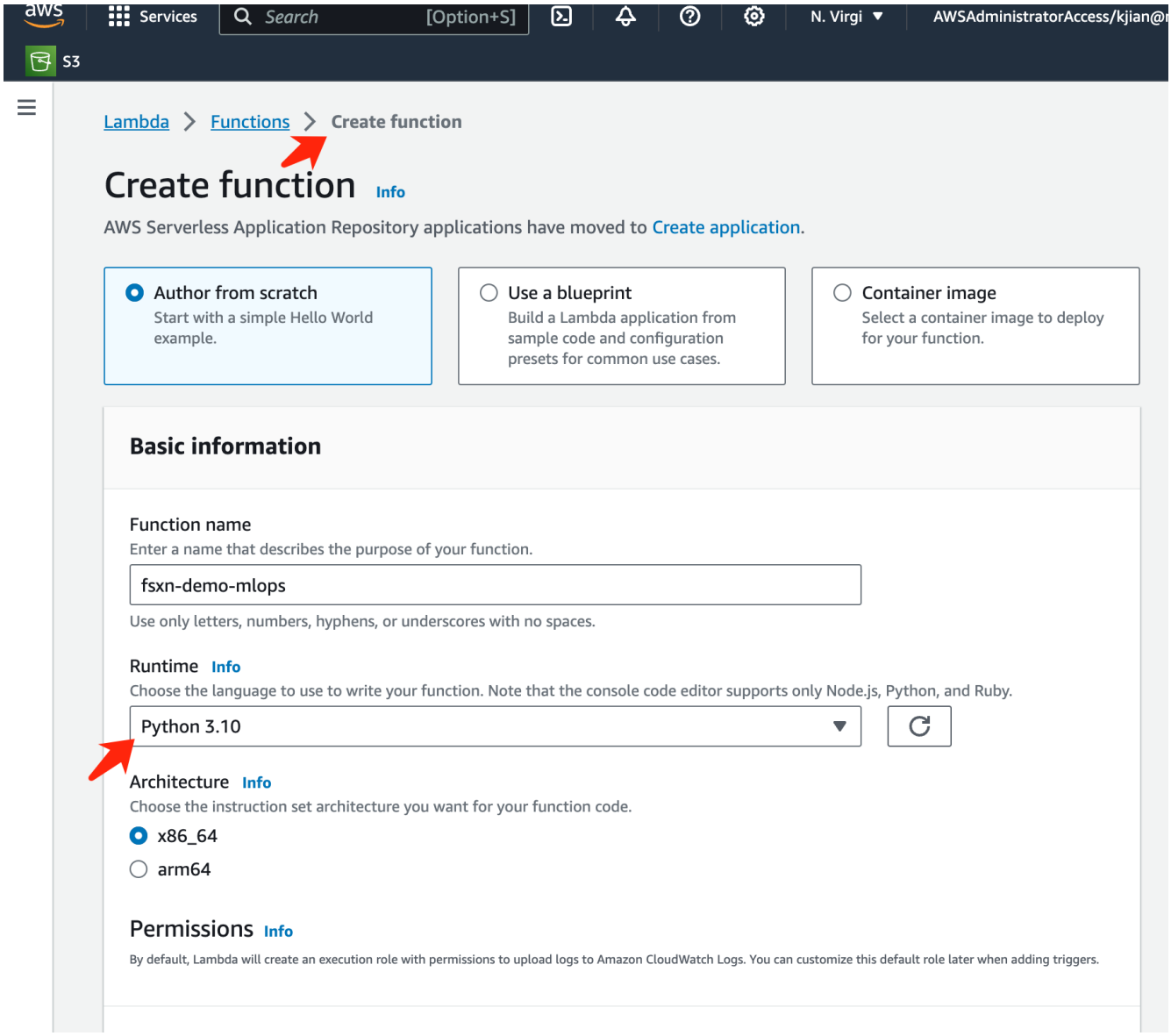
Función sin servidor de AWS Lambda

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

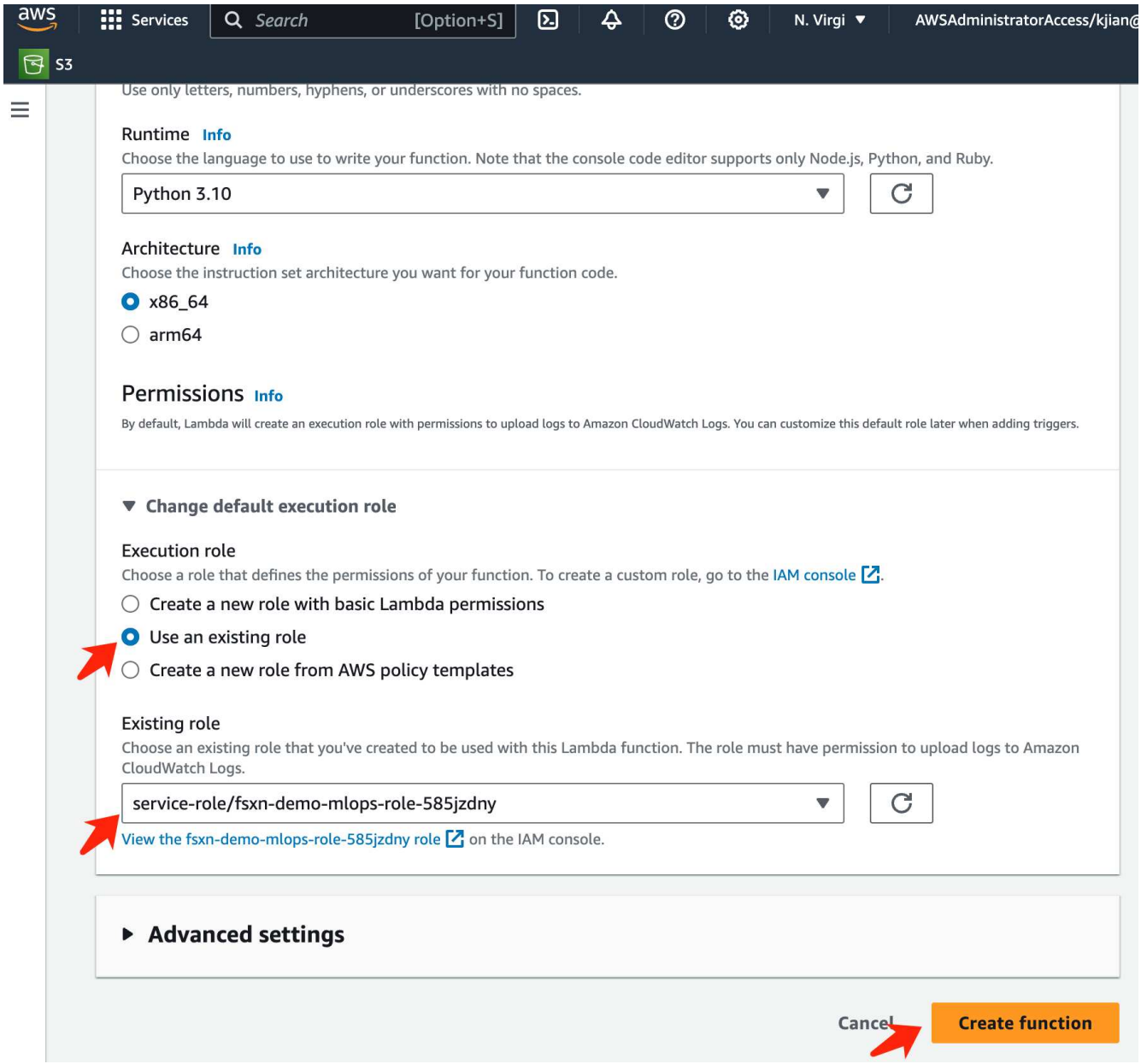
1. Para crear una función **AWS Lambda**, navegue hasta el panel correspondiente, cambie a la pestaña **Funciones** y haga clic en **Crear función**.



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



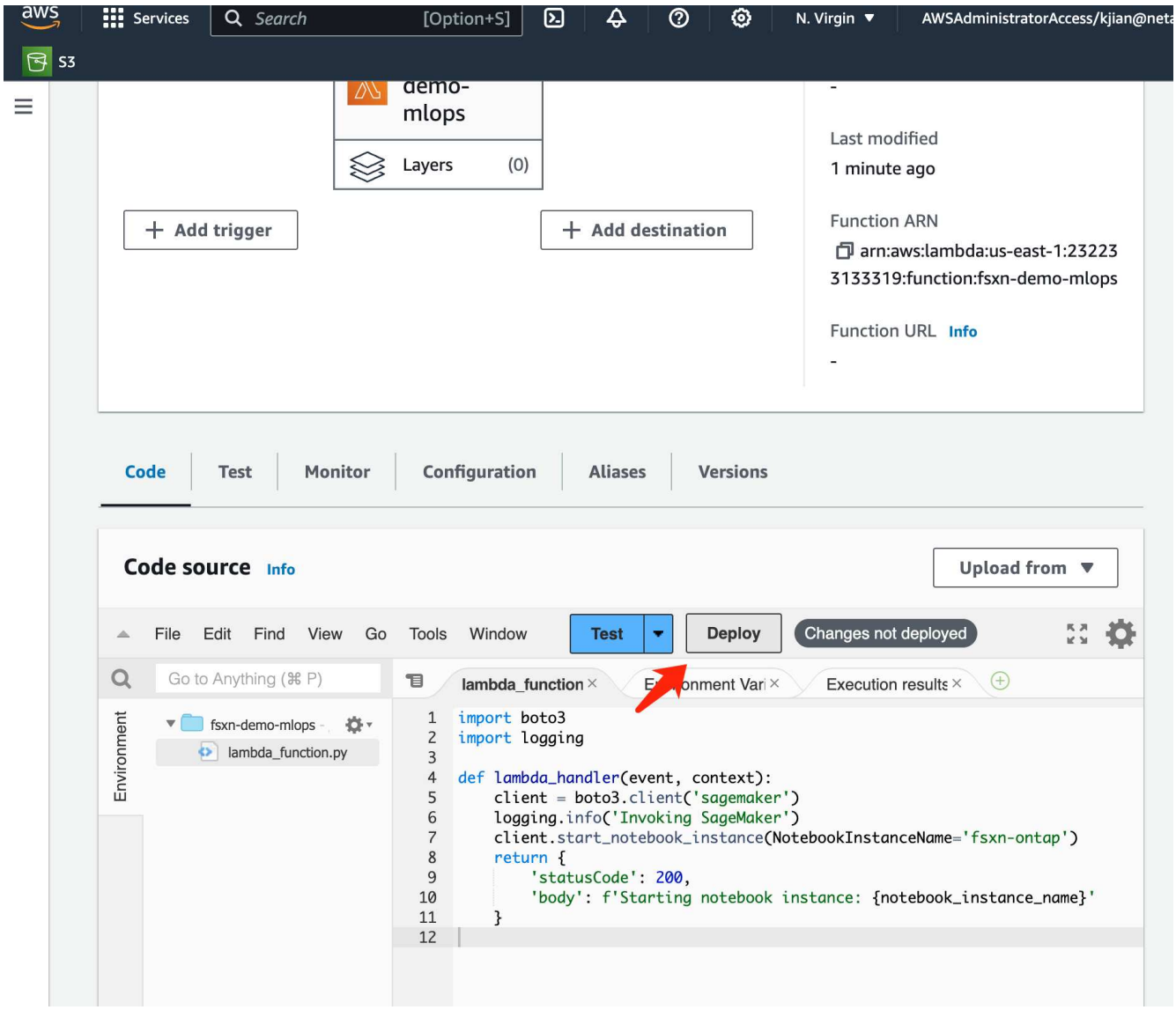
3. Verifique que el rol designado tiene el permiso requerido **AmazonSageMakerFullAccess** y haga clic en el botón **Crear función**.



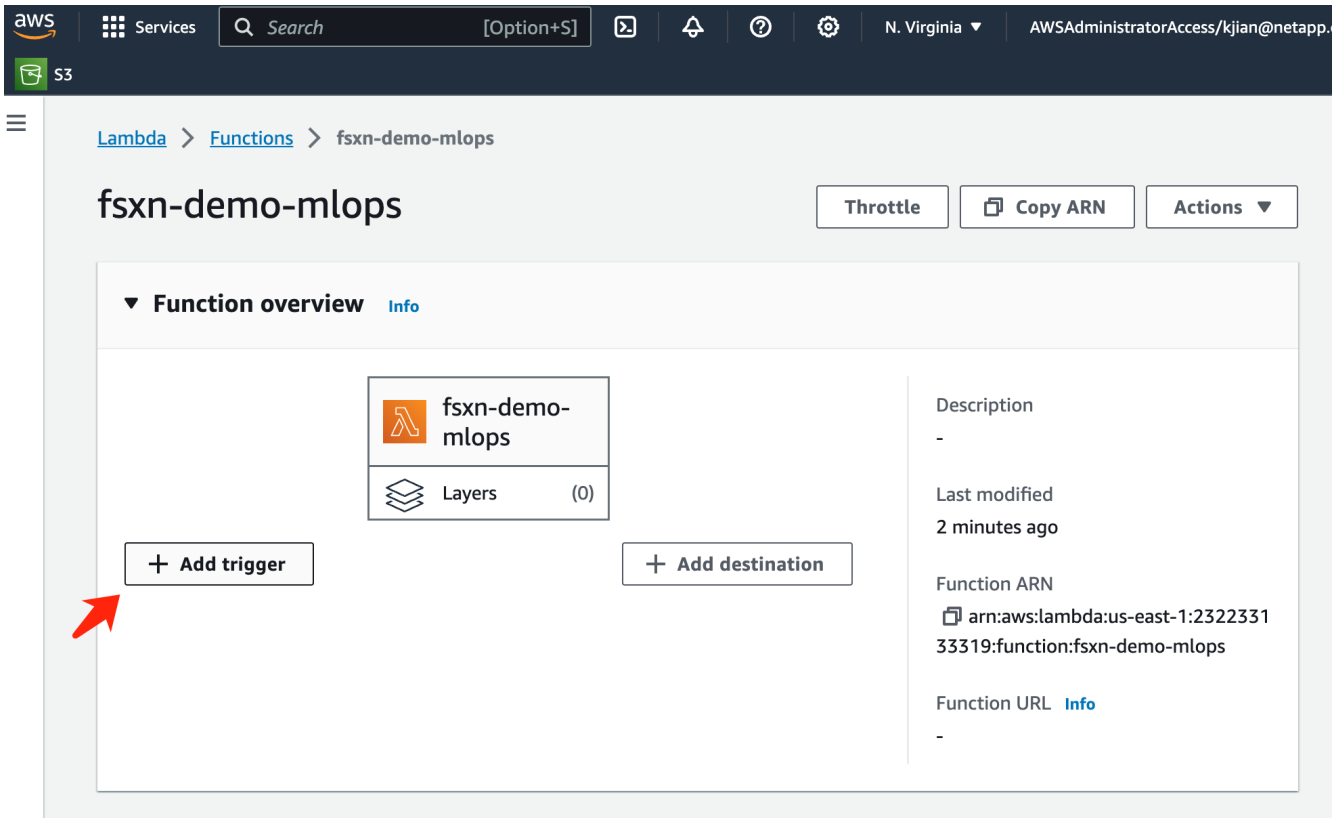
4. Seleccione la función Lambda creada. En la pestaña de código, copie y pegue el siguiente código en el área de texto. Este código inicia la instancia de notebook llamada **fsxn-ontap**.

```
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}'
    }
```

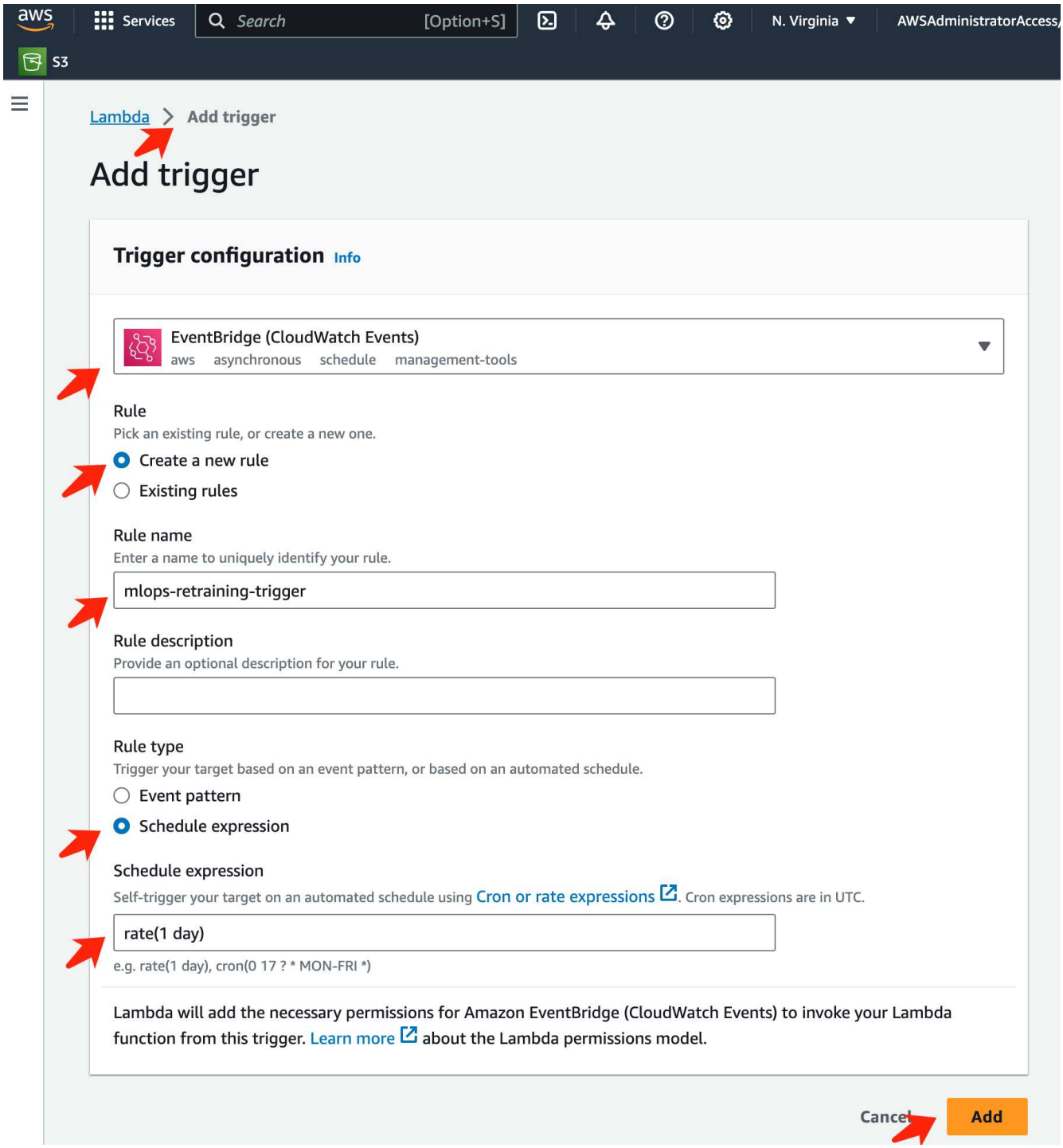
5. Haga clic en el botón **Desplegar** para aplicar este cambio de código.



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



7. 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.



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 **FSxN**, 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.

MLOps de multinube híbrida con Domino Data Lab y NetApp

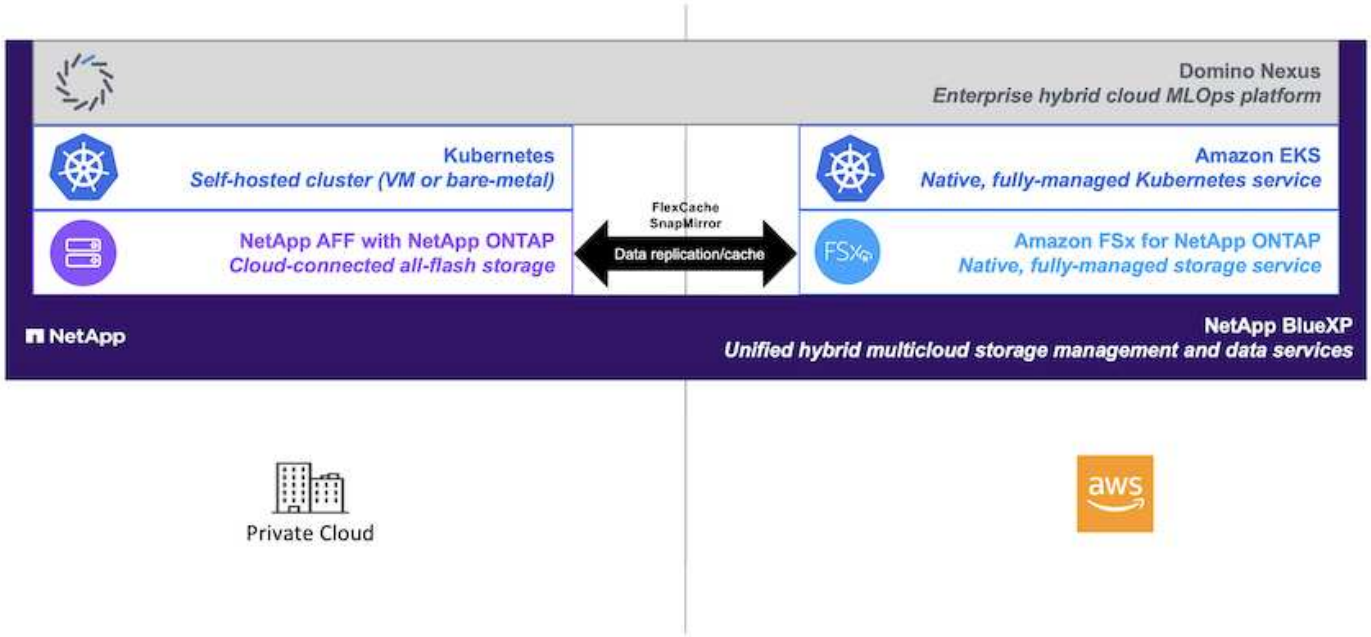
MLOps de multinube híbrida con Domino Data Lab y NetApp

Mike Oglesby, NetApp

Organizaciones de todo el mundo están adoptando actualmente la IA para transformar sus empresas y sus procesos. Por ello, la infraestructura de computación preparada para la IA suele escasear. Las empresas están adoptando arquitecturas MLOps de multinube híbrida para aprovechar los entornos informáticos disponibles en diferentes regiones, centros de datos y nubes, logrando así un equilibrio entre costes, disponibilidad y rendimiento.

Domino Nexus, de Domino Data Lab, es un plano de control de MLOps unificado que te permite ejecutar cargas de trabajo de ciencia de datos y aprendizaje automático en cualquier clúster de computación, en cualquier nube, región o on-premises. Unifica los silos de ciencia de datos en toda la empresa, de modo que dispone de un lugar para crear, poner en marcha y supervisar modelos. Del mismo modo, las funcionalidades de gestión de datos en cloud híbrido de NetApp le permiten llevar sus datos a sus trabajos y espacios de trabajo, independientemente de dónde se ejecuten. Al emparejar Domino Nexus con NetApp, dispone de la flexibilidad necesaria para programar cargas de trabajo en diferentes entornos sin tener que preocuparse por la disponibilidad de los datos. En otras palabras, tiene la capacidad de enviar sus cargas de trabajo y sus datos al entorno de computación apropiado, lo que le permite acelerar sus puestas en marcha de IA a la vez que se rigen por las normativas en torno a la privacidad y soberanía de los datos.

Esta solución demuestra la puesta en marcha de un plano de control unificado de MLOps que incorpora un clúster de Kubernetes en las instalaciones y un clúster de Elastic Kubernetes Service (EKS) que se ejecuta en Amazon Web Services (AWS).



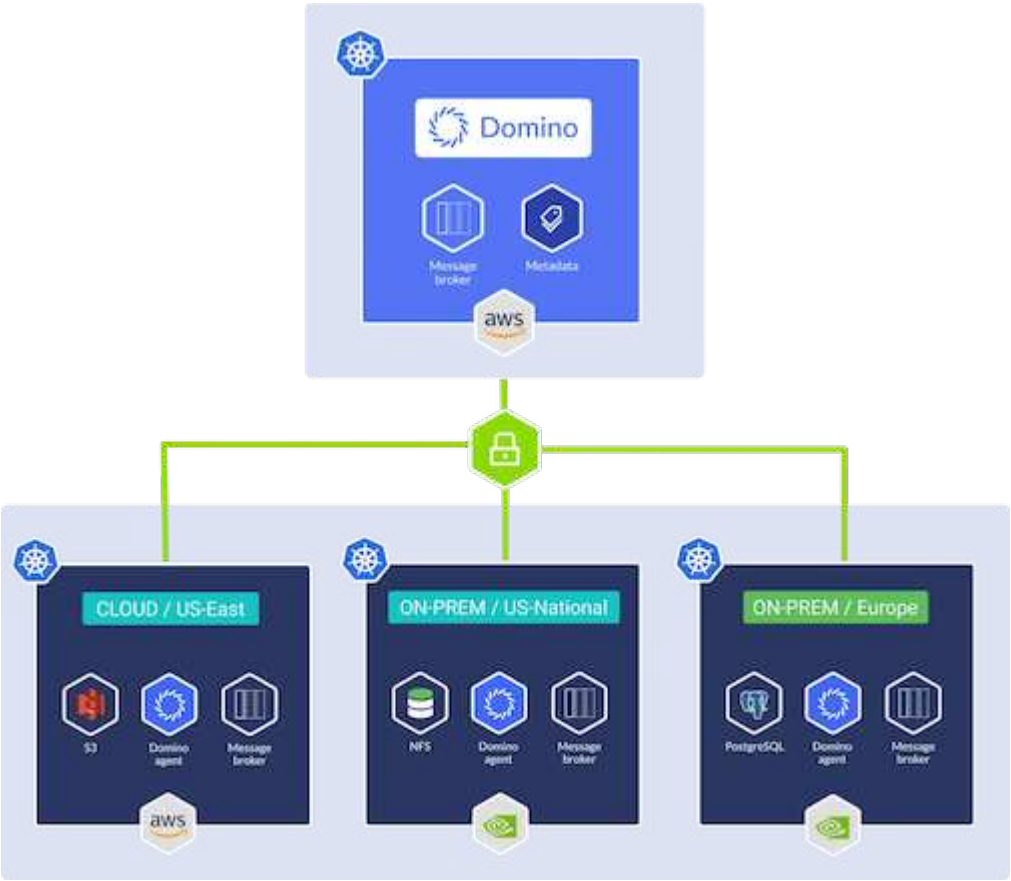
Visión general de la tecnología

Domino Data Lab

Domino Data Lab impulsa empresas basadas en modelos con su plataforma líder de IA empresarial de confianza más del 20 % de las empresas Fortune 100. Domino acelera el desarrollo y la puesta en marcha del trabajo de ciencia de datos, a la vez que aumenta la colaboración y la gestión. Con Domino, las empresas de todo el mundo pueden desarrollar mejores medicamentos, cultivar cultivos más productivos, construir mejores coches y mucho más. Fundada en 2013, Domino cuenta con el respaldo de Coatue Management, Great Hill Partners, Highland Capital, Sequoia Capital y otros inversores líderes.

Domino permite a las empresas y a sus científicos de datos crear, implementar y administrar IA en una plataforma unificada de extremo a extremo, de forma rápida, responsable y rentable. Los equipos pueden acceder a todos los datos, herramientas, computación, modelos y proyectos que necesitan en cualquier entorno, para que puedan colaborar, reutilizar el trabajo pasado, seguir los modelos en producción para mejorar la precisión, estandarizar con las mejores prácticas y hacer que la IA sea responsable y gobernada.

- **Abierto y flexible:** Acceda al ecosistema más amplio de herramientas comerciales y de código abierto, e infraestructura, para las mejores innovaciones y sin bloqueo de proveedores.
- **Sistema de registro:** Centro central de operaciones de IA y conocimiento en toda la empresa, lo que permite mejores prácticas, colaboración interfuncional, innovación más rápida y eficiencia.
- **Integrado:** Los flujos de trabajo integrados y la automatización — creados para procesos empresariales, controles y gobierno — satisfacen sus necesidades de cumplimiento y reglamentación.
- **Multinube híbrida:** Ejecuta cargas de trabajo de IA cerca de tus datos en cualquier lugar (on-premises, híbridas, en cualquier nube o en múltiples nubes) para un menor costo, un rendimiento óptimo y cumplimiento normativo.



Nexus de Domino

Domino Nexus es un único panel que le permite ejecutar cargas de trabajo de ciencia de datos y aprendizaje automático en cualquier clúster de computación, en cualquier nube, región o on-premises. Unifica los silos de ciencia de datos en toda la empresa, de modo que dispone de un lugar para crear, poner en marcha y supervisar modelos.

BlueXP de NetApp

NetApp BlueXP unifica todos los servicios de datos y almacenamiento de NetApp en una única herramienta que te permite crear, proteger y gobernar tus activos de datos multicloud híbridos. Proporciona una experiencia unificada para el almacenamiento y los servicios de datos en entornos en las instalaciones y en el cloud, y permite la sencillez operativa a través de la potencia de AIOps, con los parámetros de consumo flexible y la protección integrada que necesita el mundo actual dirigido por el cloud.

ONTAP de NetApp

ONTAP 9, la última generación del software de gestión del almacenamiento de NetApp, permite a las empresas modernizar su infraestructura y realizar la transición a un centro de datos preparado para el cloud. ONTAP ofrece las mejores capacidades de gestión de datos y permite la gestión y protección de los datos con un solo conjunto de herramientas, sin importar dónde residan. También puede mover los datos libremente a donde sea necesario: El perímetro, el núcleo o el cloud. ONTAP 9 incluye numerosas funciones que simplifican la gestión de datos, aceleran y protegen los datos esenciales y permiten disfrutar de funcionalidades de infraestructura de nueva generación en arquitecturas de cloud híbrido.

Simplificar la gestión de los datos

La gestión de los datos es crucial para las operaciones TECNOLÓGICAS empresariales y los científicos de datos, para que se utilicen recursos apropiados para las aplicaciones de IA y para entrenar conjuntos de datos de IA/ML. La siguiente información adicional sobre las tecnologías de NetApp no está disponible para esta validación, pero puede ser relevante en función de su puesta en marcha.

El software para la gestión de datos ONTAP incluye las siguientes funciones para mejorar y simplificar las operaciones, y reducir el coste total de funcionamiento:

- Compactación de datos inline y deduplicación expandida. La compactación de datos reduce el espacio perdido dentro de los bloques de almacenamiento, mientras que la deduplicación aumenta la capacidad efectiva de forma significativa. Esto es aplicable a los datos almacenados localmente y a los datos organizados en niveles en el cloud.
- Calidad de servicio (AQoS) mínima, máxima y adaptativa. Los controles granulares de calidad de servicio (QoS) ayudan a mantener los niveles de rendimiento para aplicaciones críticas en entornos altamente compartidos.
- FabricPool de NetApp. Proporciona la organización automática en niveles de datos fríos en opciones de almacenamiento en cloud privado como Amazon Web Services (AWS), Azure y la solución de almacenamiento StorageGRID de NetApp. Para obtener más información sobre FabricPool, consulte "TR-4598: Prácticas recomendadas de FabricPool".

Acelere y proteja sus datos

ONTAP no solo ofrece niveles de rendimiento y protección de datos superiores, sino que amplía estas capacidades de las siguientes maneras:

- Rendimiento y menor latencia. ONTAP ofrece la salida más alta posible con la menor latencia posible.

- Protección de datos. ONTAP ofrece capacidades integradas de protección de datos, con una administración común entre todas las plataformas.
- Cifrado de volúmenes de NetApp (NVE). ONTAP ofrece cifrado nativo en el nivel de volumen y permite la gestión de claves incorporada o externa.
- Multi-tenancy y autenticación multifactor. ONTAP permite compartir recursos de infraestructura con los niveles más altos de seguridad.

Infraestructura preparada para futuros retos

ONTAP ayuda a satisfacer las exigentes y siempre cambiantes necesidades de su empresa con las siguientes funciones:

- Escalado sencillo y operaciones no disruptivas. ONTAP admite la adición no disruptiva de capacidad a las controladoras existentes y a clústeres de escalado horizontal. Los clientes pueden empezar a utilizar tecnologías punteras como NVMe y FC 32 GB, sin necesidad de realizar costosas migraciones de datos y sin cortes.
- Conexión de cloud. ONTAP es el software de gestión de almacenamiento con mejor conexión de cloud e incluye opciones de almacenamiento definido por software e instancias nativas del cloud en todos los clouds públicos.
- Integración con aplicaciones emergentes. ONTAP ofrece servicios de datos de clase empresarial para plataformas y aplicaciones de última generación, como vehículos autónomos, ciudades inteligentes e Industria 4.0, utilizando la misma infraestructura que da soporte a las aplicaciones empresariales existentes.

Amazon FSX para ONTAP de NetApp

Amazon FSx para NetApp ONTAP es un servicio AWS de primer proveedor totalmente gestionado que proporciona un almacenamiento de archivos altamente fiable, escalable, de alto rendimiento y con muchas funciones integrado en el popular sistema de archivos ONTAP de NetApp. FSX para ONTAP combina las funciones, el rendimiento, las funcionalidades y las operaciones API de los sistemas de archivos de NetApp con la agilidad, la escalabilidad y la simplicidad de un servicio AWS totalmente gestionado.

Astra Trident de NetApp

Astra Trident permite el consumo y la gestión de recursos de almacenamiento en todas las plataformas de almacenamiento de NetApp más conocidas, tanto en el cloud público como en las instalaciones, incluidos ONTAP (AFF, FAS, Select, Cloud, Amazon FSx para NetApp ONTAP), el software Element (NetApp HCI, SolidFire), el servicio Azure NetApp Files y Cloud Volumes Service en Google Cloud. Astra Trident es una interfaz de almacenamiento de contenedores (CSI) que ordena el almacenamiento dinámico conforme a la normativa que se integra de forma nativa con Kubernetes.

Kubernetes

Kubernetes es una plataforma de orquestación de contenedores distribuida de código abierto que originalmente diseñada por Google y que ahora se mantiene mediante Cloud Native Computing Foundation (CNCF). Kubernetes permite la automatización de funciones de puesta en marcha, gestión y escalado para aplicaciones en contenedores, y es la plataforma de orquestación de contenedores dominante en entornos empresariales.

Amazon Elastic Kubernetes Service (EKS)

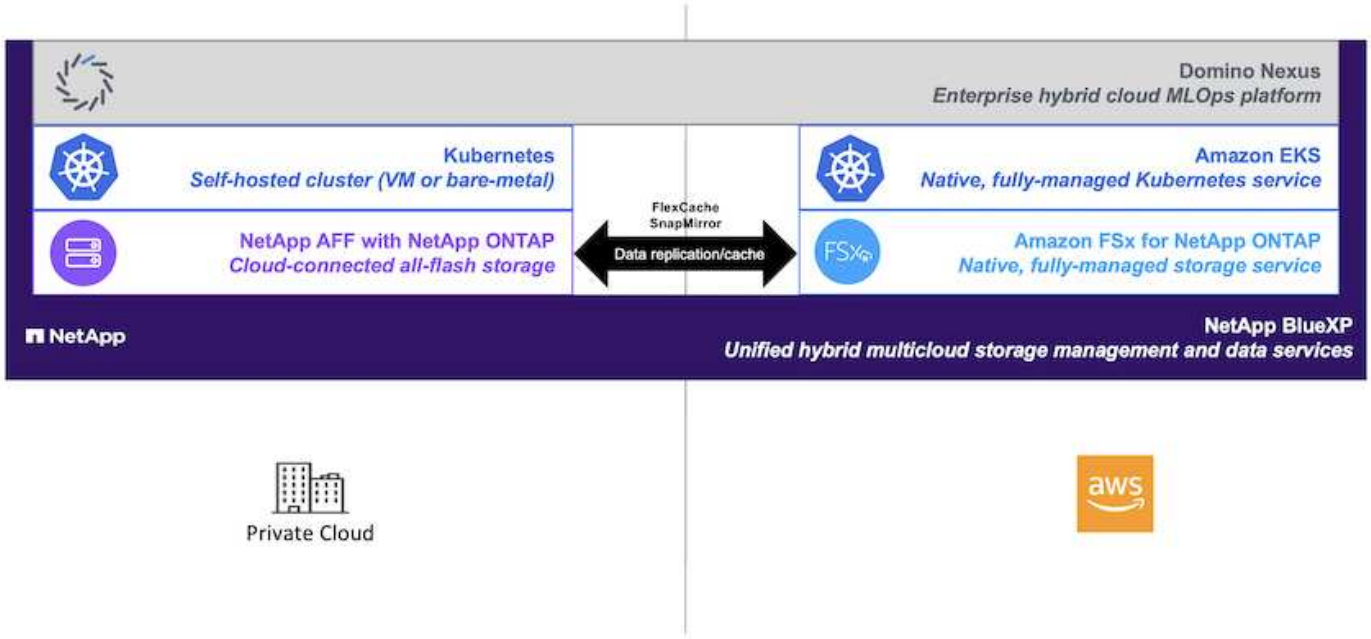
Amazon Elastic Kubernetes Service (Amazon EKS) es un servicio de Kubernetes gestionado en la nube de AWS. Amazon EKS gestiona automáticamente la disponibilidad y escalabilidad de los nodos del plano de

control de Kubernetes responsables de programar contenedores, gestionar la disponibilidad de las aplicaciones, almacenar datos del clúster y otras tareas clave. Con Amazon EKS, puede aprovechar todo el rendimiento, la escalabilidad, la fiabilidad y la disponibilidad de la infraestructura de AWS, así como las integraciones con las redes y los servicios de seguridad de AWS.

Arquitectura

Esta solución combina las capacidades de programación de cargas de trabajo de la multinube híbrida de Domino Nexus con los servicios de datos de NetApp para crear una plataforma MLOps unificada de la nube híbrida. Consulte la siguiente tabla para obtener más información.

Componente	Nombre	Entorno Oracle
Plano de control de MLOps	"Domino Enterprise AI Platform con Domino Nexus"	AWS
Entornos informáticos de la plataforma MLOps	"Planos de datos Domino Nexus"	AWS, centro de datos en las instalaciones
Plataforma de informática en las instalaciones	"Kubernetes" con "Astra Trident de NetApp"	Centro de datos en las instalaciones
Plataforma de tecnología cloud	"Amazon Elastic Kubernetes Service (EKS)" con "Astra Trident de NetApp"	AWS
Plataforma de datos en las instalaciones	"Dispositivo de almacenamiento de NetApp" alimentado por "ONTAP de NetApp"	Centro de datos en las instalaciones
Plataforma de datos cloud	"Amazon FSX para ONTAP de NetApp"	AWS



Configuración inicial

En esta sección se describen las tareas de configuración iniciales que se deben realizar para utilizar Domino Nexus con servicios de datos de NetApp en un entorno híbrido que incorpora un centro de datos local y AWS.

Requisitos previos

Antes de realizar los pasos que se describen en esta sección, asumimos que ya ha realizado las siguientes tareas:

- Ya ha puesto en marcha y configurado su plataforma de almacenamiento NetApp ONTAP en las instalaciones. Para obtener más información, consulte la "Documentación de productos de NetApp".
- Ya ha aprovisionado una instancia de Amazon FSx para NetApp ONTAP en AWS. Para obtener más información, consulte la "Página de producto de Amazon FSx para NetApp ONTAP".
- Ya ha aprovisionado un clúster de Kubernetes en su centro de datos en las instalaciones. Para obtener más información, consulte la "Guía del administrador de Domino".
- Ya ha aprovisionado un clúster de Amazon EKS en AWS. Para obtener más información, consulte la "Guía del administrador de Domino".
- Ha instalado Astra Trident de NetApp en su clúster de Kubernetes en las instalaciones. Además, ha configurado esta instancia de Trident para utilizar su plataforma de almacenamiento de NetApp ONTAP local al aprovisionar y gestionar recursos de almacenamiento. Para obtener más información, consulte la "Documentación de NetApp Astra Trident".
- Ha instalado Astra Trident de NetApp en su clúster de Amazon EKS. Además, ha configurado esta instancia de Trident de modo que utilice su instancia de Amazon FSx para NetApp ONTAP al aprovisionar y gestionar recursos de almacenamiento. Para obtener más información, consulte la "Documentación de NetApp Astra Trident".
- Debe disponer de conectividad de red bidireccional entre su centro de datos en las instalaciones y la nube privada virtual (VPC) en AWS. Para obtener más información sobre las distintas opciones para implementarlo, consulte la "Documentación de Amazon Virtual Private Network (VPN)".

Instale Domino Enterprise AI Platform en AWS

Para instalar Domino Enterprise MLOps Platform en AWS, siga las instrucciones descritas en "Guía del administrador de Domino". Debe implementar Domino en el mismo clúster de Amazon EKS que había aprovisionado previamente. Además, Astra Trident de NetApp ya debe estar instalado y configurado en este clúster de EKS, y debe especificar una clase de almacenamiento gestionado por Trident como la clase de almacenamiento compartido de su archivo de configuración de instalación de domino.yml.

> Consulte la "Guía de referencia de configuración de instalación de Domino" para obtener detalles sobre cómo especificar una clase de almacenamiento compartido en el archivo de configuración de instalación de domino.yml.

> "Informe técnico TR-4952" Explica la puesta en marcha de Domino en AWS con Amazon FSx para NetApp ONTAP y puede ser una referencia útil para solucionar los problemas que surjan.

Habilite Domino Nexus

A continuación, debe habilitar Domino Nexus. Consulte la "Guía del administrador de Domino" para obtener más detalles.

Ponga en marcha un plano de datos de Domino en su centro de datos local

A continuación, debe poner en marcha un plano de datos de Domino en su centro de datos local. Es necesario poner en marcha este plano de datos en el clúster de Kubernetes local que ya haya aprovisionado. Además, Astra Trident de NetApp ya debe estar instalado y configurado en este clúster de Kubernetes. Consulte la "Guía del administrador de Domino" para obtener más detalles.

Expone los volúmenes NetApp existentes a Domino

Esta sección describe las tareas que se deben realizar para exponer los volúmenes NFS de NetApp ONTAP existentes a la plataforma MLOps de Domino. Estos mismos pasos se aplican tanto en las instalaciones como en AWS.

¿Por qué exponer volúmenes de NetApp ONTAP a Domino?

El uso de NetApp Volumes junto con Domino ofrece las siguientes ventajas:

- Puede ejecutar cargas de trabajo en conjuntos de datos extremadamente grandes aprovechando las funcionalidades de escalado horizontal de ONTAP de NetApp.
- Puede ejecutar cargas de trabajo en varios nodos de computación sin tener que copiar los datos en cada nodo.
- Puede aprovechar las funcionalidades de sincronización y movimiento de datos de multicloud híbrido de NetApp para acceder a sus datos en varios centros de datos o clouds.
- Quieres poder crear rápida y fácilmente una caché de tus datos en otro centro de datos o en otro cloud.

Muestre los volúmenes de NFS existentes que Astra Trident no aprovisionó

Si tu volumen de NFS de NetApp ONTAP existente no se ha aprovisionado en Astra Trident, sigue los pasos descritos en esta subsección.

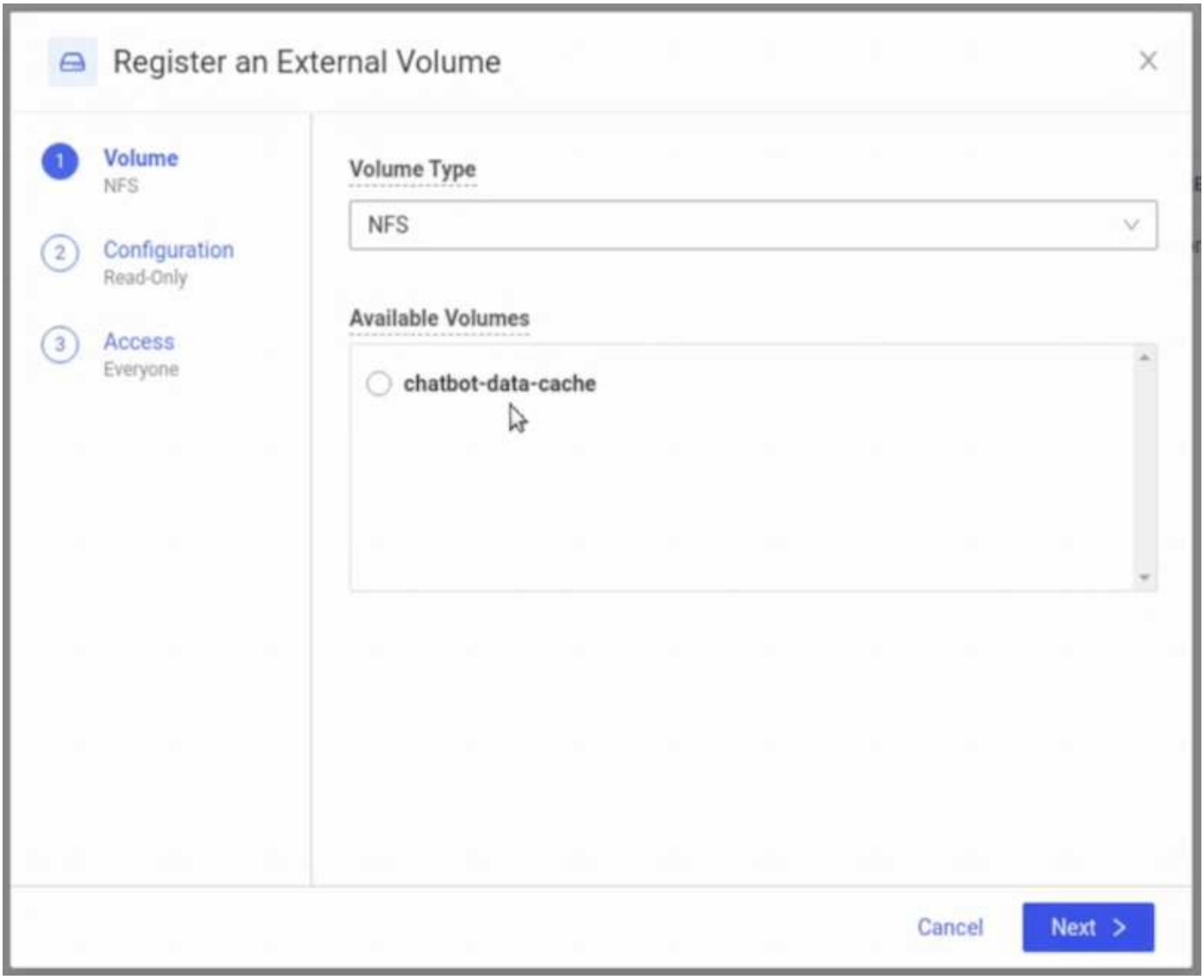
Cree un VP y una RVP en Kubernetes

> Para los volúmenes en las instalaciones, cree el VP y el PVC en su clúster de Kubernetes en las instalaciones. Para Amazon FSx para NetApp ONTAP Volumes, cree el VP y la RVP en Amazon EKS.

Primero, debe crear un volumen persistente (PV) y una reclamación de volumen persistente (PVC) en su clúster de Kubernetes. Para crear el VP y la RVP, utilice el "EJEMPLO DE PV/PVC DE NFS" De la guía de administración de Domino y actualice los valores para reflejar el entorno. Asegúrese de especificar los valores correctos para el `namespace`, `nfs.path`, y. `nfs.server` campos. Además, recomendamos indicar a sus nombres VP y RVP únicos que representen la naturaleza de los datos que se almacenan en el volumen NFS de ONTAP correspondiente. Por ejemplo, si el volumen contiene imágenes de defectos de fabricación, puede nombrar el VP, `pv-mfg-defect-images`, Y el PVC, `pvc-mfg-defect-images`.

Registrar volumen de datos externo en Domino

A continuación, debe registrar un volumen de datos externo en Domino. Para registrar un volumen de datos externo, consulte la "instrucciones" En la guía del administrador de Domino. Al registrar el volumen, asegúrese de seleccionar «NFS» en el menú desplegable «Tipo de volumen». Después de seleccionar «NFS», deberías ver tu RVP en la lista de volúmenes disponibles.



Exponer los volúmenes existentes que fueron aprovisionados por Astra Trident

Si el volumen existente se aprovisionó con Astra Trident, siga los pasos descritos en esta subsección.

Edite la RVP existente

Si tu volumen fue aprovisionado por Astra Trident, ya tienes una reclamación de volumen persistente (RVP) correspondiente al volumen. Para exponer este volumen a Domino, debe editar la PVC y agregar la siguiente etiqueta a la lista de etiquetas de la `metadata.labels` campo:

```
"dominodatalab.com/external-data-volume": "Generic"
```

Registrar volumen de datos externo en Domino

A continuación, debe registrar un volumen de datos externo en Domino. Para registrar un volumen de datos externo, consulte la "instrucciones" En la guía del administrador de Domino. Al registrar el volumen, asegúrese de seleccionar «Genérico» en el menú desplegable «Tipo de volumen». Después de seleccionar “Genérico”, deberías ver tu PVC en la lista de “Volúmenes disponibles”.

Acceda a los mismos datos en diferentes entornos

Esta sección describe las tareas que se deben realizar para acceder a los mismos datos en diferentes entornos informáticos. En la plataforma MLOps de Domino, los entornos informáticos se denominan «planos de datos». Siga las tareas descritas en esta sección si sus datos residen en un volumen de NetApp en un plano de datos, pero necesita acceder a ellos en otro plano de datos. Este tipo de escenario se suele denominar «explosión» o, cuando el entorno de destino es el cloud, «explosión de cloud». Esta funcionalidad a menudo es necesaria cuando se utilizan recursos informáticos limitados o sobresuscritos. Por ejemplo, si su clúster de computación en las instalaciones está sobresuscrito, puede que desee programar cargas de trabajo en el cloud donde pueden iniciarse inmediatamente.

Hay dos opciones recomendadas para acceder a un volumen de NetApp que reside en un plano de datos diferente. Estas opciones se describen en las subsecciones siguientes. Elija una de estas opciones en función de sus requisitos específicos. En la siguiente tabla, se describen los beneficios e inconvenientes de las dos opciones.

Opción	Beneficios	Inconvenientes
Opción 1: Caché	- Flujo de trabajo más simple - Capacidad de almacenar en caché un subconjunto de datos en función de las necesidades - Capacidad para escribir datos de nuevo en la fuente - No hay copia remota para administrar	- Aumento de la latencia en el acceso inicial a los datos a medida que la caché está hidratada.
Opción 2 - Espejo	- Copia completa del volumen fuente - Sin aumento de la latencia debido a la hidratación de la caché (después de la operación del espejo está completa)	- Debe esperar a que se complete la operación del espejo antes de acceder a los datos - Debe gestionar una copia remota - No hay capacidad de escribir de nuevo a la fuente

Opción 1: Crear una caché de un volumen que reside en un plano de datos diferente

Con "Tecnología NetApp FlexCache", Puede crear una caché de un volumen NetApp que reside en un plano de datos diferente. Por ejemplo, si tiene un volumen de NetApp en su plano de datos local y necesita acceder a dicho volumen en su plano de datos de AWS, puede crear una caché del volumen en AWS. En esta sección se describen las tareas que se deben realizar para crear una caché de un volumen NetApp que se encuentre en un plano de datos diferente.

Crear volumen FlexCache en entorno de destino

> Si el entorno de destino es su centro de datos en las instalaciones, creará el volumen FlexCache en su sistema ONTAP en las instalaciones. Si el entorno de destino es AWS, crearás el volumen de FlexCache en tu instancia de Amazon FSx para NetApp ONTAP.

Primero, debe crear un volumen de FlexCache en el entorno de destino.

Recomendamos utilizar BlueXP para crear el volumen de FlexCache. Para crear un volumen de FlexCache con BlueXP, siga las instrucciones que se describen en la "Documentación de almacenamiento en caché de volúmenes de BlueXP".

Si prefiere no usar BlueXP, puede usar ONTAP System Manager o la interfaz de línea de comandos de ONTAP para crear el volumen de FlexCache. Para crear un volumen FlexCache con System Manager, consulte las instrucciones descritas en la "Documentación de ONTAP". Para crear un volumen de FlexCache con la CLI de ONTAP, consulte las instrucciones que se describen en "Documentación de ONTAP".

Si desea automatizar este proceso, puede utilizar el "API de BlueXP", la "API REST de ONTAP", o el "Colección Ansible de ONTAP".

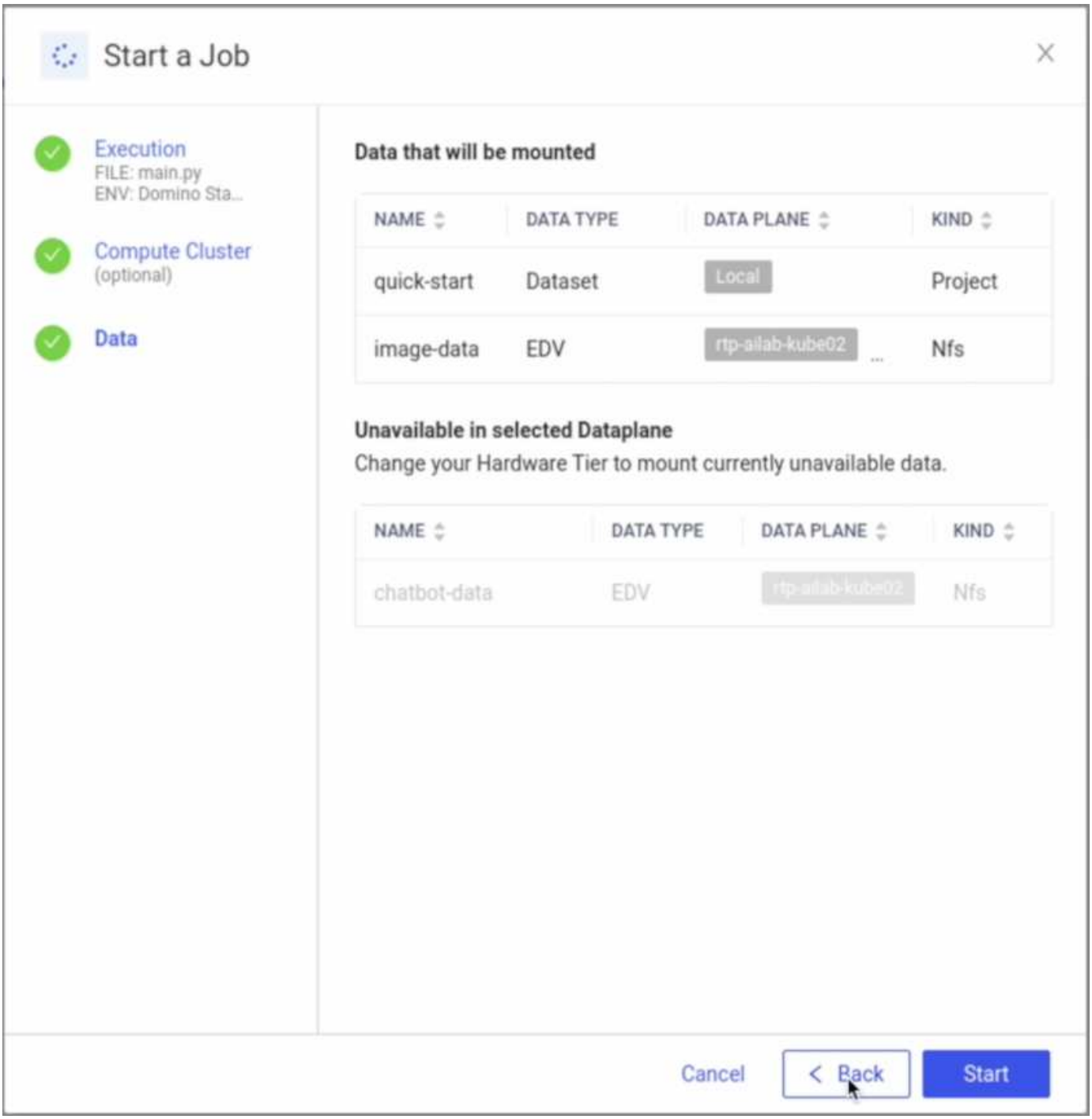
System Manager no está disponible en Amazon FSx para NetApp ONTAP.

Exponga el volumen FlexCache a Domino

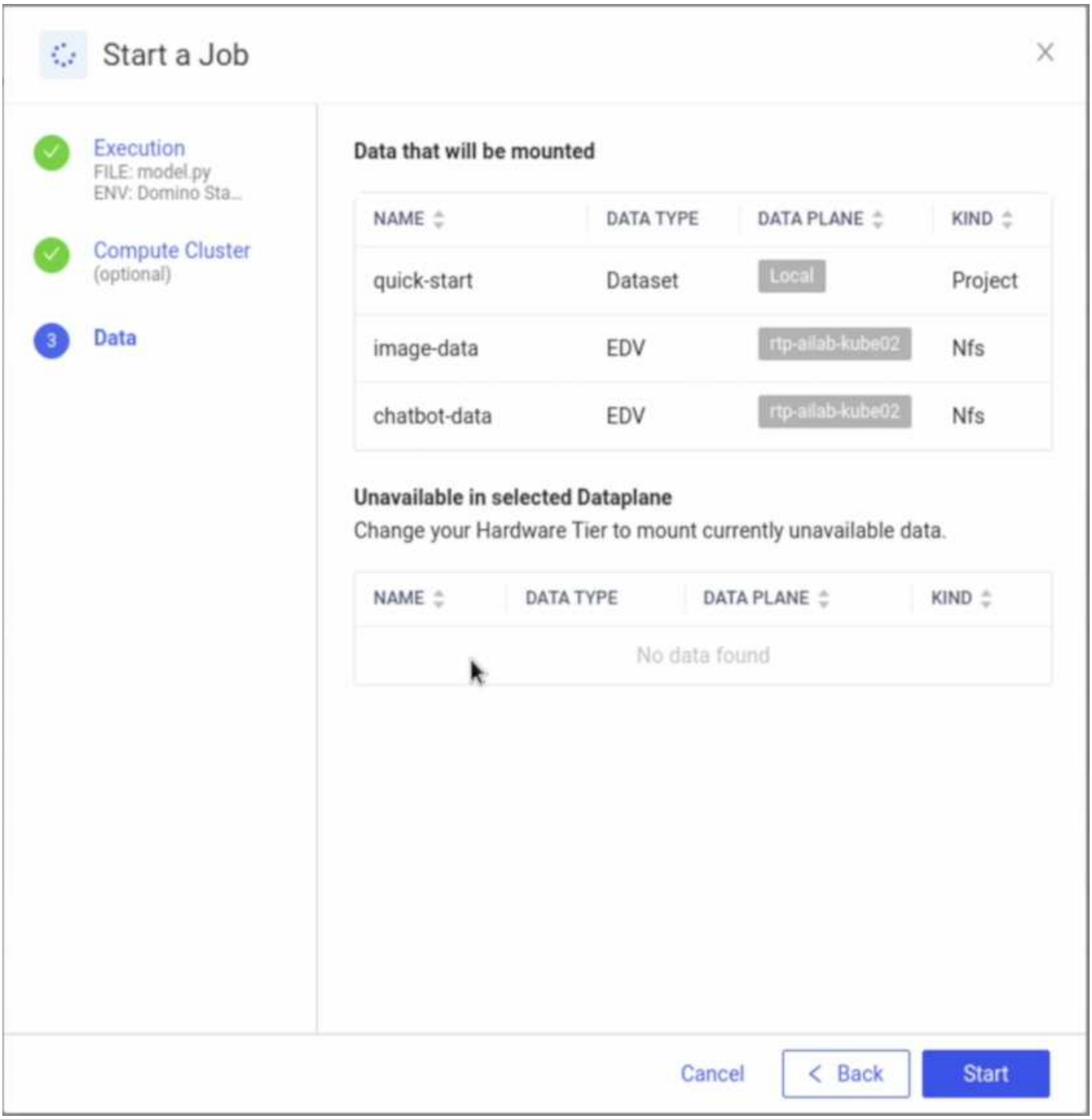
A continuación, debe exponer el volumen FlexCache a la plataforma Domino MLOps. Para exponer el volumen FlexCache a Domino, siga las instrucciones que se describen en la subsección 'Exponga volúmenes NFS existentes que no se aprovisionaban con Astra Trident' de la "Sección 'Exponga los volúmenes NetApp existentes a Domino'" de esta solución.

Ahora, podrá montar el volumen FlexCache al iniciar trabajos y espacios de trabajo en el plano de datos de destino, como se muestra en las siguientes capturas de pantalla.

Antes de crear Volumen FlexCache



Después de exponer el volumen de FlexCache a Domino



Opción 2: Replicar un volumen que reside en un plano de datos distinto

Con "Tecnología de replicación de datos SnapMirror de NetApp", Puede crear una copia de un volumen NetApp que reside en un plano de datos diferente. Por ejemplo, si tiene un volumen de NetApp en su plano de datos local y necesita acceder a dicho volumen en su plano de datos de AWS, puede crear una copia del volumen en AWS. En esta sección se describen las tareas que se deben realizar para crear una copia de un volumen NetApp que se encuentre en un plano de datos diferente.

Crear una relación de SnapMirror

Primero, debe crear una relación de SnapMirror entre el volumen de origen y un nuevo volumen de destino en el entorno de destino. Tenga en cuenta que el volumen de destino se creará como parte del proceso de creación de la relación de SnapMirror.

Recomendamos usar BlueXP para crear la relación de SnapMirror. Para crear una relación de SnapMirror con BlueXP, siga las instrucciones que se describen en "Documentación de replicación de BlueXP".

Si prefiere no usar BlueXP, puede usar ONTAP System Manager o la interfaz de línea de comandos de ONTAP para crear la relación de SnapMirror. Para crear una relación de SnapMirror con System Manager, consulte las instrucciones descritas en "Documentación de ONTAP". Para crear una relación de SnapMirror con la CLI de ONTAP, consulte las instrucciones descritas en "Documentación de ONTAP".

Si desea automatizar este proceso, puede utilizar el "API de BlueXP", la "API REST de ONTAP", o el "Colección Ansible de ONTAP".

> System Manager no está disponible en Amazon FSx para NetApp ONTAP.

Interrumpir la relación de SnapMirror

Luego, debe interrumpir la relación de SnapMirror para activar el volumen de destino para el acceso a los datos. Espere hasta que se complete la replicación inicial antes de realizar este paso.

> Puede determinar si la replicación está completa o no, compruebe el estado de reflejo en BlueXP, ONTAP System Manager o la CLI de ONTAP. Cuando se complete la replicación, el estado de reflejo será Snapmirror.

Recomendamos usar BlueXP para interrumpir la relación de SnapMirror. Para interrumpir una relación de SnapMirror con BlueXP, siga las instrucciones indicadas en "Documentación de replicación de BlueXP".

Si prefiere no usar BlueXP, puede usar ONTAP System Manager o la interfaz de línea de comandos de ONTAP para interrumpir la relación de SnapMirror. Para interrumpir una relación de SnapMirror con System Manager, consulte las instrucciones descritas en "Documentación de ONTAP". Para interrumpir una relación de SnapMirror con la CLI de ONTAP, consulte las instrucciones que se describen en "Documentación de ONTAP".

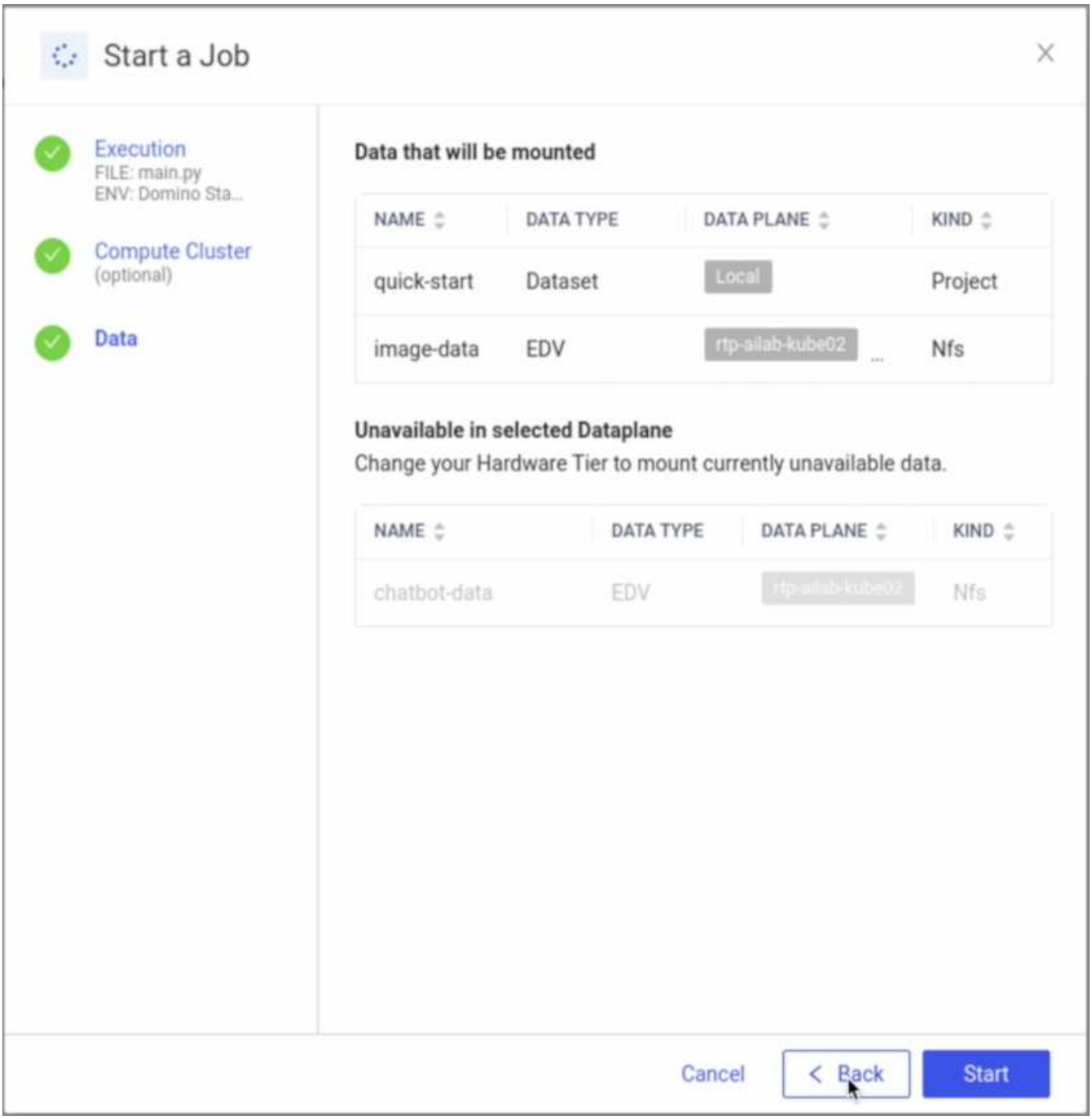
Si desea automatizar este proceso, puede utilizar el "API de BlueXP", la "API REST de ONTAP", o el "Colección Ansible de ONTAP".

Exponga el volumen de destino a Domino

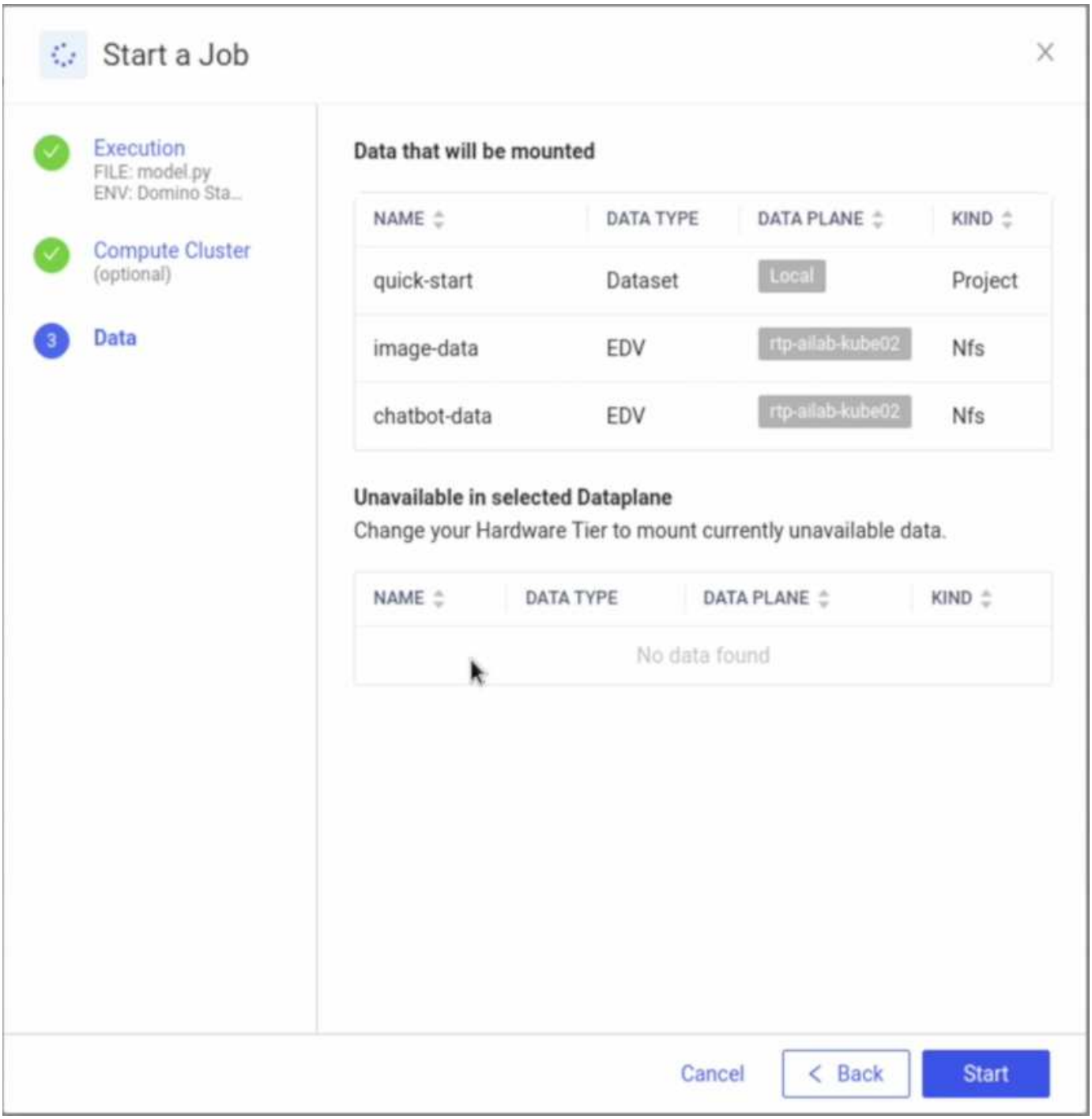
A continuación, debe exponer el volumen de destino a la plataforma Domino MLOps. Para exponer el volumen de destino a Domino, siga las instrucciones que se describen en la subsección 'Exponga volúmenes NFS existentes que no se aprovisionaban con Astra Trident' de la "Sección 'Exponga los volúmenes NetApp existentes a Domino'" de esta solución.

Ahora, podrá montar el volumen de destino al iniciar trabajos y espacios de trabajo en el plano de datos de destino como se muestra en las siguientes capturas de pantalla.

Antes de crear una relación de SnapMirror



Después de exponer el volumen de destino a Domino



Dónde encontrar información adicional

Para obtener más información sobre la información descrita en este documento, consulte los siguientes documentos y/o sitios web:

- Domino Data Lab

  "https://domino.ai"

- Nexus de Domino

  "https://domino.ai/platform/nexus"

- BlueXP de NetApp

  "https://bluexp.netapp.com"

- Software de gestión de datos ONTAP de NetApp

  "https://www.netapp.com/data-management/ontap-data-management-software/"

- Soluciones de IA de NetApp

  "https://www.netapp.com/artificial-intelligence/"

Reconocimientos

- Josh Mineroff, director ejecutivo de SA de Tech Alliances en Domino Data Lab
- Nicholas Jablonski, director técnico de campo, Domino Data Lab
- Es/Prabu Arjunan, arquitecto de soluciones en NetApp
- Brian Young, director de alianza global, partners de alianza tecnológica de NetApp

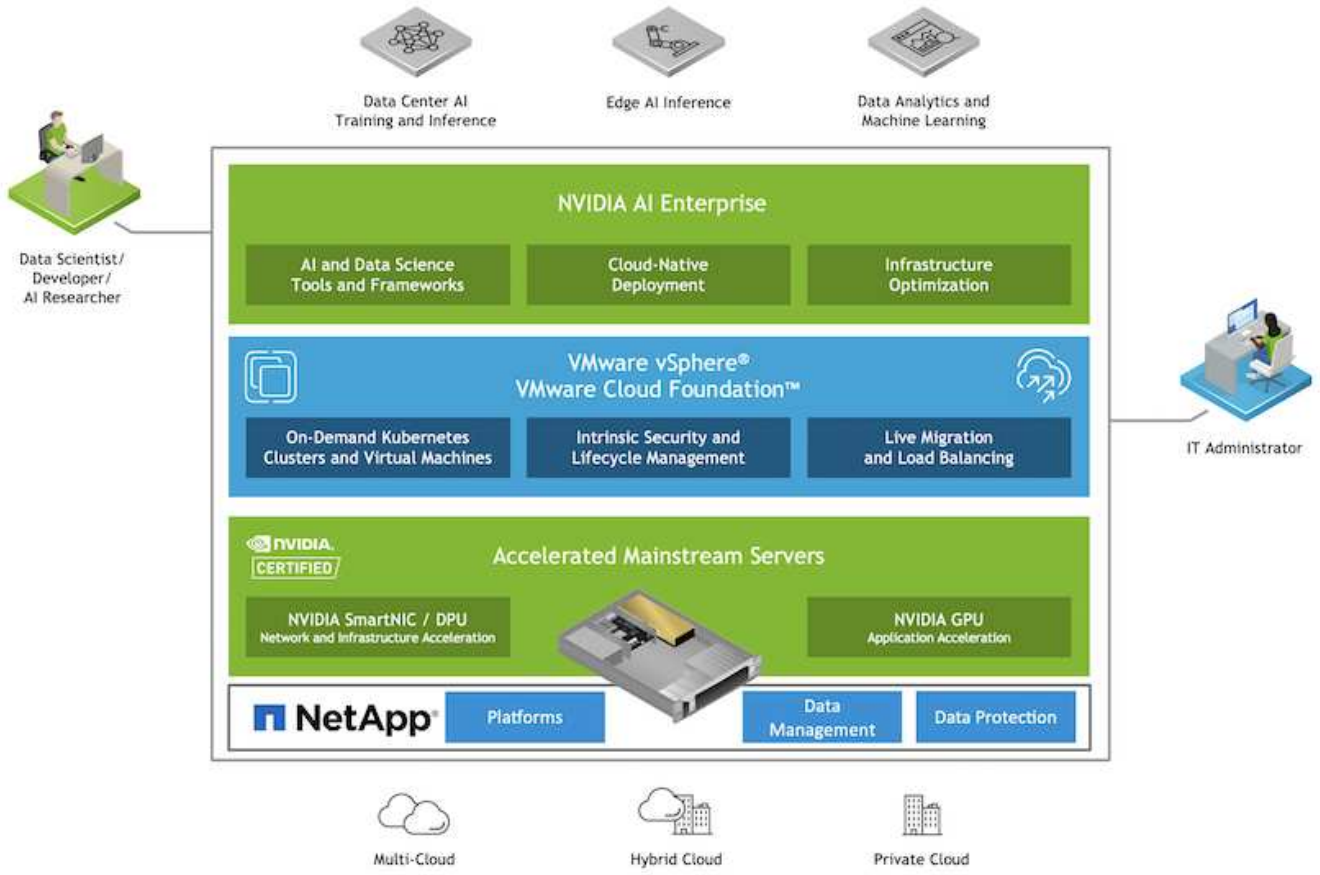
NVIDIA AI Enterprise con NetApp y VMware

NVIDIA AI Enterprise con NetApp y VMware

Mike Oglesby, NetApp

Para los arquitectos y administradores DE TECNOLOGÍA, las herramientas de IA pueden ser complicadas y no conocidas. Además, muchas plataformas de IA no están listas para las empresas. NVIDIA AI Enterprise, con la tecnología de NetApp y VMware, se ha creado para proporcionar una arquitectura de IA optimizada para la gran empresa.

NVIDIA AI Enterprise es una suite integral y nativa en el cloud de software de IA y análisis de datos optimizado, certificado y compatible con NVIDIA para ejecutarse en VMware vSphere con sistemas certificados por NVIDIA. Este software facilita la puesta en marcha, la gestión y el escalado simples y rápidos de las cargas de trabajo de IA en el entorno de cloud híbrido moderno. NVIDIA AI Enterprise, con la tecnología de NetApp y VMware, ofrece una excelente carga de trabajo de IA y gestión de datos en un paquete simplificado y conocido.



Visión general de la tecnología

IA Enterprise de NVIDIA

NVIDIA AI Enterprise es una suite integral y nativa en el cloud de software de IA y análisis de datos optimizado, certificado y compatible con NVIDIA para ejecutarse en VMware vSphere con sistemas certificados por NVIDIA. Este software facilita la puesta en marcha, la gestión y el escalado simples y rápidos de las cargas de trabajo de IA en el entorno de cloud híbrido moderno.

GPU CLOUD DE NVIDIA (NGC)

NVIDIA NGC aloja un catálogo de software optimizado para GPU para que los profesionales de la IA puedan desarrollar sus soluciones de IA. También proporciona acceso a diversos servicios de IA, incluido NVIDIA base Command para el entrenamiento de modelos, NVIDIA Fleet Command para implementar y supervisar modelos, y el registro privado NGC para acceder y gestionar de forma segura el software de IA propio. Además, los clientes de NVIDIA AI Enterprise pueden solicitar soporte a través del portal NGC.

VSphere de VMware

VMware vSphere es la plataforma de virtualización de VMware, que transforma los centros de datos en infraestructuras informáticas globales que incluyen recursos de CPU, almacenamiento y red. VSphere gestiona estas infraestructuras como un entorno operativo unificado y proporciona a los administradores las herramientas necesarias para gestionar los centros de datos que participan en ese entorno.

Los dos componentes principales de vSphere son ESXi y vCenter Server. ESXi es la plataforma de virtualización donde los administradores crean y ejecutan máquinas virtuales y dispositivos virtuales. VCenter

Server es el servicio a través del cual los administradores gestionan varios hosts conectados en una red y un pool de recursos de host.

ONTAP de NetApp

ONTAP 9, la última generación del software de gestión del almacenamiento de NetApp, permite a las empresas modernizar su infraestructura y realizar la transición a un centro de datos preparado para el cloud. ONTAP ofrece las mejores capacidades de gestión de datos y permite la gestión y protección de los datos con un solo conjunto de herramientas, sin importar dónde residan. También puede mover los datos libremente a donde sea necesario: El perímetro, el núcleo o el cloud. ONTAP 9 incluye numerosas funciones que simplifican la gestión de datos, aceleran y protegen los datos esenciales y permiten disfrutar de funcionalidades de infraestructura de nueva generación en arquitecturas de cloud híbrido.

Simplificar la gestión de los datos

La gestión de los datos es crucial para las operaciones TECNOLÓGICAS empresariales y los científicos de datos, para que se utilicen recursos apropiados para las aplicaciones de IA y para entrenar conjuntos de datos de IA/ML. La siguiente información adicional sobre las tecnologías de NetApp no está disponible para esta validación, pero puede ser relevante en función de su puesta en marcha.

El software para la gestión de datos ONTAP incluye las siguientes funciones para mejorar y simplificar las operaciones, y reducir el coste total de funcionamiento:

- Compactación de datos inline y deduplicación expandida. La compactación de datos reduce el espacio perdido dentro de los bloques de almacenamiento, mientras que la deduplicación aumenta la capacidad efectiva de forma significativa. Esto es aplicable a los datos almacenados localmente y a los datos organizados en niveles en el cloud.
- Calidad de servicio (AQoS) mínima, máxima y adaptativa. Los controles granulares de calidad de servicio (QoS) ayudan a mantener los niveles de rendimiento para aplicaciones críticas en entornos altamente compartidos.
- FabricPool de NetApp. Proporciona la organización automática en niveles de datos fríos en opciones de almacenamiento en cloud privado como Amazon Web Services (AWS), Azure y la solución de almacenamiento StorageGRID de NetApp. Para obtener más información sobre FabricPool, consulte "TR-4598: Prácticas recomendadas de FabricPool".

Acelere y proteja sus datos

ONTAP no solo ofrece niveles de rendimiento y protección de datos superiores, sino que amplía estas capacidades de las siguientes maneras:

- Rendimiento y menor latencia. ONTAP ofrece la salida más alta posible con la menor latencia posible.
- Protección de datos. ONTAP ofrece capacidades integradas de protección de datos, con una administración común entre todas las plataformas.
- Cifrado de volúmenes de NetApp (NVE). ONTAP ofrece cifrado nativo en el nivel de volumen y permite la gestión de claves incorporada o externa.
- Multi-tenancy y autenticación multifactor. ONTAP permite compartir recursos de infraestructura con los niveles más altos de seguridad.

Infraestructura preparada para futuros retos

ONTAP ayuda a satisfacer las exigentes y siempre cambiantes necesidades de su empresa con las siguientes funciones:

- Escalado sencillo y operaciones no disruptivas. ONTAP admite la adición no disruptiva de capacidad a las controladoras existentes y a clústeres de escalado horizontal. Los clientes pueden empezar a utilizar tecnologías punteras como NVMe y FC 32 GB, sin necesidad de realizar costosas migraciones de datos y sin cortes.
- Conexión de cloud. ONTAP es el software de gestión de almacenamiento con mejor conexión de cloud e incluye opciones de almacenamiento definido por software (ONTAP Select) e instancias nativas del cloud (NetApp Cloud Volumes Service) en todos los clouds públicos.
- Integración con aplicaciones emergentes. ONTAP ofrece servicios de datos de clase empresarial para plataformas y aplicaciones de última generación, como vehículos autónomos, ciudades inteligentes e Industria 4.0, utilizando la misma infraestructura que da soporte a las aplicaciones empresariales existentes.

Kit de herramientas de operaciones de datos de NetApp

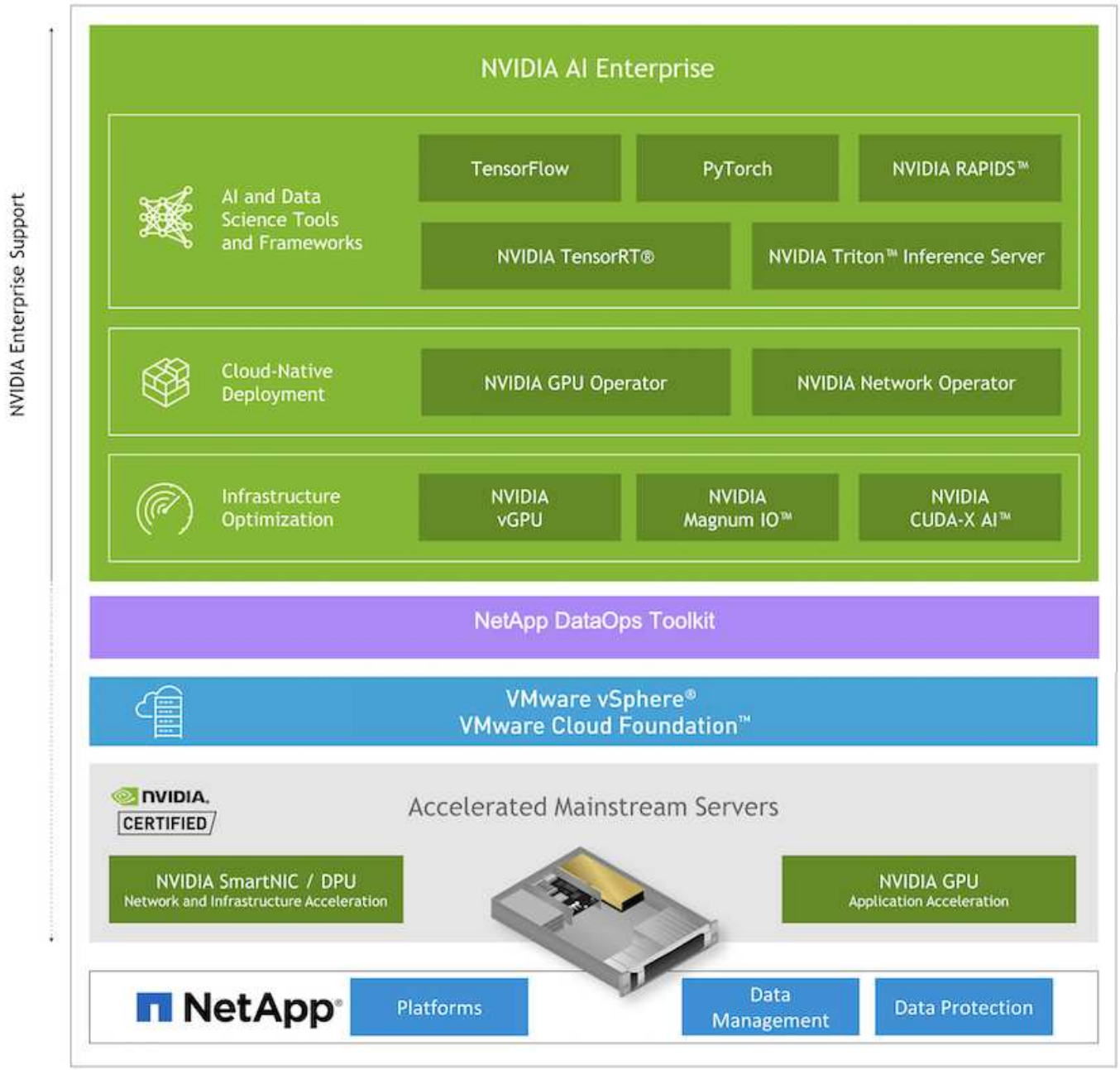
El kit de herramientas DataOps de NetApp es una herramienta basada en Python que simplifica la gestión de espacios de trabajo de desarrollo/formación y servidores de inferencia respaldados por un almacenamiento de NetApp de escalado horizontal y de alto rendimiento. Estas son algunas funcionalidades clave:

- Aprovisione con rapidez nuevos espacios de trabajo de JuppyterLab de alta capacidad respaldados por el almacenamiento de NetApp de escalado horizontal y de alto rendimiento.
- Aprovisione rápidamente nuevas instancias del servidor de inferencia de NVIDIA Triton, respaldadas por un almacenamiento empresarial de NetApp.
- Clone por vía casi instreutilizada los espacios de trabajo de JuppyterLab de gran capacidad para permitir la experimentación o la iteración rápida.
- Guarde copias Snapshot casi por vía inutilizada de espacios de trabajo JuppyterLab de gran capacidad para backup o seguimiento o creación de bases.
- Aprovisionamiento, clonado y copias Snapshot de alta capacidad y alto rendimiento casi por vía casi instusand.

Arquitectura

Esta solución se basa en una arquitectura probada y familiar que incluye NetApp, VMware y sistemas con certificación NVIDIA. Consulte la siguiente tabla para obtener más información.

Componente	Detalles
Software de IA y análisis de datos	"NVIDIA AI Enterprise para VMware"
Plataforma de virtualización	"VSphere de VMware"
Plataforma informática	"Sistemas con certificación NVIDIA"
Plataforma de gestión de datos	"ONTAP de NetApp"



Configuración inicial

En esta sección se describen las tareas de configuración inicial que se deben realizar para utilizar NVIDIA AI Enterprise con NetApp y VMware.

Requisitos previos

Antes de realizar los pasos descritos en esta sección, asumimos que ya ha implementado VMware vSphere y ONTAP de NetApp. Consulte la "Matriz de compatibilidad de productos empresariales de NVIDIA AI" Para obtener más detalles sobre las versiones de vSphere compatibles. Consulte la "Documentación sobre las soluciones de NetApp y VMware" Para obtener más información sobre cómo implementar VMware vSphere con ONTAP de NetApp.

Instale el software de host de NVIDIA AI Enterprise

Para instalar el software host NVIDIA AI Entrprise, siga las instrucciones que se describen en las secciones 1-4 de la "Guía de inicio rápido de NVIDIA AI Enterprise".

Utilice el software NVIDIA NGC

En esta sección, se describen las tareas que se deben realizar para utilizar el software empresarial NVIDIA NGC en un entorno de NVIDIA AI Enterprise.

Configuración

En esta sección se describen las tareas de configuración inicial que se deben realizar para utilizar el software empresarial NVIDIA NGC en un entorno de NVIDIA AI Enterprise.

Requisitos previos

Antes de realizar los pasos descritos en esta sección, asumimos que ya ha implementado el software host NVIDIA AI Entrprise siguiendo las instrucciones que se describen en la "Configuración inicial" página.

Cree una máquina virtual Ubuntu Guest con vGPU

En primer lugar, debe crear una máquina virtual invitada Ubuntu 20.04 con vGPU. Para crear una VM huésped de Ubuntu 20.04 con vGPU, siga las instrucciones descritas en el "Guía de puesta en marcha de NVIDIA AI Enterprise".

Descargue e instale el software invitado de NVIDIA

A continuación, debe instalar el software invitado NVIDIA requerido en la máquina virtual invitada que creó en el paso anterior. Para descargar e instalar el software invitado NVIDIA necesario en el equipo virtual invitado, siga las instrucciones que se describen en las secciones 5.1-5.4 de la "Guía de inicio rápido de NVIDIA AI Enterprise".

> Al realizar las tareas de verificación descritas en la sección 5.4, es posible que necesite utilizar una etiqueta de versión de imagen de contenedor CUDA diferente ya que la imagen de contenedor CUDA se ha actualizado desde la escritura de la guía. En nuestra validación, utilizamos 'nvidia/cuda:11.0.3-base-ubuntu20.04'.

Descargue contenedores de marcos de análisis/IA

A continuación, debe descargar las imágenes de contenedor de IA o marco de análisis necesarias de NVIDIA NGC para que estén disponibles en la máquina virtual invitada. Para descargar los contenedores de marco en el equipo virtual invitado, siga las instrucciones que se describen en "Guía de puesta en marcha de NVIDIA AI Enterprise".

Instalación y configuración del kit de herramientas de operaciones de datos de NetApp

A continuación, debe instalar el kit de herramientas DataOPS de NetApp para entornos tradicionales en el equipo virtual invitado. El kit de herramientas DataOPS de NetApp se puede usar para gestionar los volúmenes de datos de escalado horizontal en su sistema ONTAP directamente desde el terminal dentro del equipo virtual «guest». Para instalar el kit de herramientas DataOPS de NetApp en el equipo virtual invitado, realice las siguientes tareas.

1. Instalar la tubería.

```
$ sudo apt update
$ sudo apt install python3-pip
$ python3 -m pip install netapp-dataops-traditional
```

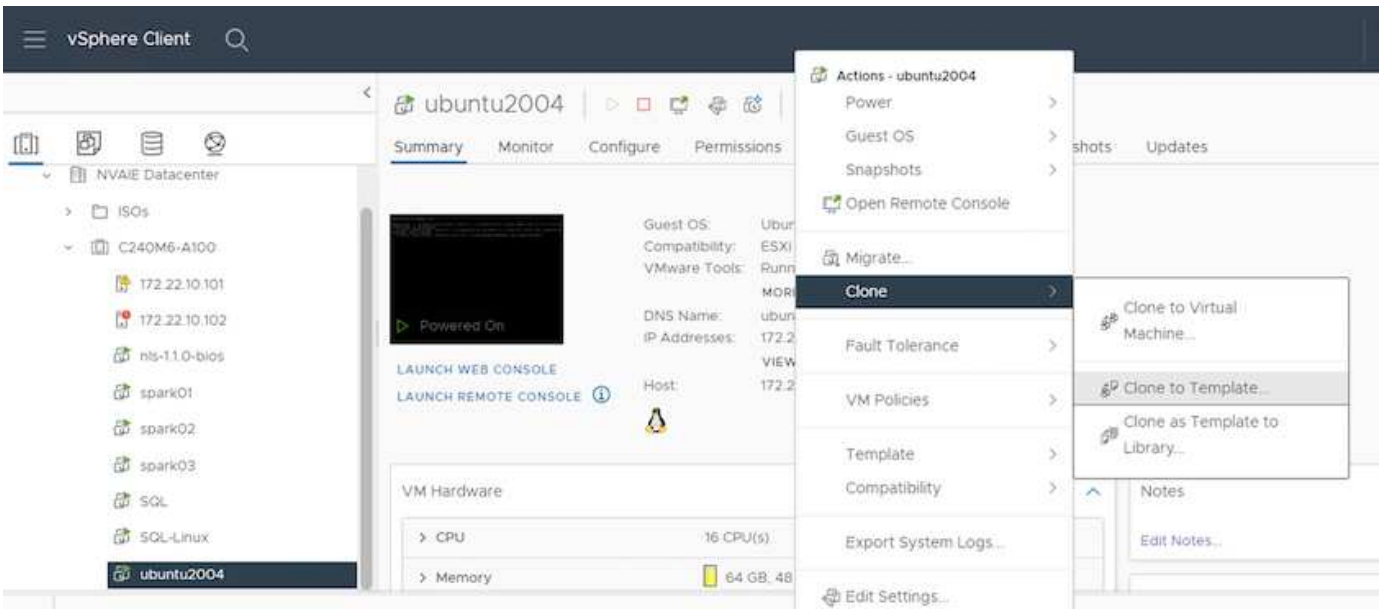
2. Cierre la sesión en el terminal de la máquina virtual invitada y vuelva a iniciarla.
3. Configurar el kit de herramientas de operaciones de datos de NetApp. Para completar este paso, necesitará detalles de acceso a la API para su sistema ONTAP. Es posible que tenga que obtenerlos del administrador de almacenamiento.

```
$ netapp_dataops_cli.py config

Enter ONTAP management LIF hostname or IP address (Recommendation: Use
SVM management interface): 172.22.10.10
Enter SVM (Storage VM) name: NVAIE-client
Enter SVM NFS data LIF hostname or IP address: 172.22.13.151
Enter default volume type to use when creating new volumes
(flexgroup/flexvol) [flexgroup]:
Enter export policy to use by default when creating new volumes
[default]:
Enter snapshot policy to use by default when creating new volumes
[none]:
Enter unix filesystem user id (uid) to apply by default when creating
new volumes (ex. '0' for root user) [0]:
Enter unix filesystem group id (gid) to apply by default when creating
new volumes (ex. '0' for root group) [0]:
Enter unix filesystem permissions to apply by default when creating new
volumes (ex. '0777' for full read/write permissions for all users and
groups) [0777]:
Enter aggregate to use by default when creating new FlexVol volumes:
aff_a400_01_NVME_SSD_1
Enter ONTAP API username (Recommendation: Use SVM account): admin
Enter ONTAP API password (Recommendation: Use SVM account):
Verify SSL certificate when calling ONTAP API (true/false): false
Do you intend to use this toolkit to trigger BlueXP Copy and Sync
operations? (yes/no): no
Do you intend to use this toolkit to push/pull from S3? (yes/no): no
Created config file: '/home/user/.netapp_dataops/config.json'.
```

Cree una plantilla de máquina virtual invitada

Por último, debe crear una plantilla de equipo virtual basada en el equipo virtual «guest». Podrá utilizar esta plantilla para crear rápidamente equipos virtuales invitados con el software NVIDIA NGC.

Si desea crear una plantilla de máquina virtual basada en su equipo virtual invitado, inicie sesión en VMware vSphere, haga clic con el botón derecho en el nombre de la máquina virtual invitada, seleccione 'Clonar', elija 'Clonar en plantilla…' y, a continuación, siga el asistente.



Ejemplo de caso práctico: Trabajo de formación de TensorFlow

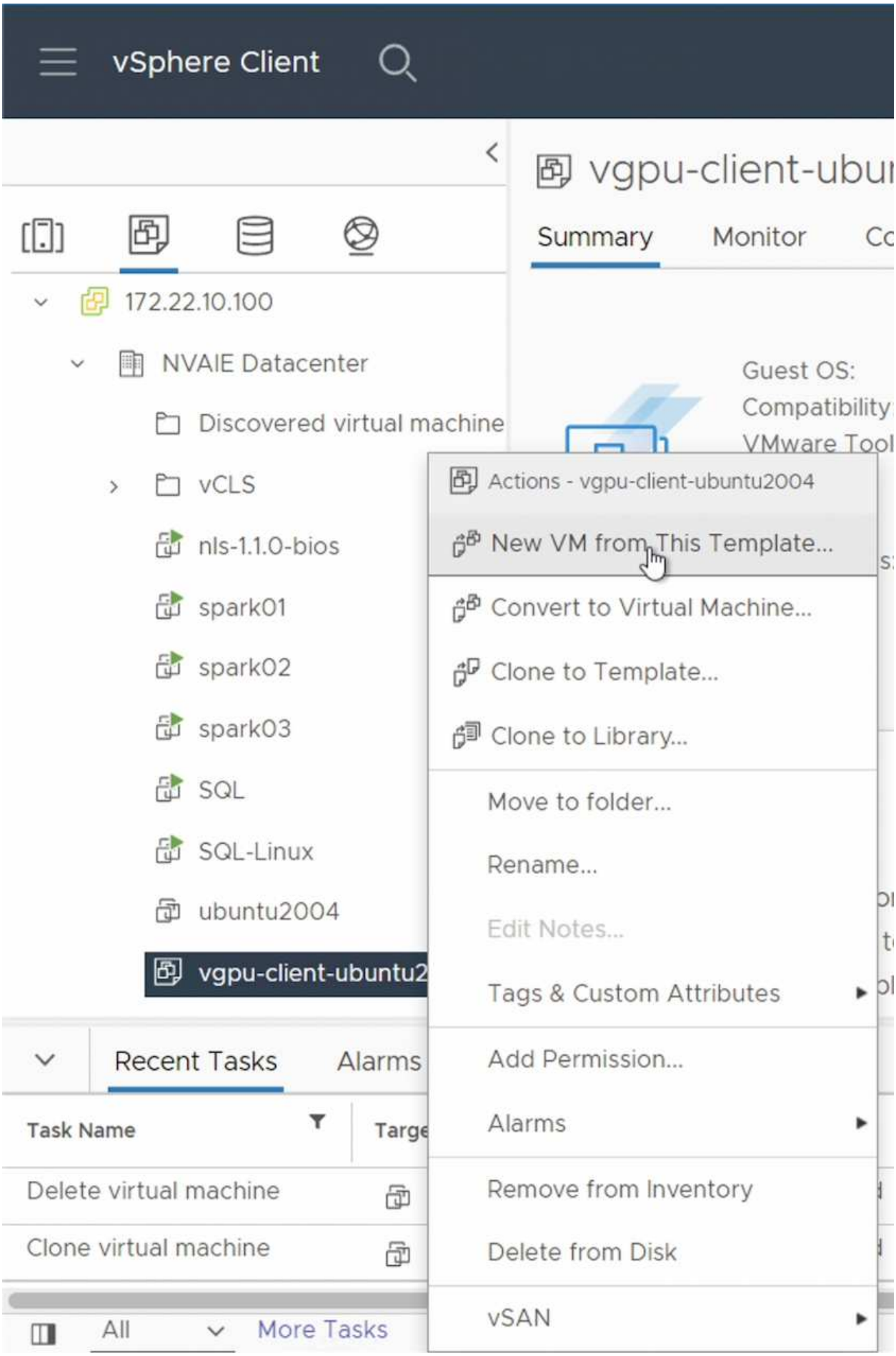
En esta sección se describen las tareas que se deben realizar para ejecutar una tarea de entrenamiento de TensorFlow dentro de un entorno NVIDIA AI Enterprise.

Requisitos previos

Antes de realizar los pasos descritos en esta sección, asumimos que ya ha creado una plantilla de equipo virtual invitado siguiendo las instrucciones que se describen en la "Configuración" página.

Crear máquina virtual invitada a partir de la plantilla

En primer lugar, debe crear un nuevo equipo virtual invitado a partir de la plantilla que ha creado en la sección anterior. Para crear una máquina virtual invitada nueva desde la plantilla, inicie sesión en VMware vSphere, haga clic en el nombre de la plantilla, seleccione "New VM from this Template…" y, a continuación, siga el asistente.


vSphere Client
vgpu-client-ubu
Summary
Monitor
172.22.10.100
NVAIE Datacenter
Discovered virtual machine
vCLS
nls-1.1.0-bios
spark01
spark02
spark03
SQL
SQL-Linux
ubuntu2004
vgpu-client-ubuntu2
Guest OS:
Compatibility
VMware Tool
Actions - vgpu-client-ubuntu2004
New VM from This Template...
Convert to Virtual Machine...
Clone to Template...
Clone to Library...
Move to folder...
Rename...
Edit Notes...
Tags & Custom Attributes
Add Permission...
Alarms
Remove from Inventory
Delete from Disk
vSAN
Recent Tasks
Alarms
Task Name
Delete virtual machine
Clone virtual machine
All
More Tasks

Cree y monte el volumen de datos

A continuación, debe crear un nuevo volumen de datos en el que almacenar el conjunto de datos de entrenamiento. Puede crear rápidamente un nuevo volumen de datos con el kit de herramientas de operaciones de datos de NetApp. El siguiente comando de ejemplo muestra la creación de un volumen llamado 'imagenet' con una capacidad de 2 TB.

```
$ netapp_dataops_cli.py create vol -n imagenet -s 2TB
```

Antes de poder completar el volumen de datos con datos, debe montarlo en la máquina virtual invitada. Puede montar rápidamente un volumen de datos con el kit de herramientas Data OPS de NetApp. El comando de ejemplo siguiente muestra el bigote del volumen que se creó en el paso anterior.

```
$ sudo -E netapp_dataops_cli.py mount vol -n imagenet -m ~/imagenet
```

Completar volumen de datos

Una vez aprovisionado y montado el volumen nuevo, el conjunto de datos de entrenamiento puede recuperarse de la ubicación de origen y colocarse en el volumen nuevo. Normalmente, esto implica extraer los datos de un lago de datos de S3 o Hadoop y a veces contará con la ayuda de un ingeniero de datos.

Ejecutar la tarea de formación de TensorFlow

Ahora, estará listo para ejecutar su trabajo de formación de TensorFlow. Para ejecutar el trabajo de entrenamiento de TensorFlow, realice las siguientes tareas.

1. Tire de la imagen del contenedor NVIDIA NGC para TensorFlow empresarial.

   ```
   $ sudo docker pull nvcr.io/nvaie/tensorflow-2-1:22.05-tf1-nvaie-2.1-py3
   ```

2. Inicie una instancia del contenedor NVIDIA NGC para TensorFlow empresarial. Utilice la opción '-v' para adjuntar el volumen de datos al contenedor.

   ```
   $ sudo docker run --gpus all -v ~/imagenet:/imagenet -it --rm
   nvcr.io/nvaie/tensorflow-2-1:22.05-tf1-nvaie-2.1-py3
   ```

3. Ejecute su programa de formación TensorFlow en el contenedor. El comando de ejemplo siguiente muestra la ejecución de un programa de entrenamiento ResNet-50 de ejemplo que se incluye en la imagen contenedora.

   ```
   $ python ./nvidia-examples/cnn/resnet.py --layers 50 -b 64 -i 200 -u
   batch --precision fp16 --data_dir /imagenet/data
   ```

Dónde encontrar información adicional

Para obtener más información sobre la información descrita en este documento, consulte los siguientes documentos y/o sitios web:

- Software de gestión de datos ONTAP de NetApp: Biblioteca de información de ONTAP

  http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=62286

- Kit de herramientas de operaciones de datos de NetApp

  "https://github.com/NetApp/netapp-dataops-toolkit"

- NVIDIA AI Enterprise con VMware

  https://www.nvidia.com/en-us/data-center/products/ai-enterprise/vmware/^]

Reconocimientos

- Bobby Oommen, Sr. Sénior, NetApp
- Ramesh Isaac, Administrador de sistemas, NetApp
- Roney Daniel, Ingeniero Técnico de Marketing de NetApp

TR-4851: Lago de datos de StorageGRID de NetApp para cargas de trabajo de conducción autónoma - Diseño de la solución

David Arnette, NetApp

En el TR-4851 se muestra el uso del almacenamiento de objetos StorageGRID de NetApp como repositorio de datos y sistema de gestión para el desarrollo de software de aprendizaje automático (ML) y aprendizaje profundo (DL). Este documento describe el flujo de datos y los requisitos del desarrollo de software de vehículos autónomos y las funciones de StorageGRID que optimizan el ciclo de vida de los datos. Esta solución se aplica a cualquier flujo de trabajo de canalización de datos multietapa típico en los procesos de desarrollo DE APRENDIZAJE AUTOMÁTICO y aprendizaje profundo.

"TR-4851: Lago de datos de StorageGRID de NetApp para cargas de trabajo de conducción autónoma - Diseño de la solución"

Plano de control de IA de NetApp

TR-4798: Plano de control de IA de NetApp

Mike Oglesby, NetApp

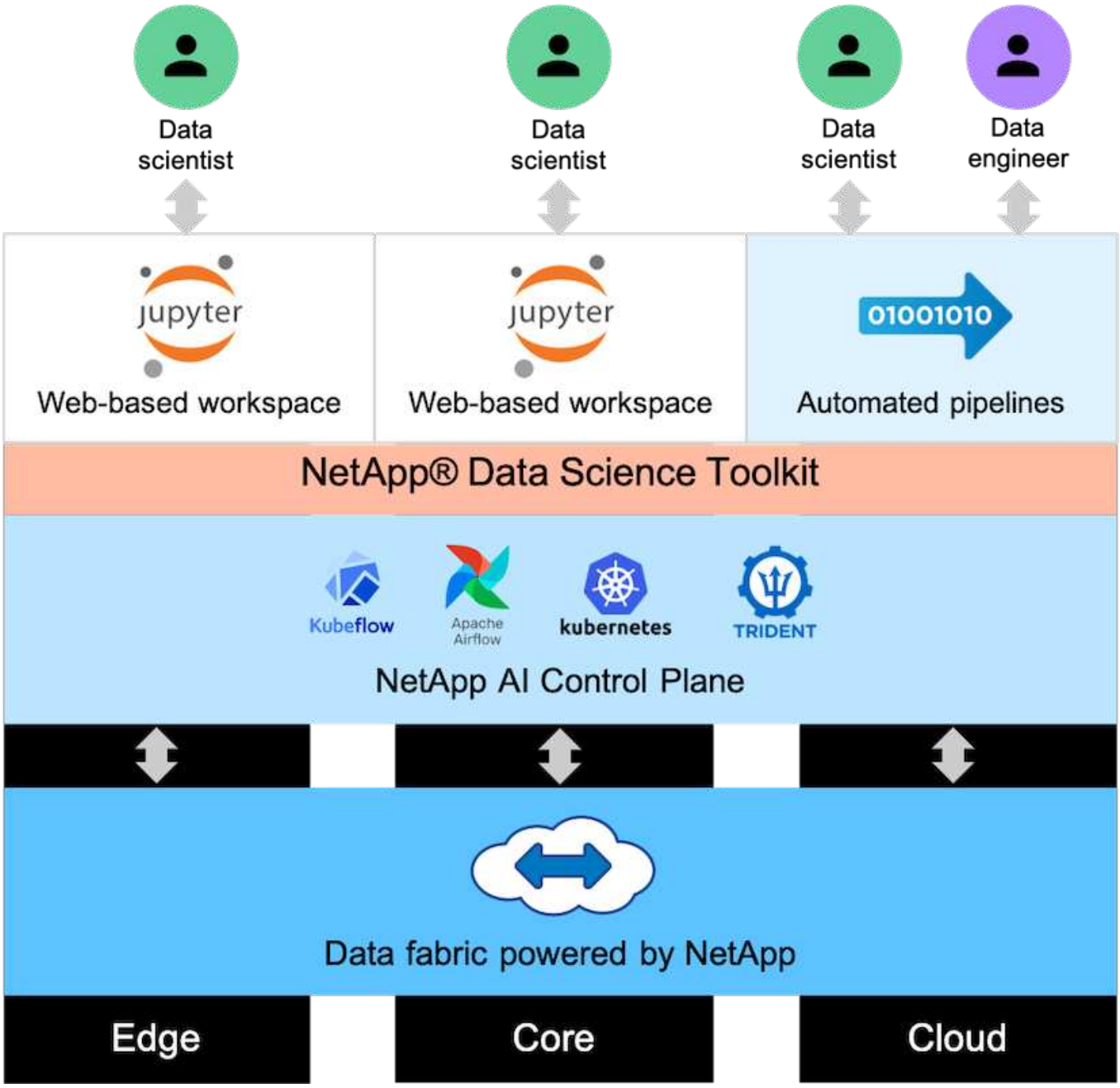
Empresas y organizaciones de todos los tamaños y sectores se están decantando por la inteligencia artificial (IA), el aprendizaje automático (ML) y el aprendizaje profundo (DL)

para resolver problemas reales, ofrecer productos y servicios innovadores y obtener una ventaja en un mercado cada vez más competitivo. A medida que las organizaciones aumentan el uso de la IA, EL ML y el AP, deben hacer frente a numerosos retos, como la escalabilidad de la carga de trabajo y la disponibilidad de los datos. En este documento se demuestra cómo puede hacer frente a estos retos usando el plano de control de IA de NetApp, una solución que empareja las funcionalidades de gestión de datos de NetApp con marcos y herramientas de código abierto más populares.

Este informe le muestra cómo clonar rápidamente un espacio de nombres de datos. También le muestra cómo replicar datos sin problemas en sitios y regiones para crear una canalización de datos coherente y unificada de IA/ML/DL. Además, le guía por la definición e implementación de flujos de trabajo de formación de IA, ML y DL que incorporan la creación casi instantánea de datos y líneas de base de modelos para su trazabilidad y versionado. Con esta solución, puede rastrear cada entrenamiento de modelo que se ejecuta en el conjunto de datos exacto que se utilizó para entrenar o validar el modelo. Por último, este documento le muestra cómo aprovisionar rápidamente espacios de trabajo de Juppyter Notebook con acceso a conjuntos de datos masivos.

Nota: Para entrenamiento distribuido al estilo HPC a escala que implique un gran número de servidores GPU que requieran acceso compartido al mismo conjunto de datos o, si necesita o prefiere un sistema de archivos paralelo, consulte el documento "TR-4890". En este informe técnico se describe cómo incluir "Solución de sistema de archivos en paralelo totalmente compatible de NetApp BeeGFS" Como parte del plano de control de IA de NetApp. Esta solución está diseñada para escalar a partir de un puñado de sistemas NVIDIA DGX A100, hasta un SuperPOD de 140 nodos completo.

El plano de control de IA de NetApp se dirige a los científicos e ingenieros de datos y, por lo tanto, se necesita experiencia mínima en NetApp o en ONTAP® de NetApp. Con esta solución, las funciones de gestión de datos se pueden ejecutar utilizando interfaces y herramientas sencillas y conocidas. Si ya dispone de almacenamiento de NetApp en su entorno, puede probar el plano de control de IA de NetApp hoy mismo. Si desea probar la solución pero no dispone de almacenamiento de NetApp, visite "cloud.netapp.com"Y puede volver a tener una solución de almacenamiento de NetApp basada en cloud en cuestión de minutos. La siguiente figura ofrece una visualización de la solución.



Conceptos y componentes

Inteligencia artificial

La IA es una disciplina informática en la que las computadoras están entrenadas para imitar las funciones cognitivas de la mente humana. Los desarrolladores de IA entrenan computadoras para aprender y resolver problemas de una manera similar, o incluso superior a, humanos. El aprendizaje profundo y el aprendizaje automático son subcampos de la IA. Las organizaciones adoptan cada vez más IA, ML y DL para dar soporte a sus necesidades empresariales cruciales. Algunos ejemplos son los siguientes:

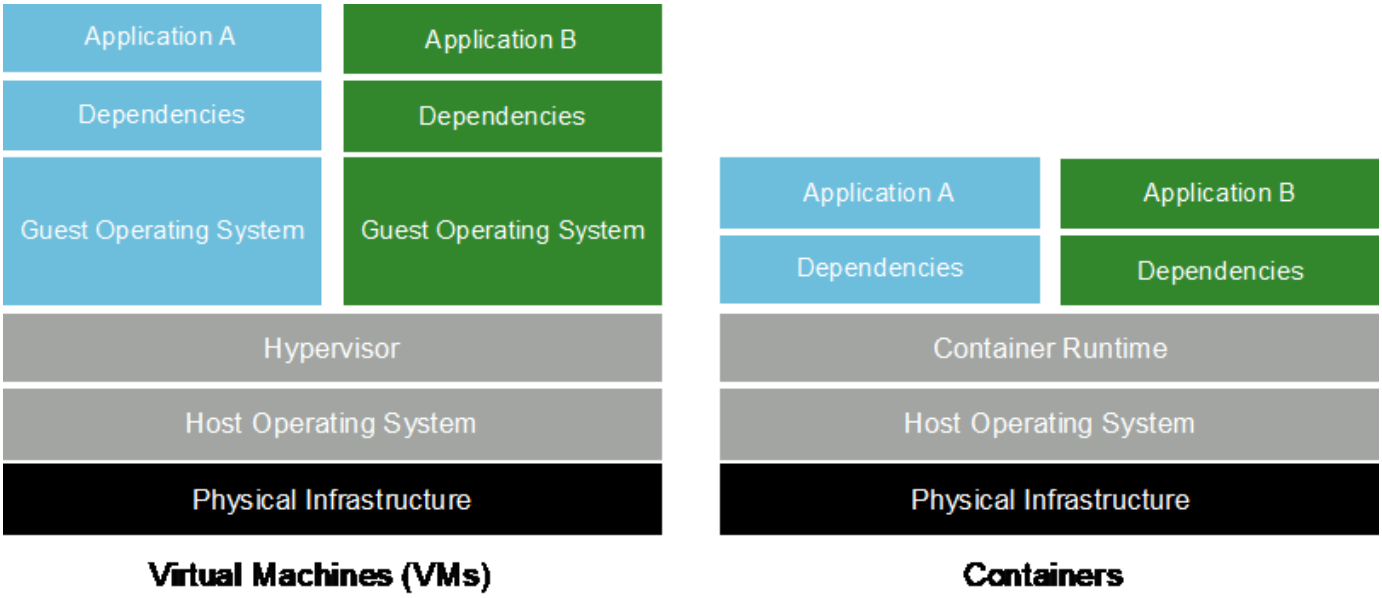
- Analizar grandes cantidades de datos para desconocer información empresarial anteriormente desconocida
- Interacción directa con los clientes mediante el procesamiento de lenguaje natural
- Automatización de diversos procesos y funciones empresariales

La formación de IA moderna y las cargas de trabajo de inferencia requieren de funcionalidades de computación en paralelo masivas. Por lo tanto, se están utilizando cada vez más GPU para ejecutar operaciones de IA, ya que las capacidades de procesamiento paralelo de las GPU son muy superiores a las de las CPU de uso general.

Contenedores

Los contenedores son instancias aisladas del espacio de usuario que se ejecutan sobre un kernel de sistema operativo host compartido. La adopción de contenedores aumenta rápidamente. Los contenedores ofrecen muchos de los mismos beneficios de uso de pruebas de espacio que las máquinas virtuales (VM). Sin embargo, debido a que se eliminan las capas de hipervisor y de sistema operativo «guest» de las que dependen las máquinas virtuales, los contenedores son mucho más ligeros. En la siguiente figura, se muestra una visualización de las máquinas virtuales en comparación con los contenedores.

Los contenedores también permiten el paquete eficiente de dependencias de aplicaciones, tiempos de ejecución, etc., directamente con una aplicación. El formato de embalaje de contenedor más utilizado es el contenedor Docker. Una aplicación que se haya contenedor en el formato de contenedor Docker se puede ejecutar en cualquier máquina que pueda ejecutar contenedores Docker. Esto es cierto incluso si las dependencias de la aplicación no están presentes en la máquina porque todas las dependencias están empaquetadas en el propio contenedor. Para obtener más información, visite la "Sitio web de Docker".



Kubernetes

Kubernetes es una plataforma de orquestación de contenedores distribuida de código abierto que originalmente diseñada por Google y que ahora se mantiene mediante Cloud Native Computing Foundation (CNCF). Kubernetes permite automatizar las funciones de puesta en marcha, gestión y escalado para aplicaciones en contenedores. En los últimos años, Kubernetes se ha convertido en la plataforma de orquestación de contenedores dominante. Aunque son compatibles con otros formatos de empaquetado y tiempos de ejecución, Kubernetes se utiliza con más frecuencia como un sistema de orquestación para contenedores Docker. Para obtener más información, visite la "Sitio web de Kubernetes".

Trident de NetApp

Trident es un orquestador de almacenamiento de código abierto desarrollado y mantenido por NetApp que simplifica en gran medida la creación, la gestión y el consumo de almacenamiento persistente para cargas de

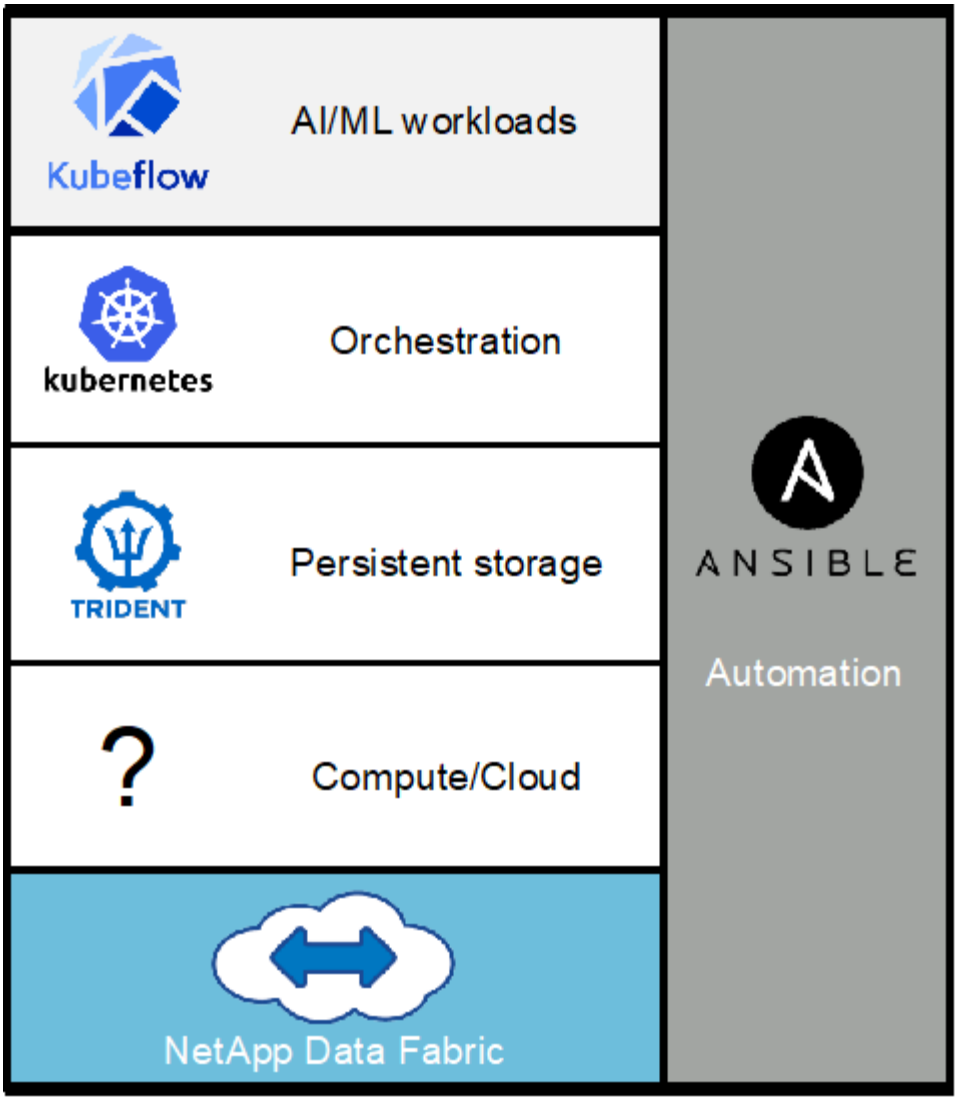
trabajo de Kubernetes. Trident, en sí misma una aplicación nativa de Kubernetes, se ejecuta directamente en un clúster de Kubernetes. Con Trident, los usuarios de Kubernetes (desarrolladores, científicos de datos, administradores de Kubernetes, etc.) pueden crear, gestionar e interactuar con volúmenes de almacenamiento persistente en el formato Kubernetes estándar, con el que ya están familiarizados. Al mismo tiempo, pueden aprovechar las funciones avanzadas de gestión de datos de NetApp y un Data Fabric con tecnología de NetApp. Trident elimina las complejidades del almacenamiento persistente y facilita el consumo. Para obtener más información, visite la "Sitio web de Trident".

DeepOps de NVIDIA

DeepOps es un proyecto de código abierto de NVIDIA que, con Ansible, automatiza la puesta en marcha de clústeres de servidores de GPU de acuerdo con las prácticas recomendadas. DeepOps es modular y se puede utilizar para realizar varias tareas de puesta en marcha. En este documento y en el ejercicio de validación descrito, DeepOps se utiliza para poner en marcha un clúster de Kubernetes que consta de nodos de trabajo de servidor GPU. Para obtener más información, visite la "Sitio web DeepOps".

Kubeflow

Kubeflow es un kit de herramientas DE IA Y ML de código abierto para Kubernetes que fue desarrollado originalmente por Google. El proyecto Kubeflow hace que la puesta en marcha de flujos de trabajo de IA y ML en Kubernetes sea sencilla, portátil y escalable. —Kubeflow elimina las complejidades de Kubernetes, lo que permite a los científicos de datos centrarse en lo que saben más ciencia de datos. Consulte la siguiente figura para ver una visualización. Kubeflow ha ido ganando terreno a medida que los departamentos DE TI de las empresas se han estandarizado cada vez más en Kubernetes. Para obtener más información, visite la "Sitio web de Kubeflow".



Tuberías de Kubeflow

Los oleoductos de Kubeflow son un componente clave de Kubeflow. Las canalizaciones de Kubeflow son una plataforma y un estándar para definir y poner en marcha flujos de trabajo DE IA Y ML escalables y portátiles. Para obtener más información, consulte "Documentación oficial de Kubeflow".

Servidor de portátiles Juppyter

Un servidor Juppyter Notebook es una aplicación web de código abierto que permite a los científicos de datos crear documentos similares a wiki llamados portátiles Juppyter que contienen código en vivo así como pruebas descriptivas. Los portátiles Juppyter se utilizan ampliamente en la comunidad de AI Y ML como medio para documentar, almacenar y compartir proyectos de IA y ML. Kubeflow simplifica el aprovisionamiento y la puesta en marcha de servidores para portátiles Juppyter en Kubernetes. Para obtener más información sobre los Cuadernos Jupyter, visite "Sitio Web de Juppyter". Para obtener más información acerca de Jupyter Notebooks en el contexto de Kubeflow, consulte "Documentación oficial de Kubeflow".

Flujo de aire Apache

Apache Airflow es una plataforma de gestión de flujos de trabajo de código abierto que permite la creación, programación y supervisión de programas para flujos de trabajo empresariales complejos. A menudo se utiliza para automatizar los flujos de trabajo de ETL y de canalización de datos, pero estos tipos de flujos de trabajo no se limitan a ellos. El proyecto de flujo de aire fue iniciado por Airbnb, pero desde entonces se ha vuelto

muy popular en la industria y ahora está bajo los auspicios de la Apache Software Foundation. El flujo de aire se escribe en Python, los flujos de trabajo del flujo de aire se crean a través de scripts Python y el flujo de aire está diseñado según el principio de "configuración como código". Muchos usuarios de flujo de aire empresarial ahora ejecutan el flujo de aire sobre Kubernetes.

Gráficos de Acíclicos dirigidos (DAG)

En el flujo de aire, los flujos de trabajo se denominan gráficos Acíclicos dirigidos (DAG). Los DAG se componen de tareas que se ejecutan en secuencia, en paralelo o en una combinación de las dos, dependiendo de la definición DAG. El programador de flujo de aire ejecuta tareas individuales en una matriz de trabajadores y cumple con las dependencias a nivel de tarea especificadas en la definición DAG. Los DAG se definen y crean a través de scripts Python.

ONTAP 9 de NetApp

ONTAP 9 de NetApp es la última generación del software de gestión del almacenamiento de NetApp y permite a empresas como la suya modernizar su infraestructura y realizar la transición a un centro de datos preparado para el cloud. Gracias a las capacidades de gestión de datos líderes del sector, ONTAP le permite gestionar y proteger sus datos con un solo conjunto de herramientas sin importar dónde residan. También puede mover los datos libremente a donde necesite: El extremo, el núcleo o el cloud. ONTAP 9 incluye numerosas funciones que simplifican la gestión de datos, aceleran y protegen sus datos esenciales y preparan su infraestructura para el futuro con arquitecturas de cloud híbrido.

Simplifique la gestión de los datos

La gestión de datos es vital para las operaciones TECNOLÓGICAS de su empresa, de modo que pueda usar los recursos adecuados para sus aplicaciones y conjuntos de datos. ONTAP incluye las siguientes funciones para facilitar y simplificar las operaciones, y reducir el coste total de funcionamiento:

- **Compactación de datos inline y deduplicación expandida.** la compactación de datos reduce el espacio perdido dentro de los bloques de almacenamiento, y la deduplicación aumenta significativamente la capacidad efectiva.
- **Calidad de servicio (QoS) mínima, máxima y adaptativa.** los controles granulares de QoS ayudan a mantener los niveles de rendimiento para aplicaciones críticas en entornos altamente compartidos.
- **ONTAP FabricPool.** esta función ofrece una organización automática en niveles de datos inactivos para opciones de almacenamiento en cloud público y privado, incluidos Amazon Web Services (AWS), Azure y almacenamiento basado en objetos StorageGRID de NetApp.

Acelere y proteja sus datos

ONTAP no solo ofrece niveles de rendimiento y protección de datos superiores, sino que amplía estas funcionalidades con las siguientes funciones:

- **Alto rendimiento y baja latencia.** ONTAP ofrece el rendimiento más alto posible con la menor latencia posible.
- **La tecnología ONTAP FlexGroup de NetApp.** Un volumen FlexGroup es un contenedor de datos de alto rendimiento que se puede escalar linealmente hasta 20 PB y 400 000 millones de archivos, lo que proporciona un espacio de nombres único que simplifica la gestión de datos.
- **Protección de datos.** ONTAP ofrece capacidades integradas de protección de datos con administración común en todas las plataformas.
- **Cifrado de volumen de NetApp.** ONTAP ofrece cifrado nativo a nivel de volumen con compatibilidad para la gestión de claves incorporada y externa.

Infraestructura preparada para futuros retos

ONTAP 9 le ayuda a satisfacer las exigentes y siempre cambiantes necesidades de su empresa:

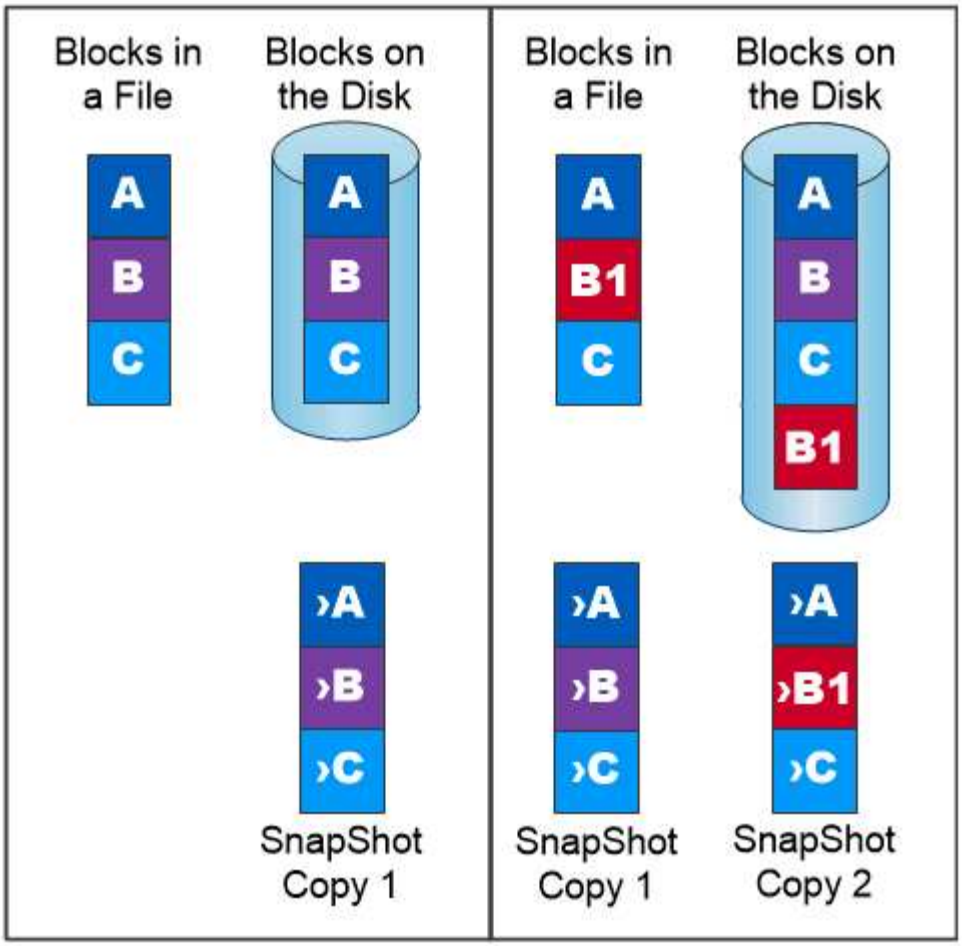
- **Escalado sencillo y operaciones no disruptivas.** ONTAP admite la adición sin interrupciones de capacidad a las controladoras existentes y a los clústeres de escalado horizontal. Puede empezar a utilizar tecnologías punteras como NVMe y FC 32 GB, sin necesidad de realizar costosas migraciones de datos y sin cortes.
- **Conexión al cloud.** ONTAP es uno de los programas de gestión del almacenamiento con mejor conexión al cloud e incluye opciones de almacenamiento definido por software (ONTAP Select) e instancias nativas del cloud (Cloud Volumes Service de NetApp) en todos los clouds públicos.
- **Integración con aplicaciones emergentes.** al utilizar la misma infraestructura que soporta las aplicaciones empresariales existentes, ONTAP ofrece servicios de datos de clase empresarial para plataformas y aplicaciones de próxima generación como OpenStack, Hadoop y MongoDB.

Copias Snapshot de NetApp

Una copia Snapshot de NetApp es una imagen puntual de solo lectura de un volumen. La imagen consume un espacio de almacenamiento mínimo y tiene una sobrecarga del rendimiento mínima, ya que solo registra los cambios que se han realizado en los archivos creados desde que se realizó la última copia Snapshot, como se muestra en la siguiente figura.

Las copias Snapshot deben su eficiencia a la tecnología de virtualización del almacenamiento central de ONTAP, el sistema de archivos de escritura en cualquier lugar (WAFL). Al igual que una base de datos, WAFL utiliza metadatos para apuntar a los bloques de datos reales en el disco. Sin embargo, a diferencia de una base de datos, WAFL no sobrescribe los bloques existentes. Escribe los datos actualizados en un bloque nuevo y cambia los metadatos. Porque ONTAP hace referencia a los metadatos cuando crea una copia Snapshot, en lugar de copiar bloques de datos, es tan eficiente que las copias Snapshot. Al hacerlo, se elimina el tiempo de búsqueda que otros sistemas incurren en la localización de los bloques a copiar, así como el costo de hacer la copia misma.

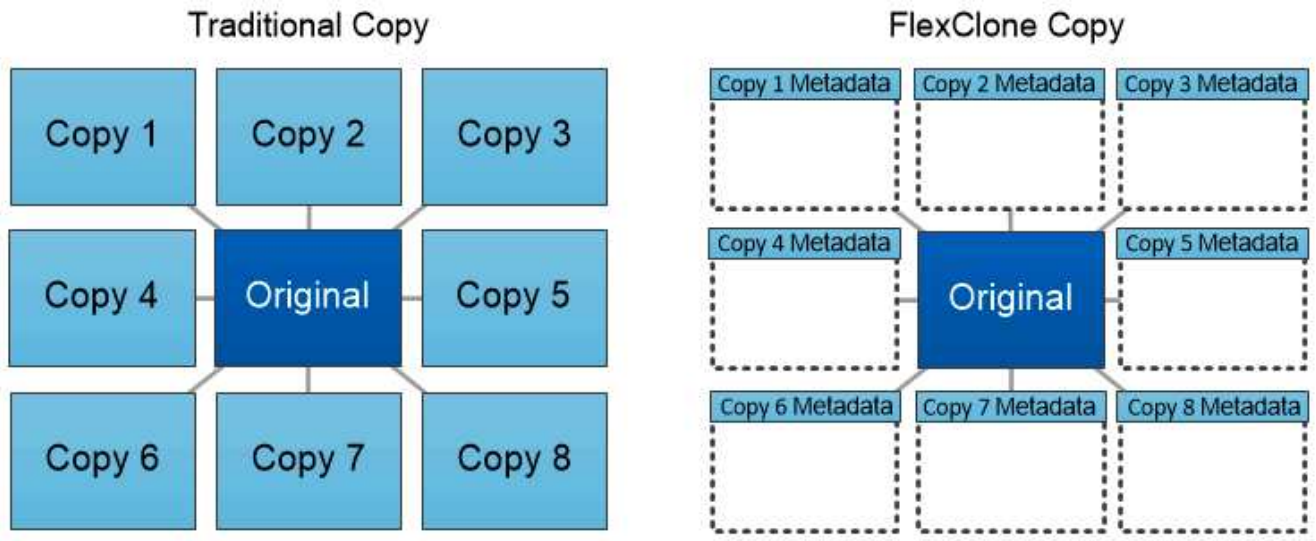
Puede utilizar una copia Snapshot para recuperar archivos o LUN individuales o para restaurar el contenido completo de un volumen. ONTAP compara la información de punteros de la copia Snapshot con los datos del disco para reconstruir el objeto faltante o dañado, sin tiempo de inactividad ni un coste de rendimiento significativo.



A Snapshot copy records only changes to the active file system since the last Snapshot copy.

Tecnología FlexClone de NetApp

La tecnología FlexClone de NetApp hace referencia a los metadatos de Snapshot para crear copias puntuales editables de un volumen. Las copias comparten bloques de datos con sus padres, sin consumir almacenamiento excepto lo que se necesita para los metadatos hasta que se escriben los cambios en la copia, como se muestra en la siguiente figura. Cuando se pueden crear copias tradicionales en minutos o incluso horas, el software FlexClone le permite copiar incluso los conjuntos de datos más grandes de forma casi instantánea. Esto lo convierte en la opción ideal para las situaciones en las que necesita varias copias de conjuntos de datos idénticos (un espacio de trabajo de desarrollo, por ejemplo) o copias temporales de un conjunto de datos (probar una aplicación contra un conjunto de datos de producción).

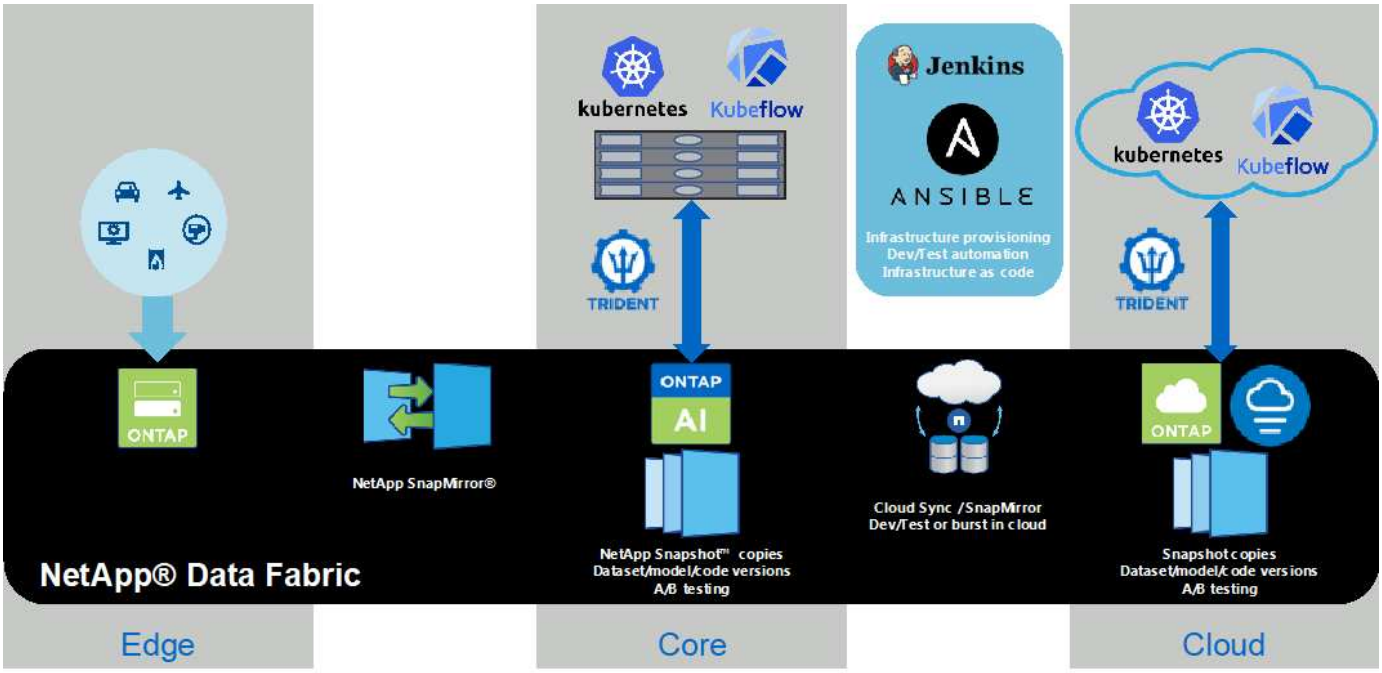


FlexClone copies share data blocks with their parents, consuming no storage except what is required for metadata.

Tecnología de replicación de datos de SnapMirror de NetApp

El software SnapMirror de NetApp es una solución de replicación unificada rentable y fácil de usar para todo Data Fabric. Replica datos a altas velocidades mediante LAN o WAN. Le proporciona una alta disponibilidad de datos y una rápida replicación de datos para todo tipo de aplicaciones, incluidas aplicaciones vitales para el negocio en entornos tanto virtuales como tradicionales. Al replicar datos en uno o varios sistemas de almacenamiento de NetApp y actualizar continuamente los datos secundarios, estos están siempre al día y disponibles cuando los necesite. No se requieren servidores de replicación externos. Consulte la figura siguiente para ver un ejemplo de una arquitectura que aprovecha la tecnología SnapMirror.

El software SnapMirror aprovecha las eficiencias del almacenamiento de ONTAP de NetApp y envía únicamente los bloques cambiados a través de la red. El software SnapMirror también usa la compresión de red incorporada para acelerar las transferencias de datos y reducir la utilización de ancho de banda hasta un 70 %. Con la tecnología SnapMirror, puede aprovechar un flujo de datos de thin replication para crear un único almacén que mantenga los reflejos activos y las copias de momentos específicos anteriores, lo que reduce el tráfico de red hasta un 50 %.



Copia y sincronización de NetApp BlueXP

La copia y sincronización de BlueXP es un servicio de NetApp que ofrece una sincronización de datos rápida y segura. Ya tenga que transferir archivos entre recursos compartidos de archivos NFS o SMB en las instalaciones, NetApp StorageGRID, NetApp ONTAP S3, NetApp Cloud Volumes Service, Azure NetApp Files, AWS S3, AWS EFS, Azure Blob, Google Cloud Storage, o IBM Cloud Object Storage, BlueXP Copy and Sync mueve los archivos a donde los necesitas de forma rápida y segura.

Una vez transferidos los datos, estarán completamente disponibles para su uso tanto en origen como en destino. BlueXP Copy and Sync puede sincronizar los datos bajo demanda al activar una actualización o sincronizar continuamente los datos en función de una programación predefinida. Independientemente de ello, BlueXP Copy y Sync solo mueve los diferenciales, por lo que se reducen al mínimo el tiempo y el dinero que se invierten en la replicación de datos.

Copia y sincronización de BlueXP es una herramienta de software como servicio (SaaS) extremadamente sencilla de configurar y usar. Las transferencias de datos activadas por BlueXP Copy and Sync se llevan a cabo por agentes de datos. Los agentes de datos de BlueXP Copy y Sync se pueden poner en marcha en AWS, Azure, Google Cloud Platform o en las instalaciones.

XCP de NetApp

XCP de NetApp es el software basado en cliente para migraciones de datos y análisis del sistema de archivos entre NetApp y NetApp. XCP se ha diseñado para escalar y lograr el máximo rendimiento utilizando todos los recursos del sistema disponibles para gestionar conjuntos de datos de gran volumen y migraciones de alto rendimiento. XCP le ayuda a obtener una visibilidad completa del sistema de archivos con la opción de generar informes.

XCP de NetApp está disponible en un único paquete compatible con los protocolos NFS y SMB. XCP incluye un binario de Linux para conjuntos de datos NFS y un ejecutable de Windows para conjuntos de datos SMB.

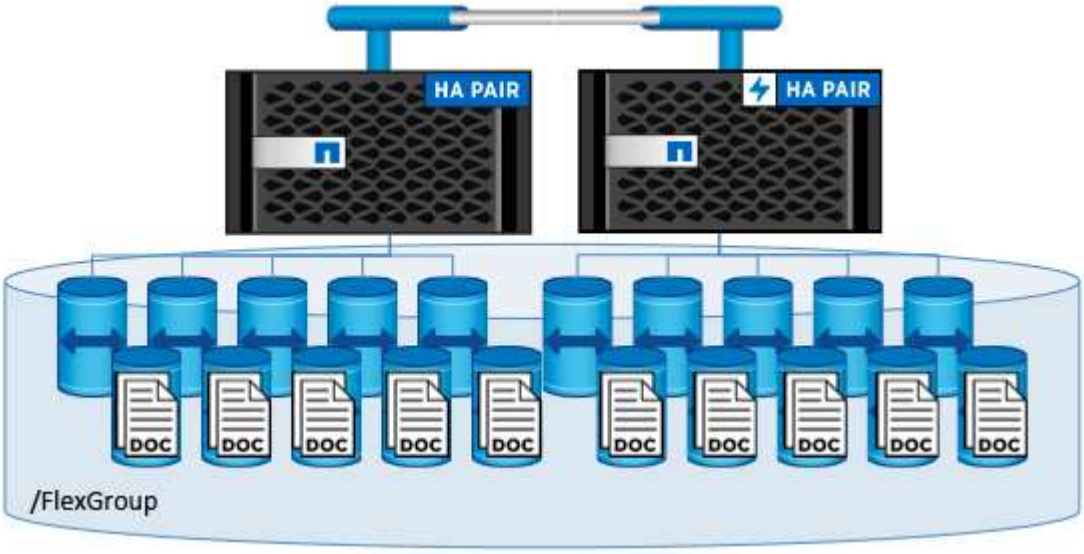
XCP File Analytics de NetApp es un software basado en host que detecta recursos compartidos de archivos, ejecuta análisis en el sistema de archivos y proporciona una consola para el análisis de archivos. XCP File Analytics es compatible con los sistemas NetApp y de otros proveedores, y se ejecuta en hosts Linux o Windows para proporcionar análisis en sistemas de archivos NFS y exportados SMB.

ONTAP FlexGroup Volumes de NetApp

Un conjunto de datos de entrenamiento puede ser una colección con hasta miles de millones de archivos. Pueden ser archivos de texto, de audio, de vídeo o cualquier otra forma de datos no estructurados que deban almacenarse y procesarse para su lectura en paralelo. El sistema de almacenamiento debe almacenar un gran número de archivos pequeños y debe leerlos en paralelo, con una entrada y salida secuencial o aleatoria

Un volumen FlexGroup es un espacio de nombres único que comprende varios volúmenes miembro constituyentes, tal y como se muestra en la siguiente figura. Desde el punto de vista de un administrador de almacenamiento, un volumen FlexGroup se gestiona y actúa como un volumen FlexVol de NetApp. Los archivos de un volumen de FlexGroup se asignan a volúmenes miembro individuales y no están repartidos en volúmenes o nodos. Ofrecen las siguientes capacidades:

- Los volúmenes FlexGroup proporcionan varios petabytes de capacidad y una baja latencia predecible para cargas de trabajo con una gran cantidad de metadatos.
- Permiten un máximo de 400 000 millones de archivos en un mismo espacio de nombres.
- Admiten operaciones en paralelo para cargas de trabajo NAS entre varias CPU, nodos, agregados y volúmenes FlexVol constituyentes.



Requisitos de hardware y software

La solución de plano de control de IA de NetApp no depende de este hardware específico. La solución es compatible con cualquier dispositivo de almacenamiento físico, instancia definida por software o servicio cloud de NetApp compatible con Trident. Entre los ejemplos se incluyen un sistema de almacenamiento AFF de NetApp, Azure NetApp Files, Cloud Volumes Service de NetApp, una instancia de almacenamiento definido por software ONTAP Select de NetApp o una instancia de Cloud Volumes ONTAP de NetApp. Además, la solución puede implementarse en cualquier clúster de Kubernetes, siempre que la versión de Kubernetes utilizada sea compatible con Kubeflow y Trident de NetApp. Si desea ver una lista de las versiones de Kubernetes compatibles con ubeflow, consulte la "Documentación oficial de Kubeflow". Si desea ver una lista de las versiones de Kubernetes compatibles con Trident, consulte "Documentación de Trident". Consulte

las siguientes tablas para obtener información detallada sobre el entorno que se utilizó para validar la solución.

Componente de la infraestructura	Cantidad	Detalles	Sistema operativo
Host de salto de implementación	1	MÁQUINA VIRTUAL	Sistema operativo Ubuntu 20.04.2 LTS
Nodos maestros de Kubernetes	1	MÁQUINA VIRTUAL	Sistema operativo Ubuntu 20.04.2 LTS
Nodos de trabajo de Kubernetes	2	MÁQUINA VIRTUAL	Sistema operativo Ubuntu 20.04.2 LTS
Nodos de trabajo de GPU de Kubernetes	2	NVIDIA DGX-1 (nativo)	NVIDIA DGX OS 4.0.5 (basado en Ubuntu 18.04.2 LTS)
Reducida	1 par de alta disponibilidad	AFF A220 de NetApp	ONTAP 9.7 P6 de NetApp

Componente de software	Versión
Flujo de aire Apache	2.0.1
Tabla de Helm para flujo de aire de Apache	8.0.8
Docker	19.03.12
Kubeflow	1.2
Kubernetes	1.18.9
Trident de NetApp	21.01.2
DeepOps de NVIDIA	Funcionalidad de implementación de Trident desde la rama maestra hasta el encargo "61898cdfda"; Todas las demás funciones de la versión 21.03

Soporte técnico

NetApp no ofrece compatibilidad empresarial con Apache Airflow, Docker, Kubeflow, Kubernetes u NVIDIA DeepOps. Si está interesado en una solución totalmente compatible con funcionalidades similares a la solución del plano de control de IA de NetApp, "Póngase en contacto con NetApp" Acerca de soluciones de IA/ML totalmente compatibles que NetApp ofrece en colaboración con partners.

Puesta en marcha de Kubernetes

En esta sección se describen las tareas que debe completar para poner en marcha un clúster de Kubernetes en el que desea implementar la solución de plano de control de IA de NetApp. Si ya tiene un clúster de Kubernetes, puede omitir esta sección siempre que ejecute una versión de Kubernetes compatible con Kubeflow y NetApp Trident. Si desea ver una lista de las versiones de Kubernetes compatibles con ubeflow, consulte la "Documentación oficial de Kubeflow". Si desea ver una lista de las versiones de Kubernetes compatibles con Trident, consulte "Documentación de Trident".

En el caso de puestas en marcha de Kubernetes en las instalaciones que incorporan nodos de reconstrucción completa con GPU de NVIDIA, NetApp recomienda usar la herramienta de puesta en marcha de Kubernetes DeepOps de NVIDIA. En esta sección, se describe la puesta en marcha de un clúster de Kubernetes con DeepOps.

Requisitos previos

Antes de realizar el ejercicio de implementación descrito en esta sección, asumimos que ya ha realizado las siguientes tareas:

1. Ya ha configurado todos los nodos Kubernetes con configuración básica (por ejemplo, un sistema NVIDIA DGX que forma parte de un pod ONTAP AI), de acuerdo con instrucciones de configuración estándar.
2. Ha instalado un sistema operativo compatible en todos los nodos maestro y de trabajo de Kubernetes y en un host de salto de implementación. Para obtener una lista de los sistemas operativos compatibles con DeepOps, consulte "Sitio DeepOps GitHub".

Utilice NVIDIA DeepOps para instalar y configurar Kubernetes

Para poner en marcha y configurar su clúster de Kubernetes con NVIDIA DeepOPS, realice las siguientes tareas desde un host de salto de implementación:

1. Descargue NVIDIA DeepOps siguiendo las instrucciones de "Página de Inicio" En el sitio de NVIDIA DeepOps GitHub.
2. Implemente Kubernetes en su clúster siguiendo las instrucciones que se indican en "Página de la Guía de implementación de Kubernetes" En el sitio de NVIDIA DeepOps GitHub.

Implementación y configuración de Trident de NetApp

Implementación y configuración de Trident de NetApp

En esta sección se describen las tareas que debe completar para instalar y configurar NetApp Trident en su clúster Kubernetes.

Requisitos previos

Antes de realizar el ejercicio de implementación descrito en esta sección, asumimos que ya ha realizado las siguientes tareas:

1. Ya existe un clúster de Kubernetes en funcionamiento y ejecuta una versión de Kubernetes compatible con Trident. Para obtener una lista de las versiones compatibles, consulte "Documentación de Trident".
2. Ya existe un dispositivo de almacenamiento de NetApp, una instancia definida por software o un servicio de almacenamiento en cloud que Trident admite.

Instale Trident

Para instalar y configurar NetApp Trident en su clúster de Kubernetes, realice las siguientes tareas desde el host de salto de implementación:

1. Implemente Trident mediante uno de los siguientes métodos:
    - Si utilizó NVIDIA DeepOps para poner en marcha su clúster de Kubernetes, también puede usar NVIDIA DeepOps para implementar Trident en el clúster de Kubernetes. Para poner en marcha Trident con DeepOps, siga "Instrucciones de puesta en marcha de Trident" En el sitio de NVIDIA DeepOps

GitHub.

- Si no utilizó NVIDIA DeepOPS para poner en marcha su clúster de Kubernetes, o si simplemente prefiere poner en marcha Trident manualmente, puede implementar Trident siguiendo el "instrucciones de puesta en funcionamiento" En la documentación de Trident. Asegúrese de crear al menos un back-end de Trident y al menos un Kubernetes StorageClass para obtener más información acerca de cómo configurar "Back-ends" y.. "Clases de almacenamiento" Consulte los subapartados vinculados en los documentos de NetApp.

> Si va a poner en marcha la solución del plano de control de IA de NetApp en un POD de IA de ONTAP, consulte "Ejemplo de Back-ends de Trident para puestas en marcha de ONTAP AI" Para algunos ejemplos de diferentes Back-ends de Trident que es posible que desee crear y. "Ejemplo de una historia Kubernetes para las puestas en marcha de IA de ONTAP" Puede que desee crear algunos ejemplos de diferentes clases de almacenamiento de Kubernetes.

Ejemplo de Back-ends de Trident para puestas en marcha de ONTAP AI

Antes de poder usar Trident para aprovisionar recursos de almacenamiento de forma dinámica dentro de su clúster de Kubernetes, debe crear uno o varios Back-ends de Trident. Los siguientes ejemplos representan diferentes tipos de backends que puede crear si va a implementar la solución de plano de control de IA de NetApp en un pod ONTAP AI. Para obtener más información acerca de backends, consulte "Documentación de Trident".

1. NetApp recomienda crear un Back-end de Trident habilitado para FlexGroup para cada LIF de datos (interfaz de red lógica que proporciona acceso a los datos) que desee utilizar en su sistema AFF de NetApp. Esto le permitirá equilibrar los montajes de volumen entre LIF

   Los siguientes comandos de ejemplo muestran la creación de dos Back-ends de Trident habilitados para FlexGroup para dos LIF de datos diferentes asociadas con la misma máquina virtual de almacenamiento (SVM) de ONTAP. Estas backends utilizan la `ontap-nas-flexgroup` controlador de almacenamiento. ONTAP admite dos tipos de volúmenes de datos principales: FlexVol y FlexGroup. Los volúmenes FlexVol tienen un tamaño limitado (a partir de la escritura, el tamaño máximo depende de la implementación específica). Por otro lado, los volúmenes FlexGroup se pueden escalar de forma lineal hasta 20 PB y 400 000 millones de archivos y, además, ofrecen un espacio de nombres único que simplifica enormemente la gestión de los datos. Por lo tanto, los volúmenes FlexGroup son óptimos para cargas de trabajo de IA y ML que dependen de grandes cantidades de datos.

   Si está trabajando con una pequeña cantidad de datos y desea usar volúmenes de FlexVol en lugar de volúmenes de FlexGroup, puede crear Back-ends de Trident que utilizan `ontap-nas` controlador de almacenamiento en lugar de `ontap-nas-flexgroup` controlador de almacenamiento.

```
$ cat << EOF > ./trident-backend-ontap-ai-flexgroups-iface1.json
{
    "version": 1,
    "storageDriverName": "ontap-nas-flexgroup",
    "backendName": "ontap-ai-flexgroups-iface1",
    "managementLIF": "10.61.218.100",
    "dataLIF": "192.168.11.11",
```

```
    "svm": "ontapai_nfs",
    "username": "admin",
    "password": "ontapai"
}
EOF
$ tridentctl create backend -f ./trident-backend-ontap-ai-flexgroups-
iface1.json -n trident
+----------------------------+---------------------
+--------------------------------------+--------+---------+
|            NAME            |   STORAGE DRIVER    |
UUID                 | STATE  | VOLUMES |
+----------------------------+---------------------
+--------------------------------------+--------+---------+
| ontap-ai-flexgroups-iface1 | ontap-nas-flexgroup | b74cbddb-e0b8-40b7-
b263-b6da6dec0bdd | online |       0 |
+----------------------------+---------------------
+--------------------------------------+--------+---------+
$ cat << EOF > ./trident-backend-ontap-ai-flexgroups-iface2.json
{
    "version": 1,
    "storageDriverName": "ontap-nas-flexgroup",
    "backendName": "ontap-ai-flexgroups-iface2",
    "managementLIF": "10.61.218.100",
    "dataLIF": "192.168.12.12",
    "svm": "ontapai_nfs",
    "username": "admin",
    "password": "ontapai"
}
EOF
$ tridentctl create backend -f ./trident-backend-ontap-ai-flexgroups-
iface2.json -n trident
+----------------------------+---------------------
+--------------------------------------+--------+---------+
|            NAME            |   STORAGE DRIVER    |
UUID                 | STATE  | VOLUMES |
+----------------------------+---------------------
+--------------------------------------+--------+---------+
| ontap-ai-flexgroups-iface2 | ontap-nas-flexgroup | 61814d48-c770-436b-
9cb4-cf7ee661274d | online |       0 |
+----------------------------+---------------------
+--------------------------------------+--------+---------+
$ tridentctl get backend -n trident
+----------------------------+---------------------
+--------------------------------------+--------+---------+
|            NAME            |   STORAGE DRIVER    |
UUID                 | STATE  | VOLUMES |
```

```
+----------------------------+---------------------
+-------------------------------------+--------+---------+
| ontap-ai-flexgroups-iface1 | ontap-nas-flexgroup | b74cbddb-e0b8-40b7-
b263-b6da6dec0bdd | online |       0 |
| ontap-ai-flexgroups-iface2 | ontap-nas-flexgroup | 61814d48-c770-436b-
9cb4-cf7ee661274d | online |       0 |
+----------------------------+---------------------
+-------------------------------------+--------+---------+
```

2. NetApp también recomienda crear uno o varios Back-ends de Trident habilitados para FlexVol. Si se utilizan FlexGroup Volumes para el entrenamiento del almacenamiento de conjuntos de datos, es posible que se desee usar volúmenes de FlexVol para almacenar resultados, resultados, información de depuración, etc. Si se desean usar volúmenes de FlexVol, se deben crear uno o varios Back-ends de Trident habilitados para FlexVol. Los siguientes comandos de ejemplo muestran la creación de una única estructura de fondo Trident habilitada para FlexVol que utiliza una sola LIF de datos.

```
$ cat << EOF > ./trident-backend-ontap-ai-flexvols.json
{
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "ontap-ai-flexvols",
    "managementLIF": "10.61.218.100",
    "dataLIF": "192.168.11.11",
    "svm": "ontapai_nfs",
    "username": "admin",
    "password": "ontapai"
}
EOF
$ tridentctl create backend -f ./trident-backend-ontap-ai-flexvols.json -n
trident
+----------------------------+---------------------
+--------------------------------------+--------+---------+
|            NAME            |   STORAGE DRIVER    |                 UUID
| STATE  | VOLUMES |
+----------------------------+---------------------
+--------------------------------------+--------+---------+
| ontap-ai-flexvols          | ontap-nas           | 52bdb3b1-13a5-4513-
a9c1-52a69657fabe | online |       0 |
+----------------------------+---------------------
+--------------------------------------+--------+---------+
$ tridentctl get backend -n trident
+----------------------------+---------------------
+--------------------------------------+--------+---------+
|            NAME            |   STORAGE DRIVER    |                 UUID
| STATE  | VOLUMES |
+----------------------------+---------------------
+--------------------------------------+--------+---------+
| ontap-ai-flexvols          | ontap-nas           | 52bdb3b1-13a5-4513-
a9c1-52a69657fabe | online |       0 |
| ontap-ai-flexgroups-iface1 | ontap-nas-flexgroup | b74cbddb-e0b8-40b7-
b263-b6da6dec0bdd | online |       0 |
| ontap-ai-flexgroups-iface2 | ontap-nas-flexgroup | 61814d48-c770-436b-
9cb4-cf7ee661274d | online |       0 |
+----------------------------+---------------------
+--------------------------------------+--------+---------+
```

Ejemplo de clases de almacenamiento Kubernetes para puestas en marcha de IA en ONTAP

Antes de poder usar Trident para aprovisionar recursos de almacenamiento de forma dinámica dentro del clúster de Kubernetes, debe crear una o varias clases de almacenamiento de Kubernetes. Los siguientes ejemplos representan diferentes tipos de

clases de almacenamiento que puede que desee crear si pone en marcha la solución de plano de control de IA de NetApp en un pool de IA de ONTAP. Si desea obtener más información sobre las clases de almacenamiento, consulte "Documentación de Trident".

1. NetApp recomienda crear un tipo de almacenamiento aparte para cada back-end de Trident habilitado para FlexGroup que haya creado en la sección "Ejemplo de Back-ends de Trident para puestas en marcha de ONTAP AI", paso 1. Estas clases de almacenamiento granular le permiten añadir montajes NFS que corresponden a LIF específicas (los LIF especificados cuando se crearon las back-ends de Trident) como un back-end concreto especificado en el archivo SPEC de StorageClass. Los comandos de ejemplo siguientes muestran la creación de dos StorageClasses que corresponden a los dos backends de ejemplo que se crearon en la sección "Ejemplo de Back-ends de Trident para puestas en marcha de ONTAP AI", paso 1. Si desea obtener más información sobre las clases de almacenamiento, consulte "Documentación de Trident".

   Para que no se elimine un volumen persistente cuando se elimine la reclamación de volumen persistente (RVP) correspondiente, en el siguiente ejemplo se utiliza un `reclaimPolicy` valor de `Retain`. Para obtener más información acerca de `reclaimPolicy` consulte el funcionario "Documentación de Kubernetes".

```
$ cat << EOF > ./storage-class-ontap-ai-flexgroups-retain-iface1.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-ai-flexgroups-retain-iface1
provisioner: netapp.io/trident
parameters:
  backendType: "ontap-nas-flexgroup"
  storagePools: "ontap-ai-flexgroups-iface1:.*"
reclaimPolicy: Retain
EOF
$ kubectl create -f ./storage-class-ontap-ai-flexgroups-retain-
iface1.yaml
storageclass.storage.k8s.io/ontap-ai-flexgroups-retain-iface1 created
$ cat << EOF > ./storage-class-ontap-ai-flexgroups-retain-iface2.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-ai-flexgroups-retain-iface2
provisioner: netapp.io/trident
parameters:
  backendType: "ontap-nas-flexgroup"
  storagePools: "ontap-ai-flexgroups-iface2:.*"
reclaimPolicy: Retain
EOF
$ kubectl create -f ./storage-class-ontap-ai-flexgroups-retain-
iface2.yaml
storageclass.storage.k8s.io/ontap-ai-flexgroups-retain-iface2 created
$ kubectl get storageclass
NAME                                PROVISIONER         AGE
ontap-ai-flexgroups-retain-iface1   netapp.io/trident   0m
ontap-ai-flexgroups-retain-iface2   netapp.io/trident   0m
```

2. NetApp también recomienda crear un StorageClass que se corresponda con el back-end Trident habilitado para FlexVol que ha creado en la sección "Ejemplo de Back-ends de Trident para puestas en marcha de ONTAP AI", paso 2. Los comandos de ejemplo siguientes muestran la creación de un solo tipo de almacenamiento para volúmenes FlexVol.

   En el siguiente ejemplo, no se ha especificado un back-end determinado en el archivo de definición StorageClass porque solo se creó un back-end Trident habilitado para FlexVol. Cuando se usa Kubernetes para administrar volúmenes que usan este clase de almacenamiento, Trident intenta usar cualquier back-end disponible que utilice `ontap-nas` controlador.

```
$ cat << EOF > ./storage-class-ontap-ai-flexvols-retain.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-ai-flexvols-retain
provisioner: netapp.io/trident
parameters:
  backendType: "ontap-nas"
reclaimPolicy: Retain
EOF
$ kubectl create -f ./storage-class-ontap-ai-flexvols-retain.yaml
storageclass.storage.k8s.io/ontap-ai-flexvols-retain created
$ kubectl get storageclass
NAME                                PROVISIONER         AGE
ontap-ai-flexgroups-retain-iface1   netapp.io/trident   1m
ontap-ai-flexgroups-retain-iface2   netapp.io/trident   1m
ontap-ai-flexvols-retain            netapp.io/trident   0m
```

3. NetApp también recomienda crear un tipo de almacenamiento genérico para volúmenes FlexGroup. En los siguientes comandos de ejemplo, se muestra la creación de un solo tipo de almacenamiento genérico para volúmenes FlexGroup.

   Tenga en cuenta que no se ha especificado un back-end determinado en el archivo de definición StorageClass. Por lo tanto, cuando se usa Kubernetes para administrar volúmenes que usan esta clase de almacenamiento, Trident intenta usar cualquier back-end disponible que utilice `ontap-nas-flexgroup` controlador.

```
$ cat << EOF > ./storage-class-ontap-ai-flexgroups-retain.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-ai-flexgroups-retain
provisioner: netapp.io/trident
parameters:
  backendType: "ontap-nas-flexgroup"
reclaimPolicy: Retain
EOF
$ kubectl create -f ./storage-class-ontap-ai-flexgroups-retain.yaml
storageclass.storage.k8s.io/ontap-ai-flexgroups-retain created
$ kubectl get storageclass
NAME                                PROVISIONER         AGE
ontap-ai-flexgroups-retain          netapp.io/trident   0m
ontap-ai-flexgroups-retain-iface1   netapp.io/trident   2m
ontap-ai-flexgroups-retain-iface2   netapp.io/trident   2m
ontap-ai-flexvols-retain            netapp.io/trident   1m
```

Despliegue de Kubeflow

En esta sección se describen las tareas que debe completar para poner en marcha Kubeflow en su clúster de Kubernetes.

Requisitos previos

Antes de realizar el ejercicio de implementación descrito en esta sección, asumimos que ya ha realizado las siguientes tareas:

1. Ya tiene un clúster de Kubernetes en funcionamiento y ejecuta una versión de Kubernetes que admite ubeflow. Para obtener una lista de las versiones compatibles, consulte "Documentación oficial de Kubeflow".
2. Ya ha instalado y configurado NetApp Trident en su clúster de Kubernetes, como se indica en "Implementación y configuración de Trident".

Establezca el tipo de almacenamiento de Kubernetes predeterminado

Antes de poner en marcha Kubeflow, debe designar un clase de almacenamiento predeterminado dentro del clúster de Kubernetes. El proceso de implementación de Kubeflow intenta aprovisionar nuevos volúmenes persistentes mediante el tipo de almacenamiento predeterminado. Si no se designa StorageClass como clase de almacenamiento predeterminado, la implementación falla. Para designar un StorageClass predeterminado en el clúster, realice la siguiente tarea desde el host de salto de implementación. Si ya ha designado un tipo de almacenamiento predeterminado en el clúster, puede omitir este paso.

1. Designe una de las clases de almacenamiento existentes como clase de almacenamiento predeterminada. Los comandos de ejemplo siguientes muestran la designación de un StorageClass llamado `ontap-ai-flexvols-retain` Como el tipo de almacenamiento predeterminado.

La `ontap-nas-flexgroup` El tipo de backend de Trident tiene un tamaño de RVP mínimo que es bastante grande. De manera predeterminada, Kubeflow intenta suministrar EVs que son sólo unos pocos GBS en tamaño. Por lo tanto, no debe designar un StorageClass que utilice `ontap-nas-flexgroup` Tipo back-end como StorageClass predeterminado para la implementación de Kubeflow.

```
$ kubectl get sc
NAME                                PROVISIONER             AGE
ontap-ai-flexgroups-retain          csi.trident.netapp.io   25h
ontap-ai-flexgroups-retain-iface1   csi.trident.netapp.io   25h
ontap-ai-flexgroups-retain-iface2   csi.trident.netapp.io   25h
ontap-ai-flexvols-retain            csi.trident.netapp.io   3s
$ kubectl patch storageclass ontap-ai-flexvols-retain -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
storageclass.storage.k8s.io/ontap-ai-flexvols-retain patched
$ kubectl get sc
NAME                                 PROVISIONER             AGE
ontap-ai-flexgroups-retain           csi.trident.netapp.io   25h
ontap-ai-flexgroups-retain-iface1    csi.trident.netapp.io   25h
ontap-ai-flexgroups-retain-iface2    csi.trident.netapp.io   25h
ontap-ai-flexvols-retain (default)   csi.trident.netapp.io   54s
```

Utilice NVIDIA DeepOps para poner en marcha Kubeflow

NetApp recomienda usar la herramienta de puesta en marcha de Kubeflow que proporciona NVIDIA DeepOps. Para poner en marcha Kubeflow en su clúster de Kubernetes con la herramienta de puesta en marcha DeepOps, siga estas tareas desde el host de salto de implementación.

Como alternativa, puede implementar Kubeflow manualmente siguiendo la "instrucciones de instalación" En la documentación oficial de Kubeflow

1. Implemente Kubeflow en su clúster siguiendo el "Instrucciones de despliegue de Kubeflow" En el sitio de NVIDIA DeepOps GitHub.
2. Tenga en cuenta la URL del panel de Kubeflow que genera la herramienta de puesta en marcha de DeepOps Kubeflow.

```
$ ./scripts/k8s/deploy_kubeflow.sh -x
…
INFO[0007] Applied the configuration Successfully!
filename="cmd/apply.go:72"
Kubeflow app installed to: /home/ai/kubeflow
It may take several minutes for all services to start. Run 'kubectl get
pods -n kubeflow' to verify
To remove (excluding CRDs, istio, auth, and cert-manager), run:
./scripts/k8s_deploy_kubeflow.sh -d
To perform a full uninstall : ./scripts/k8s_deploy_kubeflow.sh -D
Kubeflow Dashboard (HTTP NodePort): http://10.61.188.111:31380
```

3. Confirmar que todos los POD implementados en el espacio de nombres Kubeflow muestran un `STATUS` de `Running` y confirmar que ningún componente puesto en marcha en el espacio de nombres se encuentra en un estado de error. El inicio de todos los pods puede tardar varios minutos.

```
$ kubectl get all -n kubeflow
NAME                                                          READY
STATUS    RESTARTS   AGE
pod/admission-webhook-bootstrap-stateful-set-0                1/1
Running   0          95s
pod/admission-webhook-deployment-6b89c84c98-vrtbh             1/1
Running   0          91s
pod/application-controller-stateful-set-0                     1/1
Running   0          98s
pod/argo-ui-5dcf5d8b4f-m2wn4                                  1/1
Running   0          97s
pod/centraldashboard-cf4874ddc-7hcr8                          1/1
Running   0          97s
pod/jupyter-web-app-deployment-685b455447-gjhh7               1/1
Running   0          96s
pod/katib-controller-88c97d85c-kgq66                          1/1
Running   1          95s
pod/katib-db-8598468fd8-5jw2c                                 1/1
Running   0          95s
pod/katib-manager-574c8c67f9-wtrf5                            1/1
Running   1          95s
pod/katib-manager-rest-778857c989-fjbzn                       1/1
Running   0          95s
pod/katib-suggestion-bayesianoptimization-65df4d7455-qthmw    1/1
Running   0          94s
pod/katib-suggestion-grid-56bf69f597-98vwn                    1/1
Running   0          94s
pod/katib-suggestion-hyperband-7777b76cb9-9v6dq               1/1
Running   0          93s
```

```
pod/katib-suggestion-nasrl-77f6f9458c-2qzxq                    1/1
Running   0          93s
pod/katib-suggestion-random-77b88b5c79-l64j9                   1/1
Running   0          93s
pod/katib-ui-7587c5b967-nd629                                  1/1
Running   0          95s
pod/metacontroller-0                                           1/1
Running   0          96s
pod/metadata-db-5dd459cc-swzkm                                 1/1
Running   0          94s
pod/metadata-deployment-6cf77db994-69fk7                       1/1
Running   3          93s
pod/metadata-deployment-6cf77db994-mpbjt                       1/1
Running   3          93s
pod/metadata-deployment-6cf77db994-xg7tz                       1/1
Running   3          94s
pod/metadata-ui-78f5b59b56-qb6kr                               1/1
Running   0          94s
pod/minio-758b769d67-llvdr                                     1/1
Running   0          91s
pod/ml-pipeline-5875b9db95-g8t2k                               1/1
Running   0          91s
pod/ml-pipeline-persistenceagent-9b69ddd46-bt9r9               1/1
Running   0          90s
pod/ml-pipeline-scheduledworkflow-7b8d756c76-7x56s             1/1
Running   0          90s
pod/ml-pipeline-ui-79ffd9c76-fcwpd                             1/1
Running   0          90s
pod/ml-pipeline-viewer-controller-deployment-5fdc87f58-b2t9r   1/1
Running   0          90s
pod/mysql-657f87857d-l5k9z                                     1/1
Running   0          91s
pod/notebook-controller-deployment-56b4f59bbf-8bvnr            1/1
Running   0          92s
pod/profiles-deployment-6bc745947-mrdkh                        2/2
Running   0          90s
pod/pytorch-operator-77c97f4879-hmlrv                          1/1
Running   0          92s
pod/seldon-operator-controller-manager-0                       1/1
Running   1          91s
pod/spartakus-volunteer-5fdfddb779-l7qkm                       1/1
Running   0          92s
pod/tensorboard-6544748d94-nh8b2                               1/1
Running   0          92s
pod/tf-job-dashboard-56f79c59dd-6w59t                          1/1
Running   0          92s
```

```
pod/tf-job-operator-79cbfd6dbc-rb58c                          1/1
Running   0          91s
pod/workflow-controller-db644d554-cwrnb                       1/1
Running   0          97s
NAME                                                     TYPE
CLUSTER-IP      EXTERNAL-IP   PORT(S)             AGE
service/admission-webhook-service                        ClusterIP
10.233.51.169   <none>        443/TCP             97s
service/application-controller-service                   ClusterIP
10.233.4.54     <none>        443/TCP             98s
service/argo-ui                                          NodePort
10.233.47.191   <none>        80:31799/TCP        97s
service/centraldashboard                                 ClusterIP
10.233.8.36     <none>        80/TCP              97s
service/jupyter-web-app-service                          ClusterIP
10.233.1.42     <none>        80/TCP              97s
service/katib-controller                                 ClusterIP
10.233.25.226   <none>        443/TCP             96s
service/katib-db                                         ClusterIP
10.233.33.151   <none>        3306/TCP            97s
service/katib-manager                                    ClusterIP
10.233.46.239   <none>        6789/TCP            96s
service/katib-manager-rest                               ClusterIP
10.233.55.32    <none>        80/TCP              96s
service/katib-suggestion-bayesianoptimization            ClusterIP
10.233.49.191   <none>        6789/TCP            95s
service/katib-suggestion-grid                            ClusterIP
10.233.9.105    <none>        6789/TCP            95s
service/katib-suggestion-hyperband                       ClusterIP
10.233.22.2     <none>        6789/TCP            95s
service/katib-suggestion-nasrl                           ClusterIP
10.233.63.73    <none>        6789/TCP            95s
service/katib-suggestion-random                          ClusterIP
10.233.57.210   <none>        6789/TCP            95s
service/katib-ui                                         ClusterIP
10.233.6.116    <none>        80/TCP              96s
service/metadata-db                                      ClusterIP
10.233.31.2     <none>        3306/TCP            96s
service/metadata-service                                 ClusterIP
10.233.27.104   <none>        8080/TCP            96s
service/metadata-ui                                      ClusterIP
10.233.57.177   <none>        80/TCP              96s
service/minio-service                                    ClusterIP
10.233.44.90    <none>        9000/TCP            94s
service/ml-pipeline                                      ClusterIP
10.233.41.201   <none>        8888/TCP,8887/TCP   94s
```

```
service/ml-pipeline-tensorboard-ui                   ClusterIP
10.233.36.207   <none>        80/TCP              93s
service/ml-pipeline-ui                               ClusterIP
10.233.61.150   <none>        80/TCP              93s
service/mysql                                        ClusterIP
10.233.55.117   <none>        3306/TCP            94s
service/notebook-controller-service                  ClusterIP
10.233.10.166   <none>        443/TCP             95s
service/profiles-kfam                                ClusterIP
10.233.33.79    <none>        8081/TCP            92s
service/pytorch-operator                             ClusterIP
10.233.37.112   <none>        8443/TCP            95s
service/seldon-operator-controller-manager-service   ClusterIP
10.233.30.178   <none>        443/TCP             92s
service/tensorboard                                  ClusterIP
10.233.58.151   <none>        9000/TCP            94s
service/tf-job-dashboard                             ClusterIP
10.233.4.17     <none>        80/TCP              94s
service/tf-job-operator                              ClusterIP
10.233.60.32    <none>        8443/TCP            94s
service/webhook-server-service                       ClusterIP
10.233.32.167   <none>        443/TCP             87s
NAME                                                      READY   UP-
TO-DATE   AVAILABLE   AGE
deployment.apps/admission-webhook-deployment              1/1     1
1           97s
deployment.apps/argo-ui                                   1/1     1
1           97s
deployment.apps/centraldashboard                          1/1     1
1           97s
deployment.apps/jupyter-web-app-deployment                1/1     1
1           97s
deployment.apps/katib-controller                          1/1     1
1           96s
deployment.apps/katib-db                                  1/1     1
1           97s
deployment.apps/katib-manager                             1/1     1
1           96s
deployment.apps/katib-manager-rest                        1/1     1
1           96s
deployment.apps/katib-suggestion-bayesianoptimization     1/1     1
1           95s
deployment.apps/katib-suggestion-grid                     1/1     1
1           95s
deployment.apps/katib-suggestion-hyperband                1/1     1
1           95s
```

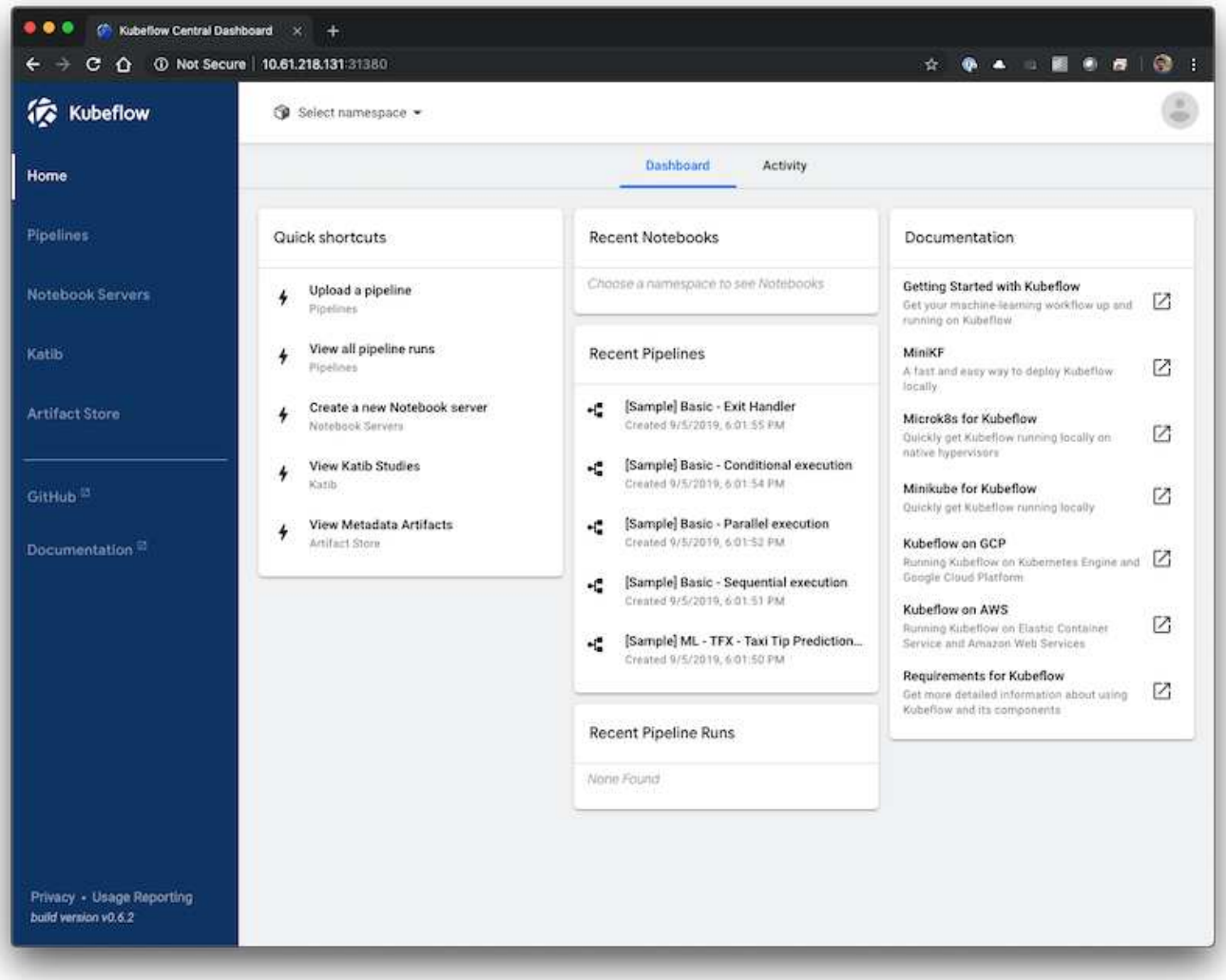
```
deployment.apps/katib-suggestion-nasrl                     1/1     1
1           95s
deployment.apps/katib-suggestion-random                    1/1     1
1           95s
deployment.apps/katib-ui                                   1/1     1
1           96s
deployment.apps/metadata-db                                1/1     1
1           96s
deployment.apps/metadata-deployment                        3/3     3
3           96s
deployment.apps/metadata-ui                                1/1     1
1           96s
deployment.apps/minio                                      1/1     1
1           94s
deployment.apps/ml-pipeline                                1/1     1
1           94s
deployment.apps/ml-pipeline-persistenceagent               1/1     1
1           93s
deployment.apps/ml-pipeline-scheduledworkflow              1/1     1
1           93s
deployment.apps/ml-pipeline-ui                             1/1     1
1           93s
deployment.apps/ml-pipeline-viewer-controller-deployment   1/1     1
1           93s
deployment.apps/mysql                                      1/1     1
1           94s
deployment.apps/notebook-controller-deployment             1/1     1
1           95s
deployment.apps/profiles-deployment                        1/1     1
1           92s
deployment.apps/pytorch-operator                           1/1     1
1           95s
deployment.apps/spartakus-volunteer                        1/1     1
1           94s
deployment.apps/tensorboard                                1/1     1
1           94s
deployment.apps/tf-job-dashboard                           1/1     1
1           94s
deployment.apps/tf-job-operator                            1/1     1
1           94s
deployment.apps/workflow-controller                        1/1     1
1           97s
NAME
DESIRED   CURRENT   READY   AGE
replicaset.apps/admission-webhook-deployment-6b89c84c98              1
1         1       97s
```

```
replicaset.apps/argo-ui-5dcf5d8b4f                                  1
1         1       97s
replicaset.apps/centraldashboard-cf4874ddc                          1
1         1       97s
replicaset.apps/jupyter-web-app-deployment-685b455447               1
1         1       97s
replicaset.apps/katib-controller-88c97d85c                          1
1         1       96s
replicaset.apps/katib-db-8598468fd8                                 1
1         1       97s
replicaset.apps/katib-manager-574c8c67f9                            1
1         1       96s
replicaset.apps/katib-manager-rest-778857c989                       1
1         1       96s
replicaset.apps/katib-suggestion-bayesianoptimization-65df4d7455    1
1         1       95s
replicaset.apps/katib-suggestion-grid-56bf69f597                    1
1         1       95s
replicaset.apps/katib-suggestion-hyperband-7777b76cb9               1
1         1       95s
replicaset.apps/katib-suggestion-nasrl-77f6f9458c                   1
1         1       95s
replicaset.apps/katib-suggestion-random-77b88b5c79                  1
1         1       95s
replicaset.apps/katib-ui-7587c5b967                                 1
1         1       96s
replicaset.apps/metadata-db-5dd459cc                                1
1         1       96s
replicaset.apps/metadata-deployment-6cf77db994                      3
3         3       96s
replicaset.apps/metadata-ui-78f5b59b56                              1
1         1       96s
replicaset.apps/minio-758b769d67                                    1
1         1       93s
replicaset.apps/ml-pipeline-5875b9db95                              1
1         1       93s
replicaset.apps/ml-pipeline-persistenceagent-9b69ddd46              1
1         1       92s
replicaset.apps/ml-pipeline-scheduledworkflow-7b8d756c76            1
1         1       91s
replicaset.apps/ml-pipeline-ui-79ffd9c76                            1
1         1       91s
replicaset.apps/ml-pipeline-viewer-controller-deployment-5fdc87f58  1
1         1       91s
replicaset.apps/mysql-657f87857d                                    1
1         1       92s
```

```
replicaset.apps/notebook-controller-deployment-56b4f59bbf            1
1         1       94s
replicaset.apps/profiles-deployment-6bc745947                        1
1         1       91s
replicaset.apps/pytorch-operator-77c97f4879                          1
1         1       94s
replicaset.apps/spartakus-volunteer-5fdfddb779                       1
1         1       94s
replicaset.apps/tensorboard-6544748d94                               1
1         1       93s
replicaset.apps/tf-job-dashboard-56f79c59dd                          1
1         1       93s
replicaset.apps/tf-job-operator-79cbfd6dbc                           1
1         1       93s
replicaset.apps/workflow-controller-db644d554                        1
1         1       97s
NAME                                                        READY   AGE
statefulset.apps/admission-webhook-bootstrap-stateful-set   1/1     97s
statefulset.apps/application-controller-stateful-set        1/1     98s
statefulset.apps/metacontroller                             1/1     98s
statefulset.apps/seldon-operator-controller-manager         1/1     92s
$ kubectl get pvc -n kubeflow
NAME             STATUS   VOLUME
CAPACITY   ACCESS MODES   STORAGECLASS               AGE
katib-mysql      Bound    pvc-b07f293e-d028-11e9-9b9d-00505681a82d
10Gi       RWO            ontap-ai-flexvols-retain   27m
metadata-mysql   Bound    pvc-b0f3f032-d028-11e9-9b9d-00505681a82d
10Gi       RWO            ontap-ai-flexvols-retain   27m
minio-pv-claim   Bound    pvc-b22727ee-d028-11e9-9b9d-00505681a82d
20Gi       RWO            ontap-ai-flexvols-retain   27m
mysql-pv-claim   Bound    pvc-b2429afd-d028-11e9-9b9d-00505681a82d
20Gi       RWO            ontap-ai-flexvols-retain   27m
```

4. En su navegador web, acceda al panel central de Kubeflow navegando hasta la URL que anotó en el paso 2.

   El nombre de usuario predeterminado es `admin@kubeflow.org`, y la contraseña predeterminada es `12341234`. Para crear usuarios adicionales, siga las instrucciones de "Documentación oficial de Kubeflow".



Ejemplo de operaciones y tareas de Kubeflow

Esta sección incluye ejemplos de diversas operaciones y tareas que puede que desee realizar utilizando Kubeflow.

Ejemplo de operaciones y tareas de Kubeflow

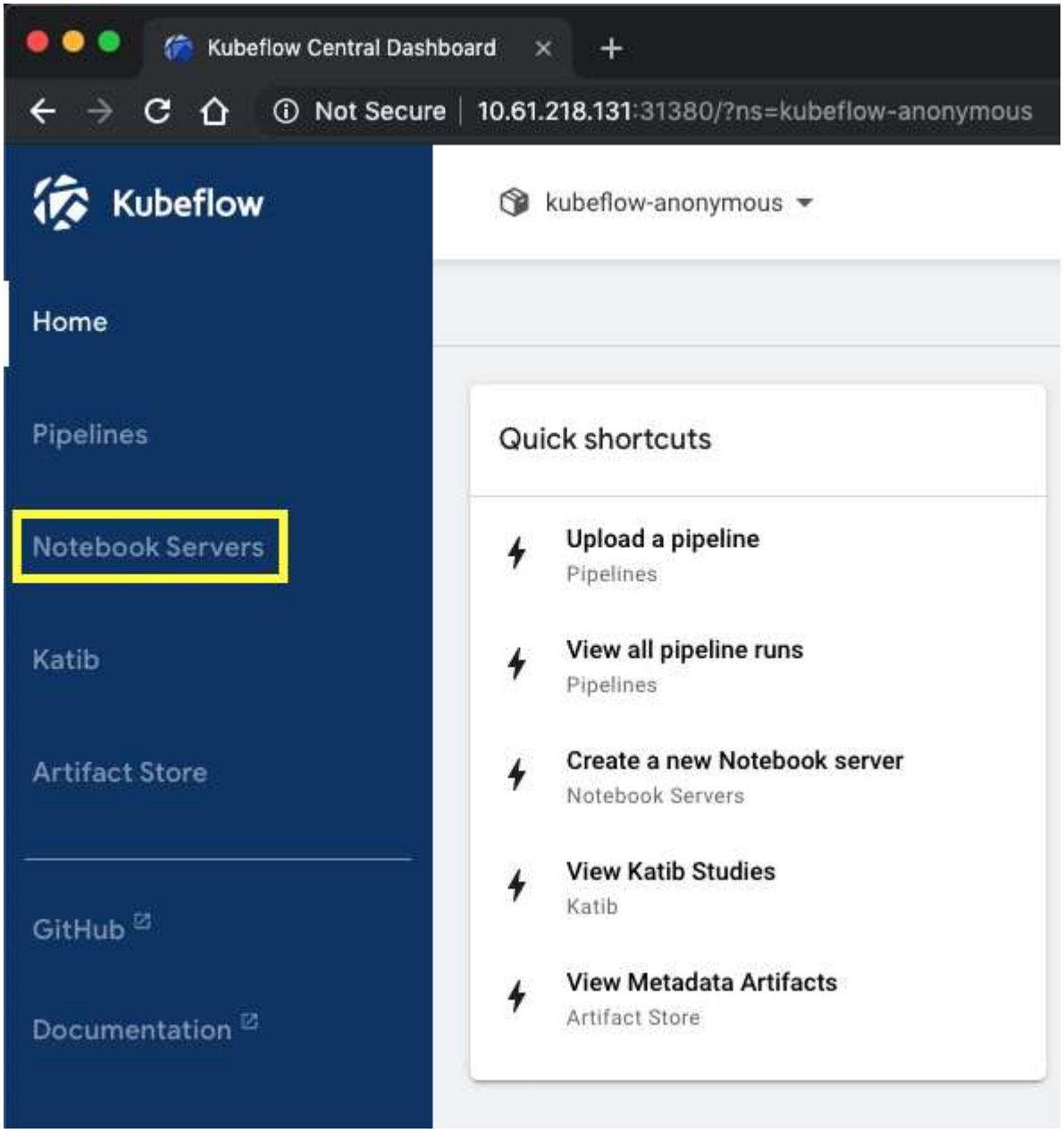
Esta sección incluye ejemplos de diversas operaciones y tareas que puede que desee realizar utilizando Kubeflow.

Aprovisione un espacio de trabajo para portátiles Juppyter para uso científico de datos o desarrollador

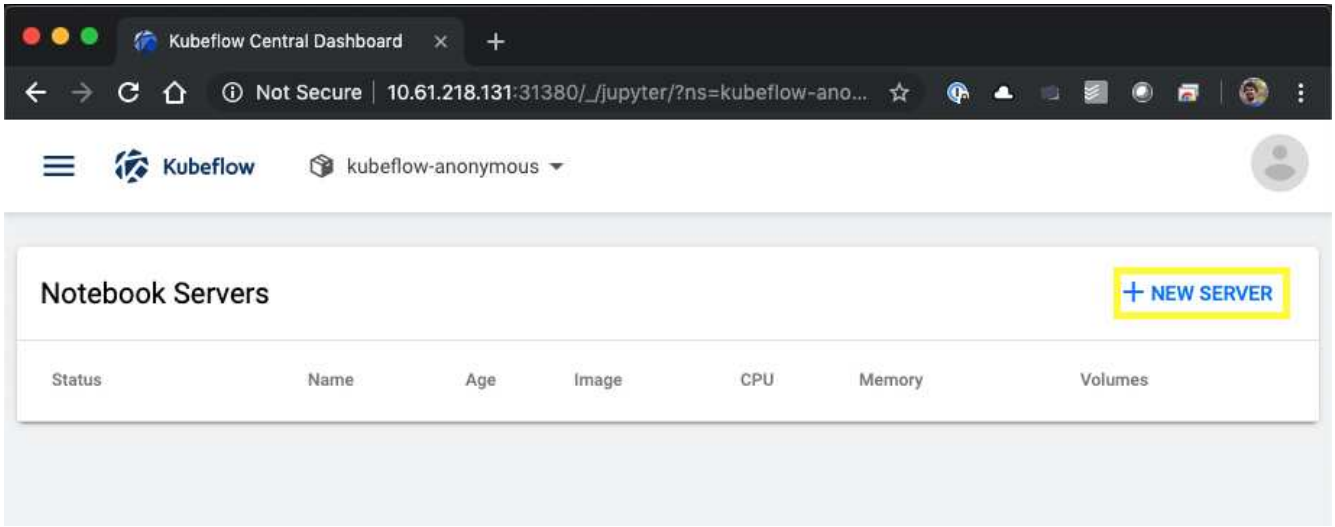
Kubeflow es capaz de suministrar rápidamente nuevos servidores Jupyter Notebook para actuar como espacios de trabajo de científicos de datos. Para aprovisionar un nuevo servidor Juppyter Notebook con Kubeflow, realice las siguientes tareas. Para obtener más información acerca de Jupyter Notebooks dentro del contexto de Kubeflow, consulte "Documentación oficial de Kubeflow".

1. Desde el panel central de Kubeflow, haga clic en Notebook Servers en el menú principal para acceder a la

página de administración del servidor Jupyter Notebook.

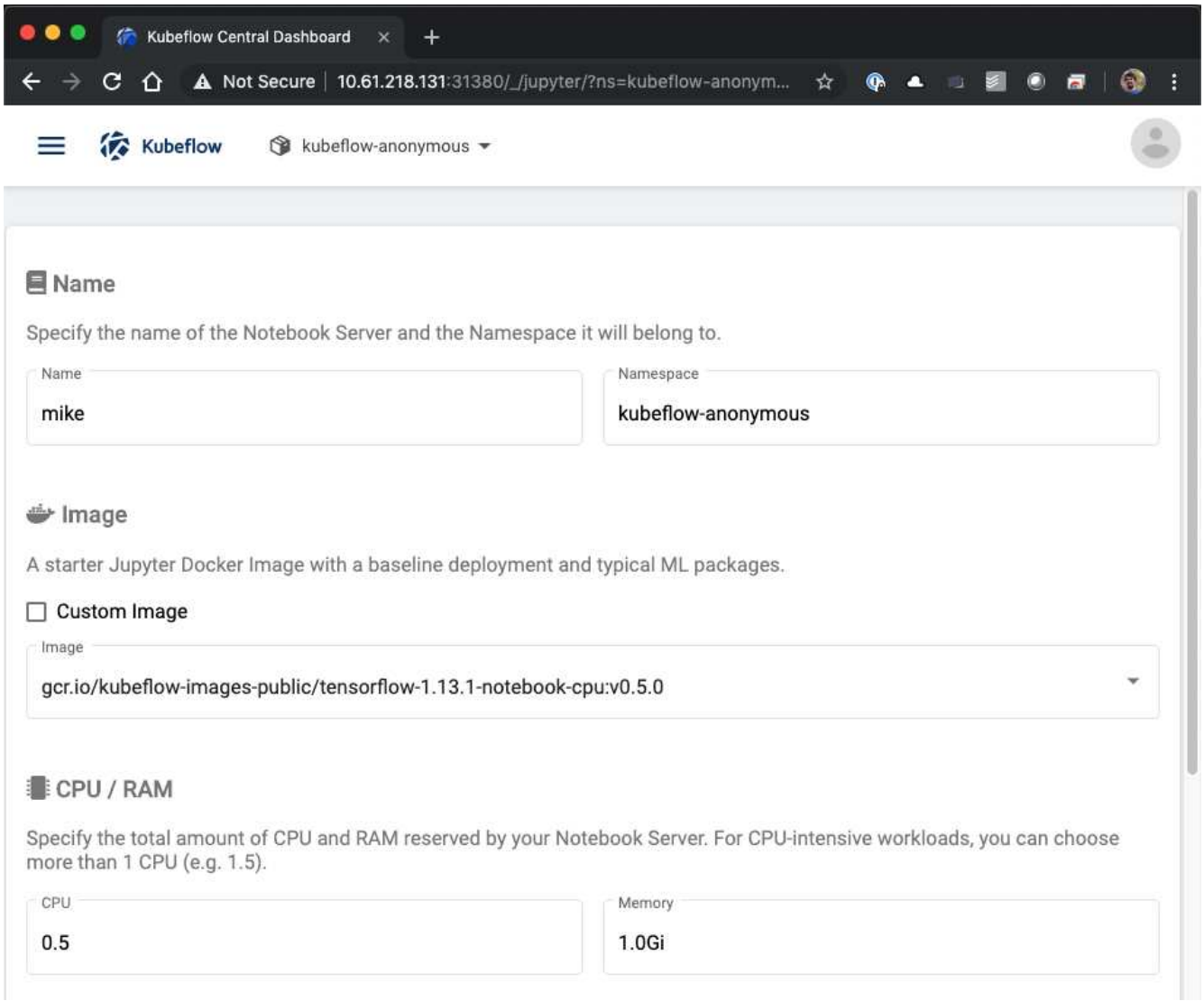


2. Haga clic en Nuevo servidor para aprovisionar un nuevo servidor Juppyter Notebook.

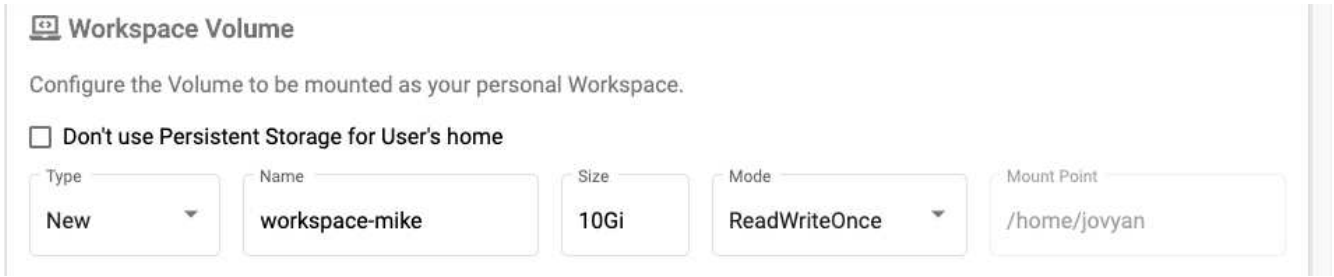


3. Asigne un nombre a su nuevo servidor, elija la imagen de Docker en la que desea que se base su servidor y especifique la cantidad de CPU y RAM que debe reservar su servidor. Si el campo espacio de nombres está en blanco, utilice el menú Seleccionar espacio de nombres en el encabezado de la página para elegir un espacio de nombres. El campo de espacio de nombres se rellena automáticamente con el espacio de nombres elegido.

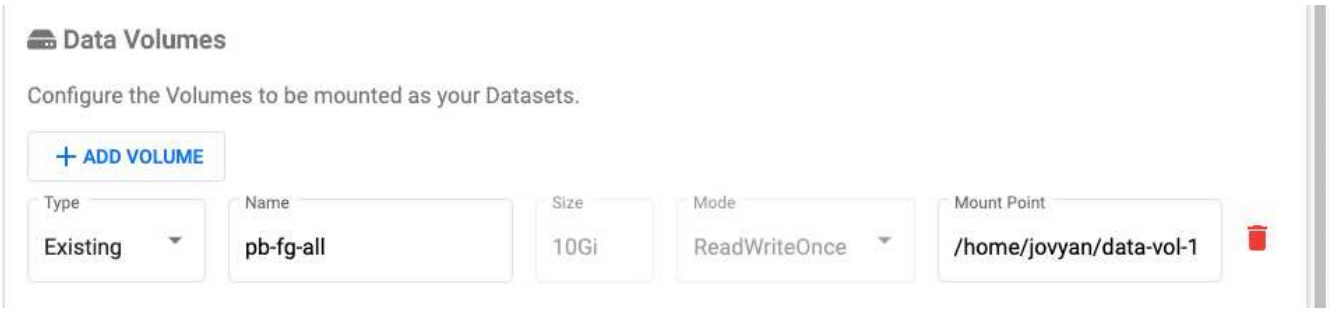
   En el siguiente ejemplo, la `kubeflow-anonymous` se elige espacio de nombres. Además, se aceptan los valores predeterminados de imagen Docker, CPU y RAM.



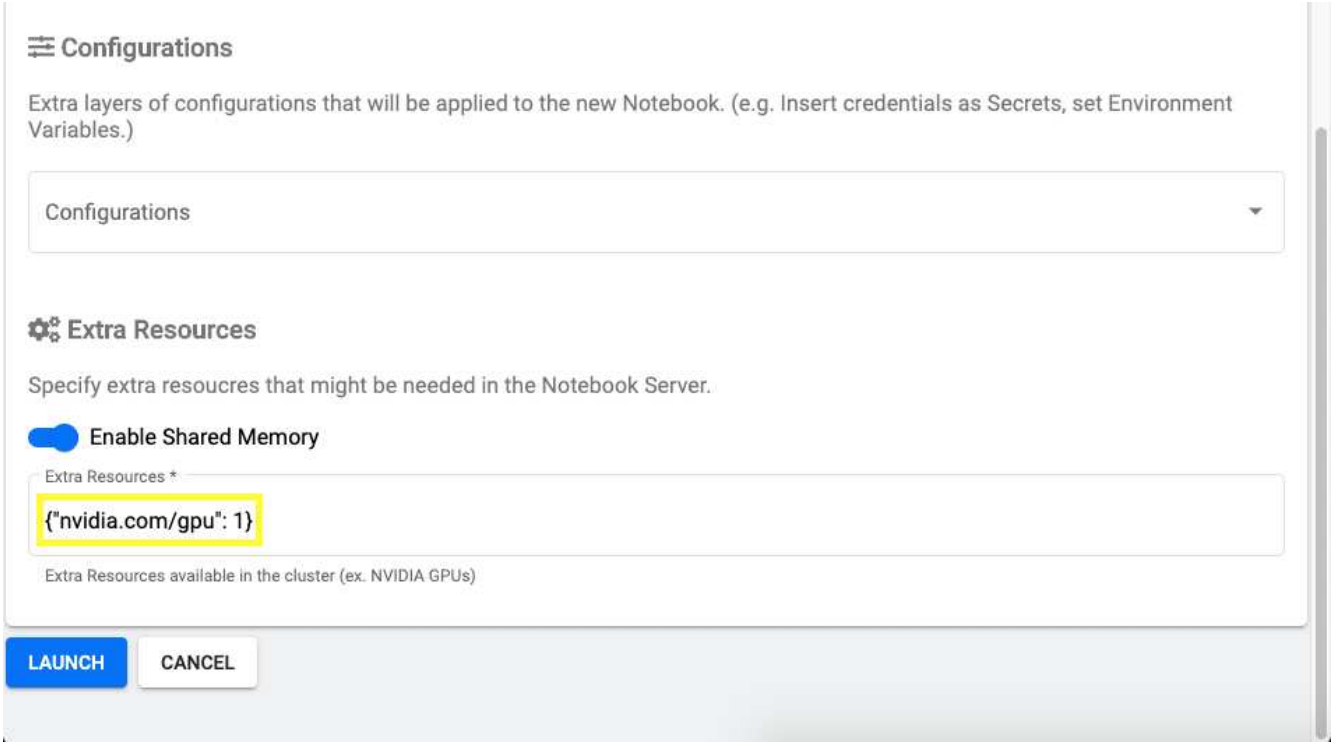
4. Especifique los detalles del volumen del área de trabajo. Si decide crear un nuevo volumen, ese volumen o RVP se aprovisionan con el tipo de almacenamiento predeterminado. Dado que se designó un clase de almacenamiento con Trident como el clase de almacenamiento predeterminado de la sección "Despliegue de Kubeflow", El volumen o PVC se aprovisiona con Trident. Este volumen se monta automáticamente como espacio de trabajo predeterminado dentro del contenedor servidor del portátil Jupyter. Los portátiles que cree un usuario en el servidor que no se guarden en un volumen de datos independiente se guardarán automáticamente en este volumen de área de trabajo. Por lo tanto, los portátiles se conservan entre reinicios.



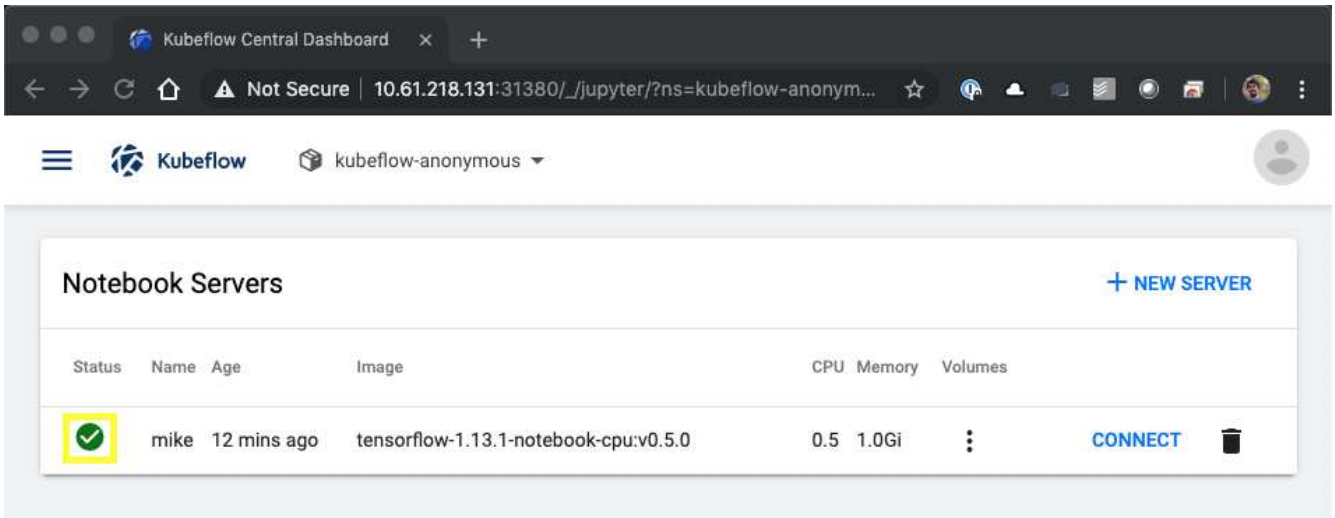
5. Añadir volúmenes de datos. En el siguiente ejemplo, se especifica un RVP existente llamado 'pb-fg-All' y acepta el punto de montaje predeterminado.



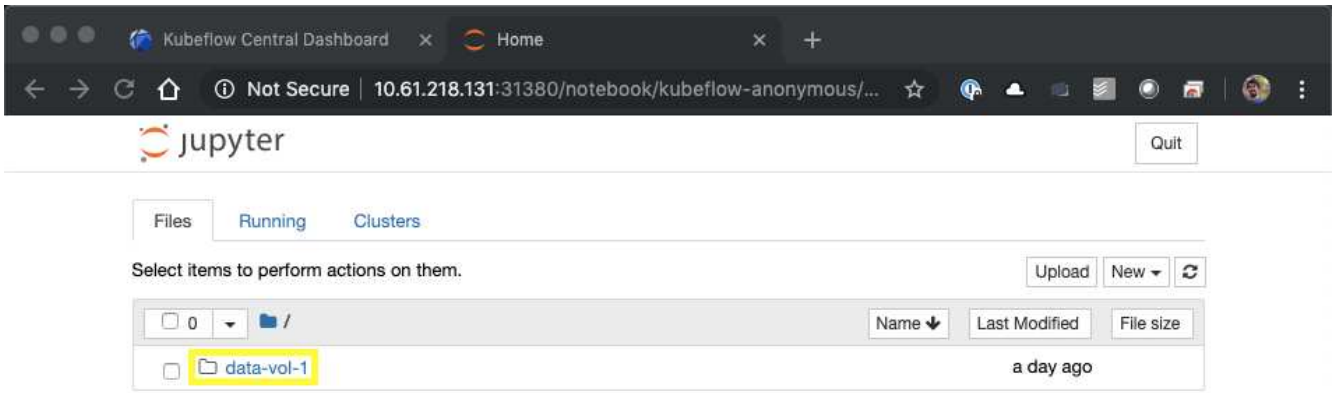
6. **Opcional:** Solicite que se asigne el número deseado de GPU al servidor de su portátil. En el siguiente ejemplo, se solicita una GPU.

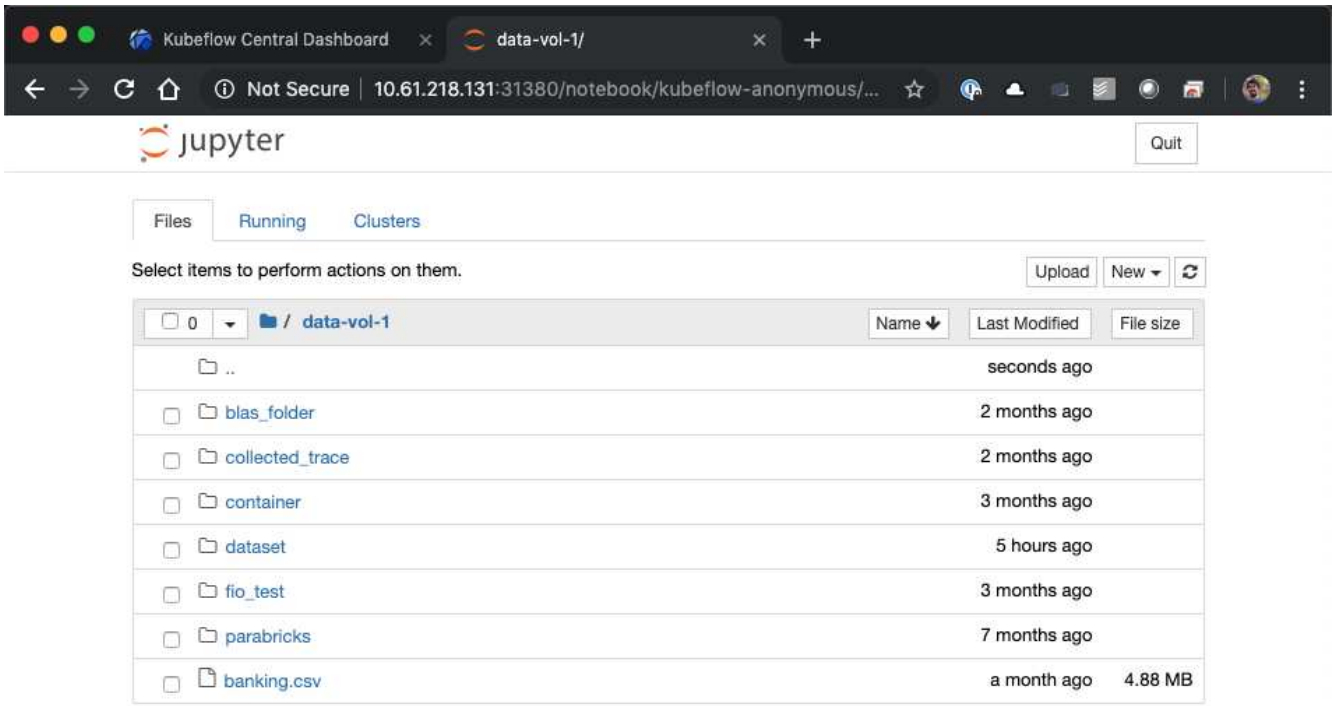


7. Haga clic en Iniciar para aprovisionar su nuevo servidor de notebook.
8. Espere a que el servidor de su portátil esté completamente aprovisionado. Esto puede tardar varios minutos si nunca ha aprovisionado un servidor con la imagen Docker que ha especificado porque es necesario descargar la imagen. Una vez que el servidor ha sido completamente aprovisionado, verá una Marca de verificación verde en la columna Estado de la página de administración del servidor Jupyter Notebook.



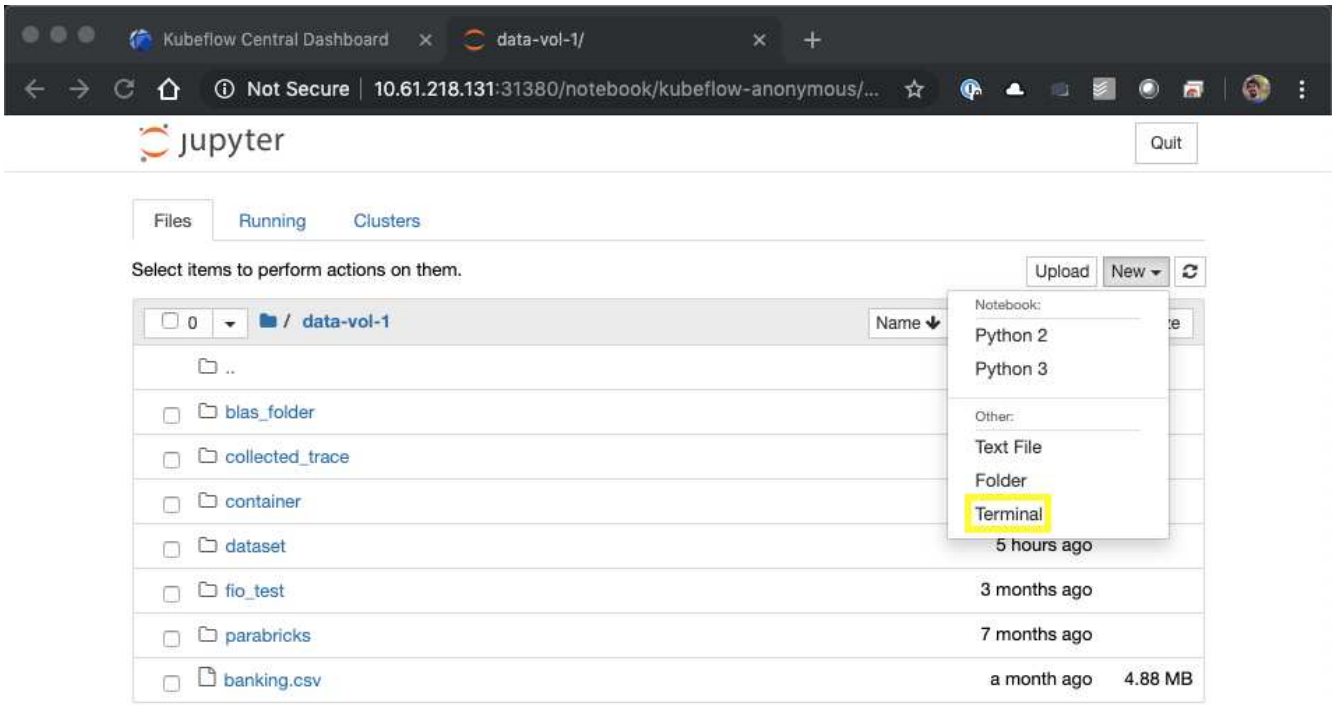
9. Haga clic en conectar para conectarse a la nueva interfaz web del servidor.
10. Confirme que el volumen del conjunto de datos especificado en el paso 6 está montado en el servidor. Tenga en cuenta que este volumen se monta en el espacio de trabajo predeterminado de forma predeterminada. Desde la perspectiva del usuario, ésta es sólo otra carpeta dentro del área de trabajo. El usuario, que es probable que un científico de datos y no sea un experto en infraestructura, no necesita tener experiencia en almacenamiento para utilizar este volumen.





11. Abra un terminal y, suponiendo que se haya solicitado un nuevo volumen en el paso 5, ejecute `df -h` Para confirmar que un nuevo volumen persistente aprovisionado por Trident se monta como espacio de trabajo predeterminado.

    El directorio de área de trabajo predeterminado es el directorio base con el que se presenta cuando se accede por primera vez a la interfaz web del servidor. Por lo tanto, cualquier artefacto que cree mediante la interfaz web se almacena en este volumen persistente aprovisionado por Trident.



```
$ df -h
Filesystem                                                      Size  Used Avail
Use% Mounted on
overlay                                                         439G   34G  382G
  9% /
tmpfs                                                            64M     0   64M
  0% /dev
tmpfs                                                           252G     0  252G
  0% /sys/fs/cgroup
/dev/sda2                                                       439G   34G  382G
  9% /etc/hosts
192.168.11.11:/trident_pvc_3dcfe7e5_d5a9_11e9_9b9d_00505681a82d   10G  320K   10G
  1% /home/jovyan
tmpfs                                                           252G     0  252G
  0% /dev/shm
192.168.11.11:/pb_fg_all                                         10T   10T   47G
100% /home/jovyan/data-vol-1
tmpfs                                                           252G   12K  252G
  1% /run/secrets/kubernetes.io/serviceaccount
tmpfs                                                           252G   12K  252G
  1% /proc/driver/nvidia
tmpfs                                                            51G  4.9M   51G
  1% /run/nvidia-persistenced/socket
udev                                                            252G     0  252G
  0% /dev/nvidia5
tmpfs                                                           252G     0  252G
  0% /proc/acpi
tmpfs                                                           252G     0  252G
  0% /proc/scsi
tmpfs                                                           252G     0  252G
  0% /sys/firmware
$
```

12. Con el terminal, ejecute `nvidia-smi` Para confirmar que se ha asignado el número correcto de GPU al servidor del portátil. En el siguiente ejemplo, se ha asignado una GPU al servidor de portátiles como se solicitó en el paso 7.

```
$ nvidia-smi
Fri Sep 13 13:52:15 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 410.104      Driver Version: 410.104      CUDA Version: N/A      |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla V100-SXM2...  On   | 00000000:86:00.0 Off |                    0 |
| N/A   38C    P0    46W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
$
```

Ejemplos de cuadernos y canalizaciones

La "Kit de herramientas para la ciencia de datos de NetApp para Kubernetes" Se puede utilizar junto con Kubeflow. El uso del kit de herramientas para la ciencia de datos de NetApp con Kubeflow ofrece las siguientes ventajas:

- Los científicos de datos pueden llevar a cabo operaciones avanzadas de gestión de datos de NetApp directamente desde un portátil Juppyter.
- Las operaciones avanzadas de gestión de datos de NetApp pueden incorporarse en flujos de trabajo automatizados mediante el marco de canalizaciones de Kubeflow.

Consulte la "Ejemplos de Kubeflow" Sección dentro del repositorio de Data Science Toolkit de NetApp, GitHub para obtener información sobre el uso del kit de herramientas con Kubeflow.

Distribución del flujo de aire de Apache

NetApp recomienda ejecutar el flujo de aire de Apache sobre Kubernetes. En esta sección se describen las tareas que debe completar para poner en marcha el flujo de aire en el clúster de Kubernetes.

> Es posible poner en marcha un flujo de aire en plataformas distintas a Kubernetes. Esta solución no cubre la posibilidad de poner en marcha un flujo de aire en plataformas distintas a Kubernetes.

Requisitos previos

Antes de realizar el ejercicio de implementación descrito en esta sección, asumimos que ya ha realizado las siguientes tareas:

1. Ya tiene un clúster de Kubernetes en funcionamiento.
2. Ya ha instalado y configurado NetApp Trident en su clúster de Kubernetes como se indica en la sección “NetApp Trident Deployment and Configuration”.

Instale el Helm

El flujo de aire se pone en marcha con Helm, un conocido administrador de paquetes para Kubernetes. Antes de implementar el flujo de aire, debe instalar Helm en el host de salto de la implementación. Para instalar Helm en el host de salto de despliegue, siga la "instrucciones de instalación" En la documentación oficial de Helm.

Establezca el tipo de almacenamiento de Kubernetes predeterminado

Antes de implementar el flujo de aire, debe designar un tipo de almacenamiento predeterminado en el clúster de Kubernetes. El proceso de implementación de flujo de aire intenta aprovisionar nuevos volúmenes persistentes mediante el tipo de almacenamiento predeterminado. Si no se designa StorageClass como clase de almacenamiento predeterminado, la implementación falla. Para designar un tipo de almacenamiento predeterminado en el clúster, siga las instrucciones que se describen en la sección "Despliegue de Kubeflow". Si ya ha designado un tipo de almacenamiento predeterminado en el clúster, puede omitir este paso.

Utilice Helm para desplegar el flujo de aire

Para poner en marcha el flujo de aire en su clúster de Kubernetes con Helm, realice las siguientes tareas desde el host de salto de implementación:

1. Despliegue el flujo de aire con Helm siguiendo la "instrucciones de puesta en funcionamiento" Para el diagrama de flujo de aire oficial en el Hub de artefactos. Los comandos de ejemplo siguientes muestran la implementación del flujo de aire con Helm. Modifique, agregue o elimine valores en la `custom-values.yaml` fichero según sea necesario en función de su entorno y de la configuración deseada.

```
$ cat << EOF > custom-values.yaml
###################################
# Airflow - Common Configs
###################################
airflow:
  ## the airflow executor type to use
  ##
  executor: "CeleryExecutor"
  ## environment variables for the web/scheduler/worker Pods (for
airflow configs)
  ##
  #
###################################
# Airflow - WebUI Configs
###################################
web:
  ## configs for the Service of the web Pods
  ##
  service:
    type: NodePort
###################################
# Airflow - Logs Configs
###################################
logs:
  persistence:
    enabled: true
###################################
# Airflow - DAGs Configs
###################################
dags:
  ## configs for the DAG git repository & sync container
  ##
  gitSync:
    enabled: true
    ## url of the git repository
    ##
    repo: "git@github.com:mboglesby/airflow-dev.git"
```

```
    ## the branch/tag/sha1 which we clone
    ##
    branch: master
    revision: HEAD
    ## the name of a pre-created secret containing files for ~/.ssh/
    ##
    ## NOTE:
    ## - this is ONLY RELEVANT for SSH git repos
    ## - the secret commonly includes files: id_rsa, id_rsa.pub,
known_hosts
    ## - known_hosts is NOT NEEDED if `git.sshKeyscan` is true
    ##
    sshSecret: "airflow-ssh-git-secret"
    ## the name of the private key file in your `git.secret`
    ##
    ## NOTE:
    ## - this is ONLY RELEVANT for PRIVATE SSH git repos
    ##
    sshSecretKey: id_rsa
    ## the git sync interval in seconds
    ##
    syncWait: 60
EOF
$ helm install airflow airflow-stable/airflow -n airflow --version 8.0.8
--values ./custom-values.yaml
...
Congratulations. You have just deployed Apache Airflow!
1. Get the Airflow Service URL by running these commands:
   export NODE_PORT=$(kubectl get --namespace airflow -o
jsonpath="{.spec.ports[0].nodePort}" services airflow-web)
   export NODE_IP=$(kubectl get nodes --namespace airflow -o
jsonpath="{.items[0].status.addresses[0].address}")
   echo http://$NODE_IP:$NODE_PORT/
2. Open Airflow in your web browser
```

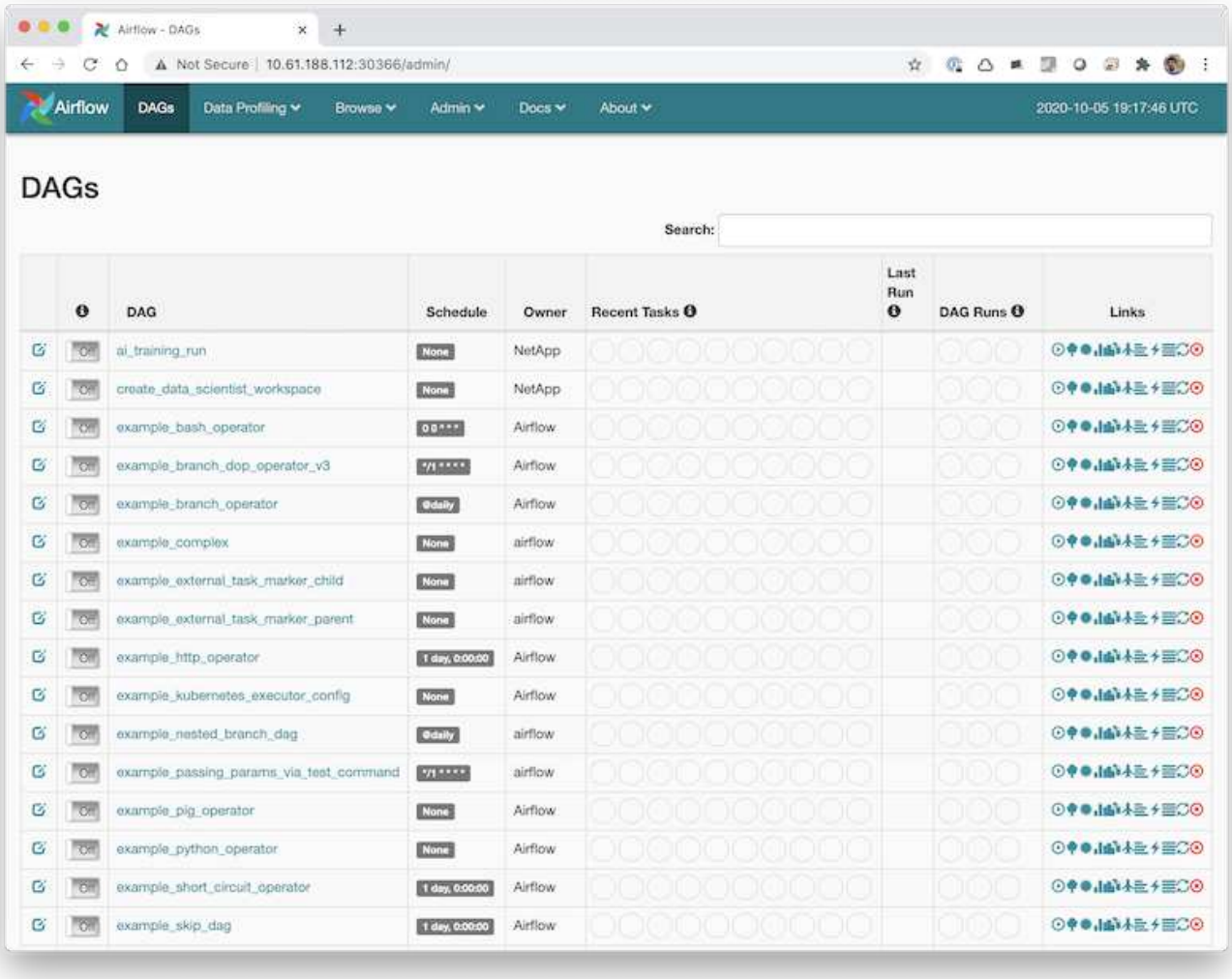
2. Confirme que todos los pods de flujo de aire estén activos y en funcionamiento. Puede tardar varios minutos en comenzar todos los pods.

```
$ kubectl -n airflow get pod
NAME                                READY   STATUS    RESTARTS   AGE
airflow-flower-b5656d44f-h8qjk      1/1     Running   0          2h
airflow-postgresql-0                1/1     Running   0          2h
airflow-redis-master-0              1/1     Running   0          2h
airflow-scheduler-9d95fcdf9-clf4b   2/2     Running   2          2h
airflow-web-59c94db9c5-z7rg4        1/1     Running   0          2h
airflow-worker-0                    2/2     Running   2          2h
```

3. Obtenga la URL del servicio web de flujo de aire siguiendo las instrucciones que se imprimieron en la consola cuando implementó el flujo de aire con Helm en el paso 1.

```
$ export NODE_PORT=$(kubectl get --namespace airflow -o
jsonpath="{.spec.ports[0].nodePort}" services airflow-web)
$ export NODE_IP=$(kubectl get nodes --namespace airflow -o
jsonpath="{.items[0].status.addresses[0].address}")
$ echo http://$NODE_IP:$NODE_PORT/
```

4. Confirme que puede acceder al servicio web de flujo de aire.



Ejemplo de flujos de trabajo de flujo de aire de Apache

La "Kit de herramientas para la ciencia de datos de NetApp para Kubernetes" Se puede utilizar junto con el flujo de aire. El uso del kit de herramientas para la ciencia de datos de NetApp con flujo de aire le permite incorporar operaciones de gestión de datos de NetApp en flujos de trabajo automatizados orquestados por flujo de aire.

Consulte la "Ejemplos de flujo de aire" Sección dentro del repositorio de Data Science Toolkit de NetApp, GitHub para obtener información sobre el uso del kit de herramientas con flujo de aire.

Ejemplo de operaciones de Trident

En esta sección se incluyen ejemplos de diversas operaciones que quizás desee realizar con Trident.

Importe un volumen existente

Si hay volúmenes existentes en su sistema/plataforma de almacenamiento de NetApp que desea montar en contenedores dentro de su clúster de Kubernetes, pero que no están ligados a las RVP en el clúster, debe importar estos volúmenes. Es posible usar la funcionalidad de importación de volúmenes de Trident para

importar estos volúmenes.

Los comandos de ejemplo siguientes muestran la importación del mismo volumen, denominado `pb_fg_all`, Dos veces, una para cada backend de Trident que se creó en el ejemplo de la sección "Ejemplo de Back-ends de Trident para puestas en marcha de ONTAP AI", paso 1. Importar el mismo volumen dos veces de esta manera le permite montar el volumen (un volumen FlexGroup existente) varias veces en diferentes LIF, como se describe en la sección "Ejemplo de Back-ends de Trident para puestas en marcha de ONTAP AI", paso 1. Para obtener más información acerca de las EVs, consulte "Documentación oficial sobre Kubernetes". Para obtener más información sobre la funcionalidad de importación de volúmenes, consulte "Documentación de Trident".

An `accessModes` valor de `ReadOnlyMany` Se especifica en los archivos de especificaciones de PVC de ejemplo. Para obtener más información acerca de `accessMode` consulte "Documentación oficial sobre Kubernetes".

> Los nombres de backend especificados en los siguientes comandos de importación de ejemplo corresponden a las backends que se crearon en el ejemplo de la sección "Ejemplo de Back-ends de Trident para puestas en marcha de ONTAP AI", paso 1. Los nombres de StorageClass especificados en el siguiente ejemplo de archivos de definición PVC corresponden a las clases de almacenamiento que se crearon en el ejemplo de la sección "Ejemplo de clases de almacenamiento Kubernetes para puestas en marcha de IA en ONTAP", paso 1.

```
$ cat << EOF > ./pvc-import-pb_fg_all-iface1.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pb-fg-all-iface1
  namespace: default
spec:
  accessModes:
    - ReadOnlyMany
  storageClassName: ontap-ai-flexgroups-retain-iface1
EOF
$ tridentctl import volume ontap-ai-flexgroups-iface1 pb_fg_all -f ./pvc-
import-pb_fg_all-iface1.yaml -n trident
+--------------------------------+--------
+-----------------------------------+----------
+-----------------------------------------+--------+---------+
|          NAME                  |  SIZE  |       STORAGE CLASS
| PROTOCOL |             BACKEND UUID                        | STATE  |
MANAGED |
+--------------------------------+--------
+-----------------------------------+----------
+-----------------------------------------+--------+---------+
| default-pb-fg-all-iface1-7d9f1 | 10 TiB | ontap-ai-flexgroups-retain-
iface1 | file     | b74cbddb-e0b8-40b7-b263-b6da6dec0bdd | online | true
|
+--------------------------------+--------
```

```
+-----------------------------------+----------
+-------------------------------------------+--------+---------+
$ cat << EOF > ./pvc-import-pb_fg_all-iface2.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pb-fg-all-iface2
  namespace: default
spec:
  accessModes:
    - ReadOnlyMany
  storageClassName: ontap-ai-flexgroups-retain-iface2
EOF
$ tridentctl import volume ontap-ai-flexgroups-iface2 pb_fg_all -f ./pvc-
import-pb_fg_all-iface2.yaml -n trident
+--------------------------------+--------
+-----------------------------------+----------
+-------------------------------------------+--------+---------+
|          NAME                  |  SIZE  |       STORAGE CLASS
| PROTOCOL |             BACKEND UUID                        | STATE  |
MANAGED |
+--------------------------------+--------
+-----------------------------------+----------
+-------------------------------------------+--------+---------+
| default-pb-fg-all-iface2-85aee | 10 TiB | ontap-ai-flexgroups-retain-
iface2 | file     | 61814d48-c770-436b-9cb4-cf7ee661274d | online | true
|
+--------------------------------+--------
+-----------------------------------+----------
+-------------------------------------------+--------+---------+
$ tridentctl get volume -n trident
+-----------------------------------+---------
+-----------------------------------+----------
+--------------------------------------+--------+---------+
|               NAME               |  SIZE   |           STORAGE CLASS
| PROTOCOL |             BACKEND UUID             | STATE  | MANAGED |
+-----------------------------------+---------
+-----------------------------------+----------
+--------------------------------------+--------+---------+
| default-pb-fg-all-iface1-7d9f1   | 10 TiB  | ontap-ai-flexgroups-retain-
iface1 | file     | b74cbddb-e0b8-40b7-b263-b6da6dec0bdd | online | true
|
| default-pb-fg-all-iface2-85aee   | 10 TiB  | ontap-ai-flexgroups-retain-
iface2 | file     | 61814d48-c770-436b-9cb4-cf7ee661274d | online | true
|
+-----------------------------------+---------
```

```
+-----------------------------------+----------
+--------------------------------------+--------+---------+
$ kubectl get pvc
NAME                  STATUS   VOLUME                                CAPACITY
ACCESS MODES   STORAGECLASS                        AGE
pb-fg-all-iface1      Bound    default-pb-fg-all-iface1-7d9f1
10995116277760   ROX            ontap-ai-flexgroups-retain-iface1   25h
pb-fg-all-iface2      Bound    default-pb-fg-all-iface2-85aee
10995116277760   ROX            ontap-ai-flexgroups-retain-iface2   25h
```

Aprovisione un nuevo volumen

Puede usar Trident para aprovisionar un nuevo volumen en su plataforma o sistema de almacenamiento de NetApp. Los siguientes comandos de ejemplo muestran el aprovisionamiento de un volumen FlexVol nuevo. En este ejemplo, se aprovisiona el volumen con el tipo de almacenamiento que se creó en el ejemplo de la sección "Ejemplo de clases de almacenamiento Kubernetes para puestas en marcha de IA en ONTAP", paso 2.

An `accessModes` valor de `ReadWriteMany` Se especifica en el siguiente archivo de definición de PVC de ejemplo. Para obtener más información acerca de `accessMode` consulte "Documentación oficial sobre Kubernetes".

```
$ cat << EOF > ./pvc-tensorflow-results.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: tensorflow-results
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-ai-flexvols-retain
EOF
$ kubectl create -f ./pvc-tensorflow-results.yaml
persistentvolumeclaim/tensorflow-results created
$ kubectl get pvc
NAME                              STATUS    VOLUME
CAPACITY         ACCESS MODES   STORAGECLASS                        AGE
pb-fg-all-iface1                  Bound     default-pb-fg-all-iface1-7d9f1
10995116277760   ROX            ontap-ai-flexgroups-retain-iface1   26h
pb-fg-all-iface2                  Bound     default-pb-fg-all-iface2-85aee
10995116277760   ROX            ontap-ai-flexgroups-retain-iface2   26h
tensorflow-results                Bound     default-tensorflow-results-
2fd60   1073741824       RWX            ontap-ai-flexvols-retain
25h
```

Ejemplo de tareas de alto rendimiento para puestas en marcha de ONTAP AI

En esta sección, se incluyen ejemplos de varias tareas de alto rendimiento que se pueden ejecutar cuando se pone en marcha Kubernetes en un pod ONTAP AI.

Ejemplo de tareas de alto rendimiento para puestas en marcha de ONTAP AI

En esta sección, se incluyen ejemplos de varias tareas de alto rendimiento que se pueden ejecutar cuando se pone en marcha Kubernetes en un pod ONTAP AI.

Ejecute una carga de trabajo de IA de un solo nodo

Para ejecutar una tarea DE IA y ML de un solo nodo en el clúster de Kubernetes, realice las siguientes tareas desde el host de puesta en marcha. Con Trident, puede crear de forma rápida y sencilla un volumen de datos, con potencialmente petabytes de datos al que se puede acceder una carga de trabajo de Kubernetes. Para que un volumen de datos de este tipo sea accesible desde un pod de Kubernetes, solo tiene que especificar una RVP en la definición del pod. Este paso es una operación nativa de Kubernetes, no se necesita experiencia en NetApp.

En esta sección se supone que ya ha realizado un contenedor (en el formato de contenedor de Docker) con la carga de trabajo específica DE IA y ML que intenta ejecutar en su clúster de Kubernetes.

1. Los siguientes comandos de ejemplo muestran la creación de un trabajo de Kubernetes para una carga de trabajo de prueba de ImageNET que utiliza el conjunto de datos de TensorFlow. Para obtener más información acerca del conjunto de datos ImageNET, consulte "Sitio web de ImageNET".

   Este trabajo de ejemplo solicita ocho GPU y, por lo tanto, puede ejecutarse en un solo nodo de trabajo de GPU con ocho o más GPU. Este trabajo de ejemplo se puede enviar en un clúster para el que no hay un nodo de trabajo con ocho o más GPU o esté ocupado actualmente con otra carga de trabajo. Si es así, el trabajo permanece en estado pendiente hasta que dicho nodo de trabajo esté disponible.

   Además, para maximizar el ancho de banda de almacenamiento, el volumen que contiene los datos de entrenamiento necesarios se monta dos veces en el pod que crea este trabajo. Otro volumen también se monta en el pod. Este segundo volumen se utilizará para almacenar resultados y métricas. Estos volúmenes se hacen referencia en la definición de trabajo utilizando los nombres de las RVP. Para obtener más información sobre los trabajos de Kubernetes, consulte "Documentación oficial sobre Kubernetes".

   An `emptyDir` volumen con un `medium` valor de `Memory` está montado en `/dev/shm` en el pod que crea este trabajo de ejemplo. El tamaño predeterminado de `/dev/shm` El volumen virtual que se crea automáticamente mediante el tiempo de ejecución del contenedor Docker puede en ocasiones ser insuficiente para las necesidades de TensorFlow. Montaje de un `emptyDir` volumen como en el ejemplo siguiente proporciona un tamaño suficiente `/dev/shm` volumen virtual. Para obtener más información acerca de `emptyDir` volúmenes, consulte "Documentación oficial sobre Kubernetes".

   El contenedor único que se especifica en esta definición de trabajo de ejemplo se proporciona un `securityContext > privileged` valor de `true`. Este valor significa que el contenedor tiene acceso raíz en el host de forma efectiva. Esta anotación se utiliza en este caso porque la carga de trabajo específica que se está ejecutando requiere acceso raíz. Específicamente, una operación de caché clara que ejecuta la carga de trabajo requiere acceso raíz. Si esto o no `privileged: true` la anotación es necesaria depende de los requisitos de la carga de trabajo específica que se esté ejecutando.

```
$ cat << EOF > ./netapp-tensorflow-single-imagenet.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: netapp-tensorflow-single-imagenet
spec:
  backoffLimit: 5
  template:
    spec:
      volumes:
      - name: dshm
        emptyDir:
          medium: Memory
      - name: testdata-iface1
        persistentVolumeClaim:
          claimName: pb-fg-all-iface1
      - name: testdata-iface2
```

```
        persistentVolumeClaim:
          claimName: pb-fg-all-iface2
      - name: results
        persistentVolumeClaim:
          claimName: tensorflow-results
      containers:
      - name: netapp-tensorflow-py2
        image: netapp/tensorflow-py2:19.03.0
        command: ["python", "/netapp/scripts/run.py", "--
dataset_dir=/mnt/mount_0/dataset/imagenet", "--dgx_version=dgx1", "--
num_devices=8"]
        resources:
          limits:
            nvidia.com/gpu: 8
        volumeMounts:
        - mountPath: /dev/shm
          name: dshm
        - mountPath: /mnt/mount_0
          name: testdata-iface1
        - mountPath: /mnt/mount_1
          name: testdata-iface2
        - mountPath: /tmp
          name: results
        securityContext:
          privileged: true
      restartPolicy: Never
EOF
$ kubectl create -f ./netapp-tensorflow-single-imagenet.yaml
job.batch/netapp-tensorflow-single-imagenet created
$ kubectl get jobs
NAME                                        COMPLETIONS   DURATION   AGE
netapp-tensorflow-single-imagenet           0/1           24s        24s
```

2. Confirme que el trabajo que ha creado en el paso 1 se está ejecutando correctamente. El siguiente comando de ejemplo confirma que se creó un solo pod para el trabajo, tal como se especifica en la definición de trabajos, y que este pod se ejecuta actualmente en uno de los nodos de trabajo de la GPU.

```
$ kubectl get pods -o wide
NAME                                                READY   STATUS
RESTARTS   AGE
IP              NODE            NOMINATED NODE
netapp-tensorflow-single-imagenet-m7x92             1/1     Running     0
3m    10.233.68.61    10.61.218.154   <none>
```

3. Confirme que el trabajo que ha creado en el paso 1 se ha completado correctamente. Los siguientes

comandos de ejemplo confirman que el trabajo se ha completado correctamente.

```
$ kubectl get jobs
NAME                                          COMPLETIONS   DURATION
AGE
netapp-tensorflow-single-imagenet             1/1           5m42s
10m
$ kubectl get pods
NAME                                                  READY   STATUS
RESTARTS   AGE
netapp-tensorflow-single-imagenet-m7x92               0/1     Completed
0          11m
$ kubectl logs netapp-tensorflow-single-imagenet-m7x92
[netapp-tensorflow-single-imagenet-m7x92:00008] PMIX ERROR: NO-
PERMISSIONS in file gds_dstore.c at line 702
[netapp-tensorflow-single-imagenet-m7x92:00008] PMIX ERROR: NO-
PERMISSIONS in file gds_dstore.c at line 711
Total images/sec = 6530.59125
================ Clean Cache !!! ==================
mpirun -allow-run-as-root -np 1 -H localhost:1 bash -c 'sync; echo 1 >
/proc/sys/vm/drop_caches'
==========================================
mpirun -allow-run-as-root -np 8 -H localhost:8 -bind-to none -map-by
slot -x NCCL_DEBUG=INFO -x LD_LIBRARY_PATH -x PATH python
/netapp/tensorflow/benchmarks_190205/scripts/tf_cnn_benchmarks/tf_cnn_be
nchmarks.py --model=resnet50 --batch_size=256 --device=gpu
--force_gpu_compatible=True --num_intra_threads=1 --num_inter_threads=48
--variable_update=horovod --batch_group_size=20 --num_batches=500
--nodistortions --num_gpus=1 --data_format=NCHW --use_fp16=True
--use_tf_layers=False --data_name=imagenet --use_datasets=True
--data_dir=/mnt/mount_0/dataset/imagenet
--datasets_parallel_interleave_cycle_length=10
--datasets_sloppy_parallel_interleave=False --num_mounts=2
--mount_prefix=/mnt/mount_%d --datasets_prefetch_buffer_size=2000
--datasets_use_prefetch=True --datasets_num_private_threads=4
--horovod_device=gpu >
/tmp/20190814_105450_tensorflow_horovod_rdma_resnet50_gpu_8_256_b500_ima
genet_nodistort_fp16_r10_m2_nockpt.txt 2>&1
```

4. **Opcional:** limpiar artefactos de trabajo. Los siguientes comandos de ejemplo muestran la eliminación del objeto de trabajo creado en el paso 1.

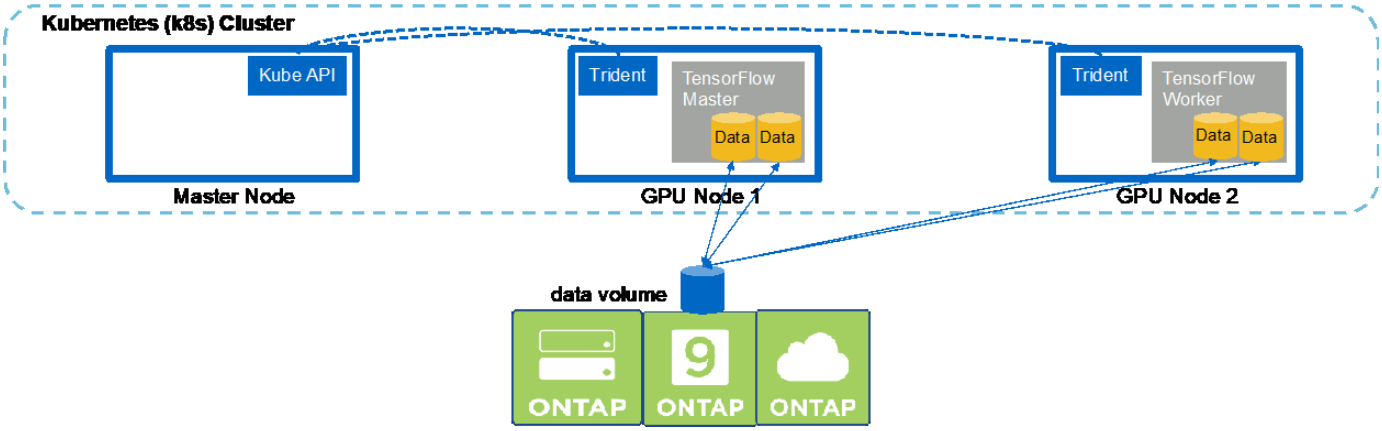
   Cuando se elimina el objeto de trabajo, Kubernetes elimina automáticamente todos los pods asociados.

```
$ kubectl get jobs
NAME                                                COMPLETIONS   DURATION
AGE
netapp-tensorflow-single-imagenet                   1/1           5m42s
10m
$ kubectl get pods
NAME                                                      READY   STATUS
RESTARTS   AGE
netapp-tensorflow-single-imagenet-m7x92                   0/1     Completed
0          11m
$ kubectl delete job netapp-tensorflow-single-imagenet
job.batch "netapp-tensorflow-single-imagenet" deleted
$ kubectl get jobs
No resources found.
$ kubectl get pods
No resources found.
```

Ejecute una carga de trabajo de IA distribuida síncrona

Para ejecutar un trabajo de IA y ML multinodo síncrono en un clúster de Kubernetes, lleve a cabo las siguientes tareas en el host de saltos de la puesta en marcha. Este proceso le permite aprovechar los datos almacenados en un volumen de NetApp y utilizar más GPU de las que puede proporcionar un único nodo de trabajo. Consulte la siguiente figura para obtener una descripción de un trabajo de IA distribuido sincrónica.

> Los trabajos distribuidos síncronos pueden ayudar a aumentar el rendimiento y la precisión de la formación en comparación con los trabajos distribuidos de manera asíncrona. Un análisis de los pros y los contras de los trabajos síncronos frente a los trabajos asincrónicos está fuera del alcance de este documento.



1. En los siguientes comandos de ejemplo, se muestra la creación de un trabajador que participa en la ejecución síncrona y distribuida de la misma tarea de prueba de rendimiento TensorFlow que se ejecutó en un solo nodo en el ejemplo de la sección "Ejecute una carga de trabajo de IA de un solo nodo". En este ejemplo específico, sólo se implementa un único trabajador porque el trabajo se ejecuta en dos nodos de

trabajo.

Esta puesta en marcha de trabajo de ejemplo solicita ocho GPU y, por lo tanto, puede ejecutarse en un único nodo de trabajo de GPU con ocho o más GPU. Si los nodos de trabajo de la GPU tienen más de ocho GPU, para maximizar el rendimiento, es posible que desee aumentar este número para ser igual al número de GPU que disponen los nodos de trabajo. Para obtener más información sobre las puestas en marcha de Kubernetes, consulte "Documentación oficial sobre Kubernetes".

En este ejemplo se crea una puesta en marcha de Kubernetes, ya que este trabajador en contenedor específico nunca lo completaría por sí solo. Por lo tanto, no tiene sentido implementarlo usando la construcción de trabajos de Kubernetes. Si su trabajador está diseñado o escrito para completar por sí solo, entonces podría tener sentido utilizar la construcción del trabajo para desplegar a su trabajador.

El POD especificado en esta especificación de implementación de ejemplo recibe una `hostNetwork` valor de `true`. Este valor significa que el pod utiliza la pila de red del nodo de trabajo del host en lugar de la pila de red virtual que Kubernetes suele crear para cada pod. Esta anotación se utiliza en este caso porque la carga de trabajo específica depende de Open MPI, NCCL y Horovod para ejecutar la carga de trabajo de forma síncrona distribuida. Por lo tanto, requiere acceso a la pila de red del host. Un debate sobre Open MPI, NCCL y Horovod está fuera del alcance de este documento. Si esto o no `hostNetwork: true` la anotación es necesaria depende de los requisitos de la carga de trabajo específica que se esté ejecutando. Para obtener más información acerca de `hostNetwork` consulte "Documentación oficial sobre Kubernetes".

```
$ cat << EOF > ./netapp-tensorflow-multi-imagenet-worker.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: netapp-tensorflow-multi-imagenet-worker
spec:
  replicas: 1
  selector:
    matchLabels:
      app: netapp-tensorflow-multi-imagenet-worker
  template:
    metadata:
      labels:
        app: netapp-tensorflow-multi-imagenet-worker
    spec:
      hostNetwork: true
      volumes:
      - name: dshm
        emptyDir:
          medium: Memory
      - name: testdata-iface1
        persistentVolumeClaim:
          claimName: pb-fg-all-iface1
      - name: testdata-iface2
        persistentVolumeClaim:
          claimName: pb-fg-all-iface2
```

```
      - name: results
        persistentVolumeClaim:
          claimName: tensorflow-results
      containers:
      - name: netapp-tensorflow-py2
        image: netapp/tensorflow-py2:19.03.0
        command: ["bash", "/netapp/scripts/start-slave-multi.sh",
"22122"]
        resources:
          limits:
            nvidia.com/gpu: 8
        volumeMounts:
        - mountPath: /dev/shm
          name: dshm
        - mountPath: /mnt/mount_0
          name: testdata-iface1
        - mountPath: /mnt/mount_1
          name: testdata-iface2
        - mountPath: /tmp
          name: results
        securityContext:
          privileged: true
EOF
$ kubectl create -f ./netapp-tensorflow-multi-imagenet-worker.yaml
deployment.apps/netapp-tensorflow-multi-imagenet-worker created
$ kubectl get deployments
NAME                                      DESIRED   CURRENT   UP-TO-DATE
AVAILABLE   AGE
netapp-tensorflow-multi-imagenet-worker   1         1         1
1           4s
```

2. Confirme que el despliegue del trabajador que creó en el paso 1 se inició correctamente. Los siguientes comandos de ejemplo confirman que se creó un solo pod de trabajadores para la implementación, tal y como se indica en la definición de la puesta en marcha, y que este pod se ejecuta actualmente en uno de los nodos de trabajo de la GPU.

```
$ kubectl get pods -o wide
NAME                                                      READY
STATUS    RESTARTS   AGE
IP              NODE            NOMINATED NODE
netapp-tensorflow-multi-imagenet-worker-654fc7f486-v6725   1/1
Running   0          60s   10.61.218.154   10.61.218.154   <none>
$ kubectl logs netapp-tensorflow-multi-imagenet-worker-654fc7f486-v6725
22122
```

3. Cree un trabajo de Kubernetes en un maestro que se inicia, participe y realice un seguimiento de la ejecución de un trabajo de varios nodos síncronos. Los siguientes comandos de ejemplo crean un maestro que inicia sesión, participa en y realiza un seguimiento de la ejecución síncrona distribuida de la misma tarea de prueba de rendimiento TensorFlow que se ejecutó en un solo nodo del ejemplo de la sección "Ejecute una carga de trabajo de IA de un solo nodo".

   Este trabajo maestro de ejemplo solicita ocho GPU y, por lo tanto, puede ejecutarse en un único nodo de trabajo de GPU con ocho o más GPU. Si los nodos de trabajo de la GPU tienen más de ocho GPU, para maximizar el rendimiento, es posible que desee aumentar este número para ser igual al número de GPU que disponen los nodos de trabajo.

   El POD maestro especificado en esta definición de trabajo de ejemplo recibe una `hostNetwork` valor de `true`, así como se le dio a la cápsula de trabajo un `hostNetwork` valor de `true` en el paso 1. Consulte el paso 1 para obtener más información acerca de por qué es necesario este valor.

```
$ cat << EOF > ./netapp-tensorflow-multi-imagenet-master.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: netapp-tensorflow-multi-imagenet-master
spec:
  backoffLimit: 5
  template:
    spec:
      hostNetwork: true
      volumes:
      - name: dshm
        emptyDir:
          medium: Memory
      - name: testdata-iface1
        persistentVolumeClaim:
          claimName: pb-fg-all-iface1
      - name: testdata-iface2
        persistentVolumeClaim:
          claimName: pb-fg-all-iface2
      - name: results
        persistentVolumeClaim:
          claimName: tensorflow-results
      containers:
      - name: netapp-tensorflow-py2
        image: netapp/tensorflow-py2:19.03.0
        command: ["python", "/netapp/scripts/run.py", "--
dataset_dir=/mnt/mount_0/dataset/imagenet", "--port=22122", "--
num_devices=16", "--dgx_version=dgx1", "--
nodes=10.61.218.152,10.61.218.154"]
        resources:
          limits:
            nvidia.com/gpu: 8
```

```
        volumeMounts:
        - mountPath: /dev/shm
          name: dshm
        - mountPath: /mnt/mount_0
          name: testdata-iface1
        - mountPath: /mnt/mount_1
          name: testdata-iface2
        - mountPath: /tmp
          name: results
        securityContext:
          privileged: true
      restartPolicy: Never
EOF
$ kubectl create -f ./netapp-tensorflow-multi-imagenet-master.yaml
job.batch/netapp-tensorflow-multi-imagenet-master created
$ kubectl get jobs
NAME                                        COMPLETIONS   DURATION   AGE
netapp-tensorflow-multi-imagenet-master   0/1           25s        25s
```

4. Confirme que el trabajo maestro que creó en el paso 3 se está ejecutando correctamente. El siguiente comando de ejemplo confirma que se creó un único pod maestro para el trabajo, tal como se indica en la definición de trabajos, y que este pod se ejecuta actualmente en uno de los nodos de trabajo de la GPU. También debe ver que el pod de trabajo que originalmente vio en el paso 1 sigue en ejecución y que los pods maestro y trabajador se ejecutan en diferentes nodos.

```
$ kubectl get pods -o wide
NAME                                                              READY
STATUS    RESTARTS   AGE
IP              NODE            NOMINATED NODE
netapp-tensorflow-multi-imagenet-master-ppwwj                     1/1
Running   0          45s   10.61.218.152   10.61.218.152   <none>
netapp-tensorflow-multi-imagenet-worker-654fc7f486-v6725   1/1
Running   0          26m   10.61.218.154   10.61.218.154   <none>
```

5. Confirme que el trabajo maestro que ha creado en el paso 3 se ha completado correctamente. Los siguientes comandos de ejemplo confirman que el trabajo se ha completado correctamente.

```
$ kubectl get jobs
NAME                                        COMPLETIONS   DURATION   AGE
netapp-tensorflow-multi-imagenet-master   1/1           5m50s      9m18s
$ kubectl get pods
NAME                                                              READY
STATUS      RESTARTS   AGE
netapp-tensorflow-multi-imagenet-master-ppwwj                     0/1
Completed   0          9m38s
```

```
netapp-tensorflow-multi-imagenet-worker-654fc7f486-v6725   1/1
Running     0          35m
$ kubectl logs netapp-tensorflow-multi-imagenet-master-ppwwj
[10.61.218.152:00008] WARNING: local probe returned unhandled
shell:unknown assuming bash
rm: cannot remove '/lib': Is a directory
[10.61.218.154:00033] PMIX ERROR: NO-PERMISSIONS in file gds_dstore.c at
line 702
[10.61.218.154:00033] PMIX ERROR: NO-PERMISSIONS in file gds_dstore.c at
line 711
[10.61.218.152:00008] PMIX ERROR: NO-PERMISSIONS in file gds_dstore.c at
line 702
[10.61.218.152:00008] PMIX ERROR: NO-PERMISSIONS in file gds_dstore.c at
line 711
Total images/sec = 12881.33875
================ Clean Cache !!! ==================
mpirun -allow-run-as-root -np 2 -H 10.61.218.152:1,10.61.218.154:1 -mca
pml ob1 -mca btl ^openib -mca btl_tcp_if_include enp1s0f0 -mca
plm_rsh_agent ssh -mca plm_rsh_args "-p 22122" bash -c 'sync; echo 1 >
/proc/sys/vm/drop_caches'
==========================================
mpirun -allow-run-as-root -np 16 -H 10.61.218.152:8,10.61.218.154:8
-bind-to none -map-by slot -x NCCL_DEBUG=INFO -x LD_LIBRARY_PATH -x PATH
-mca pml ob1 -mca btl ^openib -mca btl_tcp_if_include enp1s0f0 -x
NCCL_IB_HCA=mlx5 -x NCCL_NET_GDR_READ=1 -x NCCL_IB_SL=3 -x
NCCL_IB_GID_INDEX=3 -x
NCCL_SOCKET_IFNAME=enp5s0.3091,enp12s0.3092,enp132s0.3093,enp139s0.3094
-x NCCL_IB_CUDA_SUPPORT=1 -mca orte_base_help_aggregate 0 -mca
plm_rsh_agent ssh -mca plm_rsh_args "-p 22122" python
/netapp/tensorflow/benchmarks_190205/scripts/tf_cnn_benchmarks/tf_cnn_be
nchmarks.py --model=resnet50 --batch_size=256 --device=gpu
--force_gpu_compatible=True --num_intra_threads=1 --num_inter_threads=48
--variable_update=horovod --batch_group_size=20 --num_batches=500
--nodistortions --num_gpus=1 --data_format=NCHW --use_fp16=True
--use_tf_layers=False --data_name=imagenet --use_datasets=True
--data_dir=/mnt/mount_0/dataset/imagenet
--datasets_parallel_interleave_cycle_length=10
--datasets_sloppy_parallel_interleave=False --num_mounts=2
--mount_prefix=/mnt/mount_%d --datasets_prefetch_buffer_size=2000 --
datasets_use_prefetch=True --datasets_num_private_threads=4
--horovod_device=gpu >
/tmp/20190814_161609_tensorflow_horovod_rdma_resnet50_gpu_16_256_b500_im
agenet_nodistort_fp16_r10_m2_nockpt.txt 2>&1
```

6. Elimine la implementación del trabajador cuando ya no la necesite. Los siguientes comandos de ejemplo muestran la eliminación del objeto de implementación de trabajo que se creó en el paso 1.

Cuando se elimina el objeto de implementación de trabajo, Kubernetes elimina automáticamente todos los pods de trabajador asociados.

```
$ kubectl get deployments
NAME                                      DESIRED   CURRENT   UP-TO-DATE
AVAILABLE   AGE
netapp-tensorflow-multi-imagenet-worker   1         1         1
1           43m
$ kubectl get pods
NAME                                                        READY
STATUS      RESTARTS   AGE
netapp-tensorflow-multi-imagenet-master-ppwwj               0/1
Completed   0          17m
netapp-tensorflow-multi-imagenet-worker-654fc7f486-v6725   1/1
Running     0          43m
$ kubectl delete deployment netapp-tensorflow-multi-imagenet-worker
deployment.extensions "netapp-tensorflow-multi-imagenet-worker" deleted
$ kubectl get deployments
No resources found.
$ kubectl get pods
NAME                                            READY   STATUS
RESTARTS   AGE
netapp-tensorflow-multi-imagenet-master-ppwwj   0/1     Completed   0
18m
```

7. **Opcional:** Limpie los artefactos del trabajo maestro. Los siguientes comandos de ejemplo muestran la eliminación del objeto de trabajo maestro que se creó en el paso 3.

   Cuando se elimina el objeto de trabajo maestro, Kubernetes elimina automáticamente todos los pods maestros asociados.

```
$ kubectl get jobs
NAME                                      COMPLETIONS   DURATION   AGE
netapp-tensorflow-multi-imagenet-master   1/1           5m50s      19m
$ kubectl get pods
NAME                                            READY   STATUS
RESTARTS   AGE
netapp-tensorflow-multi-imagenet-master-ppwwj   0/1     Completed   0
19m
$ kubectl delete job netapp-tensorflow-multi-imagenet-master
job.batch "netapp-tensorflow-multi-imagenet-master" deleted
$ kubectl get jobs
No resources found.
$ kubectl get pods
No resources found.
```

Pruebas de rendimiento

Hemos realizado una comparativa simple del rendimiento como parte de la creación de esta solución. Hemos ejecutado varias tareas estándar de prueba de rendimiento de IA de NetApp mediante Kubernetes, y comparamos los resultados de las pruebas de rendimiento con ejecuciones realizadas mediante un sencillo comando Docker run. No observamos diferencias notables en cuanto al rendimiento. Por lo tanto, llegamos a la conclusión de que el uso de Kubernetes para orquestar trabajos de entrenamiento de IA en contenedores no afecta negativamente al rendimiento. Consulte la siguiente tabla para ver los resultados de nuestra comparativa de rendimiento.

Prueba de rendimiento	Conjunto de datos	Docker Run (imágenes/s)	Kubernetes (imágenes/s)
TensorFlow de un solo nodo	Datos sintéticos	6,667.2475	6,661.93125
TensorFlow de un solo nodo	ImageNET	6,570.2025	6,530.59125
TensorFlow distribuido síncrono de dos nodos	Datos sintéticos	13,213.70625	13,218.288125
TensorFlow distribuido síncrono de dos nodos	ImageNET	12,941.69125	12,881.33875

Conclusión

Empresas y organizaciones de todos los tamaños y sectores se están decantando por la inteligencia artificial (IA), el aprendizaje automático (ML) y el aprendizaje profundo (DL) para solucionar problemas del mundo real, ofrecer productos y servicios innovadores y obtener una ventaja en un mercado cada vez más competitivo. A medida que las organizaciones aumentan el uso de la IA, EL ML y el AP, deben hacer frente a numerosos retos, como la escalabilidad de la carga de trabajo y la disponibilidad de los datos. Estos desafíos se pueden abordar mediante el uso de la solución de plano de control de IA de NetApp.

Esta solución le permite clonar rápidamente un espacio de nombres de los datos. Además, le permite definir e implementar flujos de trabajo de entrenamiento de IA, ML y DL que incorporan la creación casi instantánea de bases de datos y modelos para su trazabilidad y versionado. Con esta solución, puede rastrear cada entrenamiento de modelo único que se ejecuta en los conjuntos de datos exactos con los que se ha entrenado el modelo y/o validado. Por último, esta solución le permite aprovisionar rápidamente espacios de trabajo de los portátiles Juppyter con acceso a conjuntos de datos masivos.

Dado que esta solución está dirigida a científicos e ingenieros de datos, se requiere una experiencia mínima en NetApp o ONTAP de NetApp. Con esta solución, las funciones de gestión de datos se pueden ejecutar utilizando interfaces y herramientas sencillas y conocidas. Además, esta solución utiliza componentes totalmente de código abierto y libres. Por lo tanto, si ya dispone de almacenamiento de NetApp en su entorno, puede implantar esta solución hoy mismo. Si desea probar esta solución pero no dispone de almacenamiento de NetApp, visite "cloud.netapp.com"Y puede tener una solución de almacenamiento de NetApp basada en cloud en un instante.

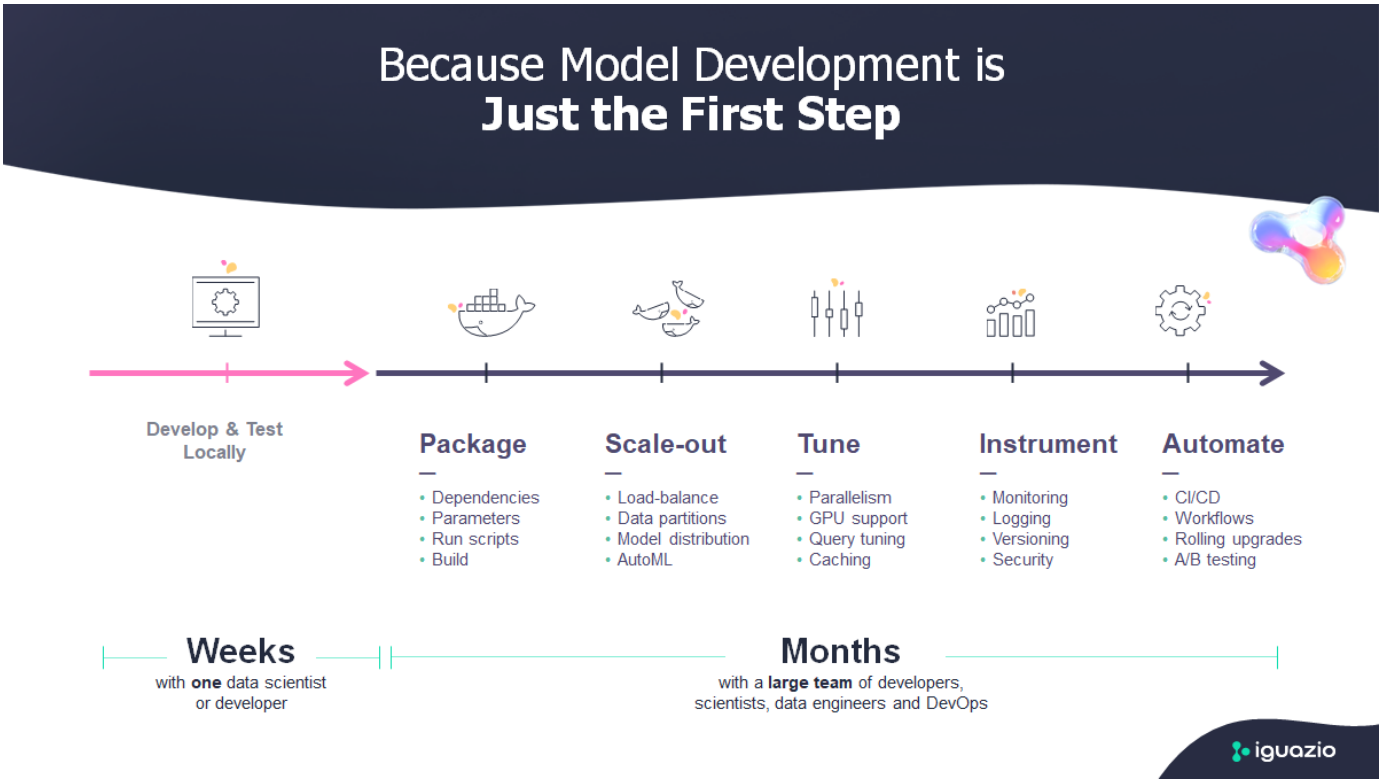
MLRun Pipeline con Iguazio

TR-4834: NetApp e Iguazio para el suministro de MLRun

Rick Huang, David Arnette, NetApp Marcelo Litovsky, Iguazio

Este documento contiene los detalles de la canalización MLRun usando NetApp ONTAP AI, el plano de control de IA de NetApp, el software Cloud Volumes de NetApp y la plataforma de ciencia de datos de Iguazio. Utilizamos la función Nuclio serverless, volúmenes persistentes de Kubernetes, volúmenes Cloud de NetApp, copias Snapshot de NetApp, consola Grafana, Y otros servicios en la plataforma Iguazio para crear una canalización de datos de extremo a extremo para la simulación de la detección de fallos de red. Hemos integrado las tecnologías de Iguazio y NetApp para permitir una rápida puesta en marcha de modelos, replicación de datos y supervisión de la producción tanto en las instalaciones como en el cloud.

El trabajo de un científico de datos debe centrarse en el entrenamiento y el ajuste de los modelos de aprendizaje automático y de inteligencia artificial (IA). Sin embargo, según una investigación realizada por Google, los científicos de datos gastan aproximadamente un 80 % de su tiempo en averiguar cómo hacer que sus modelos funcionen con aplicaciones empresariales y se ejecutan a escala, como se muestra en la siguiente imagen, en la que se ilustra el desarrollo de modelos en el flujo de trabajo de IA/ML.



Para gestionar proyectos de IA y ML integrales, se necesita una comprensión más amplia de los componentes empresariales. Aunque DevOps ha retomado la definición, la integración y la puesta en marcha de estos tipos de componentes, las operaciones de aprendizaje automático siguen un flujo similar que incluye proyectos de IA/ML. Para hacerse una idea de lo que puede tocar una canalización de IA/ML integral en la empresa, consulte la siguiente lista de componentes requeridos:

- Reducida
- Redes
- Oracle
- Sistemas de ficheros
- Contenedores
- Integración continua y canalización de puesta en marcha continua (CI/CD)
- Entorno de desarrollo integrado (IDE)
- Seguridad
- Políticas de acceso a los datos
- Hardware subyacente
- Cloud
- Virtualización
- Conjuntos de herramientas y bibliotecas de ciencia de datos

En este artículo mostramos cómo la asociación entre NetApp y Iguazio simplifica drásticamente el desarrollo de una canalización completa de IA/ML. Esta simplificación acelera el plazo de comercialización de todas sus aplicaciones de IA/ML.

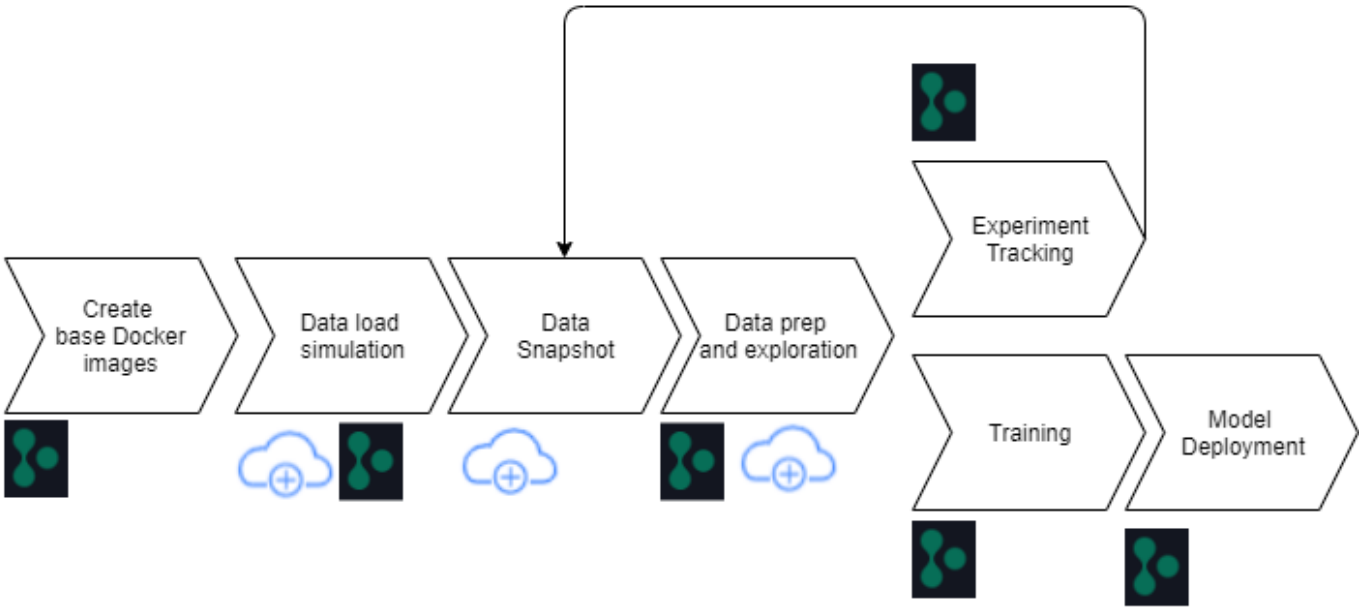
Público objetivo

El mundo de la ciencia de datos tiene múltiples disciplinas de tecnología de la información y negocio.

- El científico de datos necesita la flexibilidad para poder usar las herramientas y las bibliotecas de elección.
- El ingeniero de datos necesita saber cómo fluyen los datos y dónde residen.
- Un ingeniero de DevOps necesita herramientas para integrar nuevas aplicaciones de IA/ML en sus canalizaciones de CI/CD.
- Los usuarios empresariales quieren tener acceso a aplicaciones de IA/ML. Describimos cómo NetApp y Iguazio ayudan a cada una de estas funciones a aportar valor empresarial con nuestras plataformas.

Descripción general de la solución

Esta solución sigue el ciclo de vida de una aplicación de IA/ML. Empezamos con la labor de los científicos de datos para definir los diferentes pasos necesarios para preparar datos y entrenar y poner en marcha modelos. Seguimos con el trabajo necesario para crear una canalización completa con la capacidad de realizar un seguimiento de los artefactos, experimentar con la ejecución y desplegar en Kubeflow. Para completar el ciclo completo, integramos la canalización con NetApp Cloud Volumes para permitir el versionado de datos, tal y como se puede ver en la siguiente imagen.



iguazio

Cloud Volumes ONTAP

Visión general de la tecnología

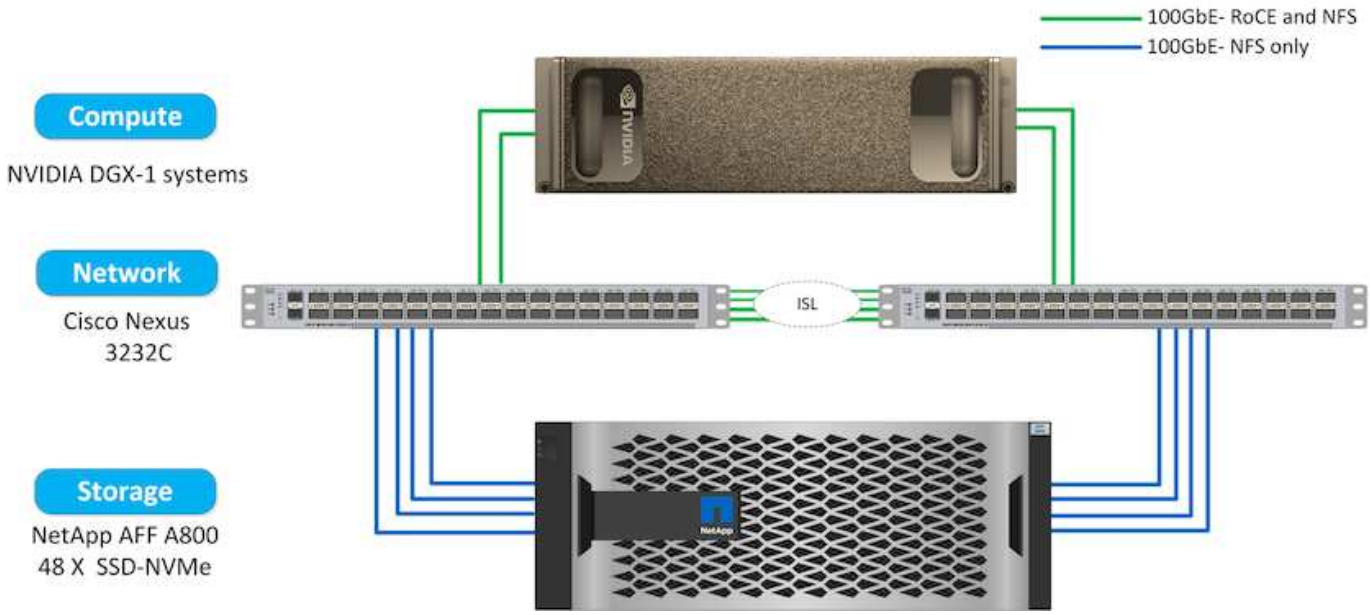
Información general de NetApp

NetApp es un referente en materia de datos para el cloud híbrido. NetApp proporciona una gama completa de servicios de datos en el cloud híbrido que simplifican la gestión de aplicaciones y datos en entornos de cloud y en las instalaciones para acelerar la transformación digital. Junto con nuestros partners, NetApp permite a organizaciones globales aprovechar al máximo todo el potencial de sus datos para ampliar los puntos de contacto con los clientes, fomentar una mayor innovación y optimizar sus operaciones.

ONTAP AI de NetApp

ONTAP AI de NetApp, impulsado por los sistemas NVIDIA DGX y el almacenamiento all-flash conectado al cloud de NetApp, optimiza el flujo de datos con total confianza y acelera el análisis, la formación y la inferencia con su Data Fabric, que abarca desde el perímetro al núcleo y al cloud. Proporciona a las organizaciones DE TI una arquitectura que ofrece las siguientes ventajas:

- Elimina las complejidades de diseño
- Permite un escalado independiente de las capacidades de computación y almacenamiento
- Permite a los clientes empezar con poco y escalar sin problemas
- Ofrece una gama de opciones de almacenamiento para diferentes tipos de rendimiento y coste.ONTAP AI ofrece pilas de infraestructuras convergentes que incorporan NVIDIA DGX-1, un sistema de IA a escala de petaflops y switches Ethernet de alto rendimiento NVIDIA Mellanox para unificar cargas de trabajo de IA, simplificar la puesta en marcha y acelerar el retorno de la inversión. Para este informe técnico, hemos aprovechado la inteligencia artificial de ONTAP con un sistema de almacenamiento DGX-1 y AFF A800 de NetApp. En la siguiente imagen, se muestra la topología de ONTAP AI con el sistema DGX-1 utilizado en esta validación.



Plano de control de IA de NetApp

El plano de control de IA de NetApp permite aprovechar la IA Y EL APRENDIZAJE AUTOMÁTICO con una solución que ofrece una escalabilidad extrema, una puesta en marcha optimizada y una disponibilidad de datos ininterrumpida. La solución de plano de control de IA integra Kubernetes y Kubeflow con una estructura de datos habilitada por NetApp. Kubernetes, la plataforma de orquestación de contenedores estándar del sector para puestas en marcha nativas del cloud, permite la escalabilidad y la portabilidad de las cargas de trabajo. Kubeflow es una plataforma de aprendizaje automático de código abierto que simplifica la gestión y la puesta en marcha, lo que permite a los desarrolladores hacer más ciencia de datos en menos tiempo. Una estructura de datos habilitada por NetApp ofrece una disponibilidad y portabilidad de datos sin concesiones para garantizar que sus datos están accesibles en toda la canalización, desde el perímetro al núcleo y al cloud. En este informe técnico se utiliza el plano de control de IA de NetApp en una canalización MLRun. La siguiente imagen muestra la página de gestión de clústeres de Kubernetes, donde se pueden tener distintos extremos para cada clúster. Conectamos volúmenes persistentes de NFS al clúster de Kubernetes, y las siguientes imágenes muestran un volumen persistente conectado al clúster donde "Trident de NetApp" ofrece compatibilidad con almacenamiento persistente y capacidades de gestión de datos.

Kubernetes Clusters

Discover Cluster

4 Kubernetes Clusters

kubernetes

https://3.20.111.39:6443
Cluster Endpoint

v1.15.5
Cluster Version

19.07.1
Trident Version

0
Working Environments

kubernetes

https://172.31.14.31:6443
Cluster Endpoint

v1.15.5
Cluster Version

19.07.1
Trident Version

1
Working Environments

Persistent Volumes for Kubernetes

Connected with Kubernetes Cluster

Cloud Volumes ONTAP is connected to 1 Kubernetes cluster. View Cluster

You can connect another Kubernetes cluster to this Cloud Volumes ONTAP system. If the Kubernetes cluster is in a different network than Cloud Volumes ONTAP, specify a custom export policy to provide access to clients.

Kubernetes Cluster

Select Kubernetes Cluster

kubernetes

Custom Export Policy *(Optional)*

Custom Export Policy

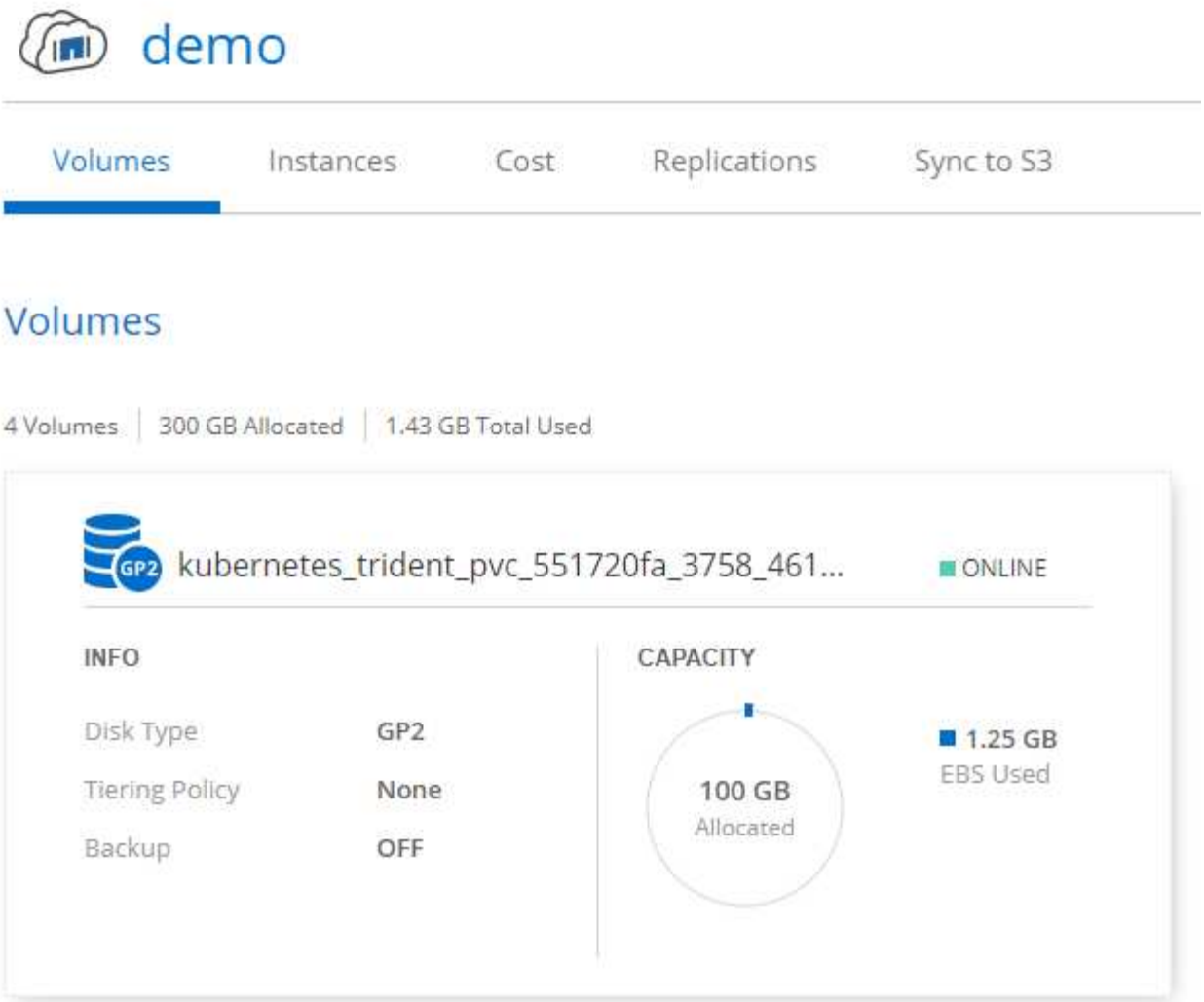
172.31.0.0/16

☑ Set as default storage class

◉ NFS ○ iSCSI

Connect

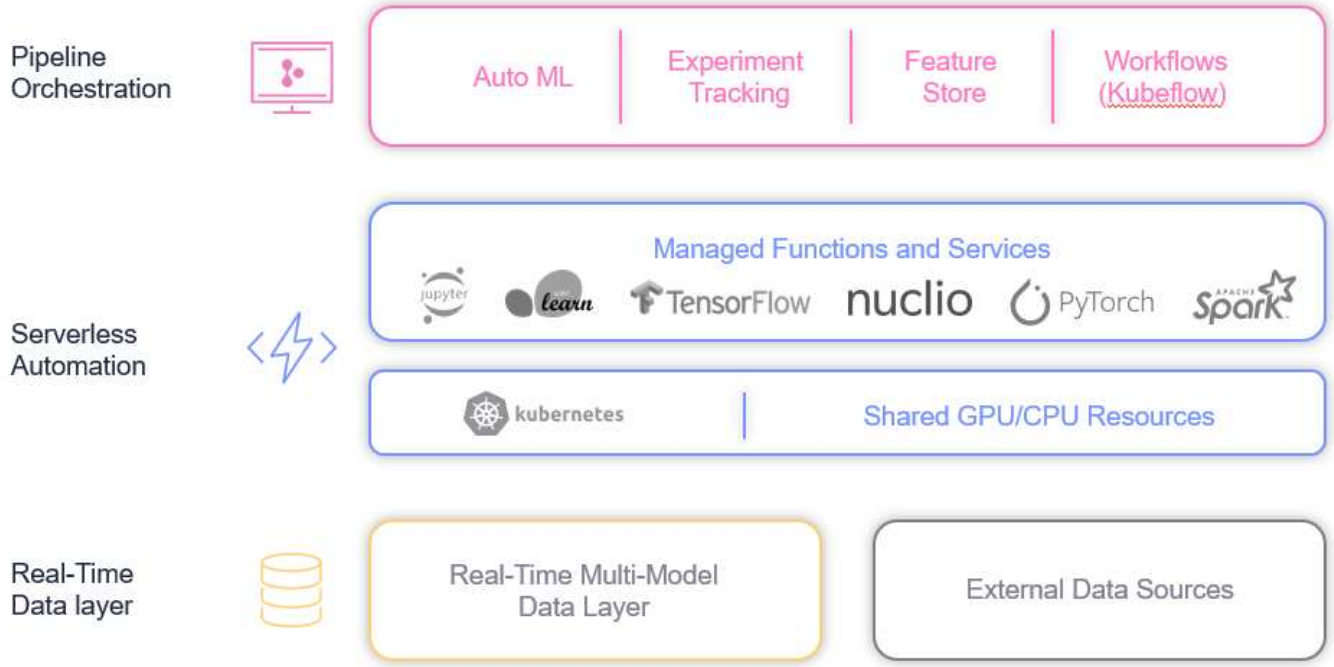
Cancel



Visión General de Iguazio

La Plataforma de Ciencias de datos de Iguazio es una plataforma de ciencia de datos como servicio (PaaS) totalmente integrada y segura que simplifica el desarrollo, acelera el rendimiento, facilita la colaboración y aborda los retos operativos. Esta plataforma incorpora los siguientes componentes, y la Plataforma de Ciencias de datos de Iguazio se presenta en la siguiente imagen:

- Banco de trabajo de ciencia de datos que incluye portátiles Juppyter, motores de análisis integrados y paquetes Python
- Gestión de modelos con seguimiento de experimentos y funcionalidades de canalizaciones automatizadas
- Datos gestionados y servicios ML a través de un clúster de Kubernetes escalable
- Nuclio, un marco de funciones sin servidor en tiempo real
- Una capa de datos extremadamente rápida y segura compatible con SQL, NoSQL, bases de datos de series temporales, archivos (objetos sencillos) y streaming
- Integración con fuentes de datos de terceros como NetApp, Amazon S3, HDFS, bases de datos de SQL y protocolos de transmisión o mensajería
- Paneles en tiempo real basados en Grafana



Requisitos de software y hardware

Configuración de red

A continuación, se muestra el requisito de configuración de red para configurar en el cloud:

- El clúster de Iguazio y los volúmenes cloud de NetApp deben estar en el mismo cloud privado virtual.
- El administrador de la nube debe tener acceso al puerto 6443 en los nodos de la aplicación Iguazio.
- Utilizamos Amazon Web Services en este informe técnico. Sin embargo, los usuarios tienen la opción de poner en marcha la solución en cualquier proveedor de cloud.para las pruebas en las instalaciones en IA de ONTAP con NVIDIA DGX-1, hemos utilizado el servicio DNS alojado en Iguazio para mayor comodidad.

Los clientes deben poder acceder a los dominios DNS creados de forma dinámica. Los clientes pueden utilizar su propio DNS si lo desean.

Requisitos de hardware

Puede instalar Iguazio en sus instalaciones en su propio clúster. Hemos verificado la solución en ONTAP AI de NetApp con un sistema NVIDIA DGX-1. La siguiente tabla enumera el hardware utilizado para probar esta solución.

Hardware subyacente	Cantidad
Sistemas DGX-1	1
Sistema AFF A800 de NetApp	1 pareja de alta disponibilidad (ha), incluye 2 controladoras y 48 SSD NVMe (3,8 TB o superior)
Switches de red Cisco Nexus 3232C	2

En la siguiente tabla se enumeran los componentes de software necesarios para las pruebas in situ:

De NetApp	Versión u otra información
Software de gestión de datos ONTAP de NetApp	9.7
Firmware de switch Cisco NX-OS	7.0(3)I6(1)
SO DGX DE NVIDIA	4.4 - Ubuntu 18.04 LTS
Plataforma contenedora Docker	19.03.5
Versión del contenedor	20.01-tf1-py2
Marco de aprendizaje automático	TensorFlow 1.15.0
Iguazio	La versión es 2.8 o posterior
Servidor ESX	6.5

Esta solución se ha probado totalmente con Iguazio versión 2.5 y NetApp Cloud Volumes ONTAP para AWS. El clúster de Iguazio y el software de NetApp se ejecutan en AWS.

De NetApp	Versión o tipo
Iguazio	La versión es 2.8 o posterior
Nodo de aplicación	M5.4xgrande
Nodo de datos	I3.4xgrande

Resumen de casos de uso de predicción de fallos de dispositivos de red

Este caso de uso se basa en un cliente de Iguazio en el espacio de telecomunicaciones en Asia. Con 100 000 clientes empresariales y 125 000 eventos de interrupción de la red al año, era muy importante predecir y tomar medidas proactivas para evitar que los fallos de red afecten a los clientes. Esta solución ofreció las siguientes ventajas:

- Análisis predictivo para fallos de red
- Integración con un sistema de emisión de boletos
- Tomar medidas proactivas para evitar fallos de la red como resultado de esta implementación de Iguazio, el 60 % de los fallos se impidieron de forma proactiva.

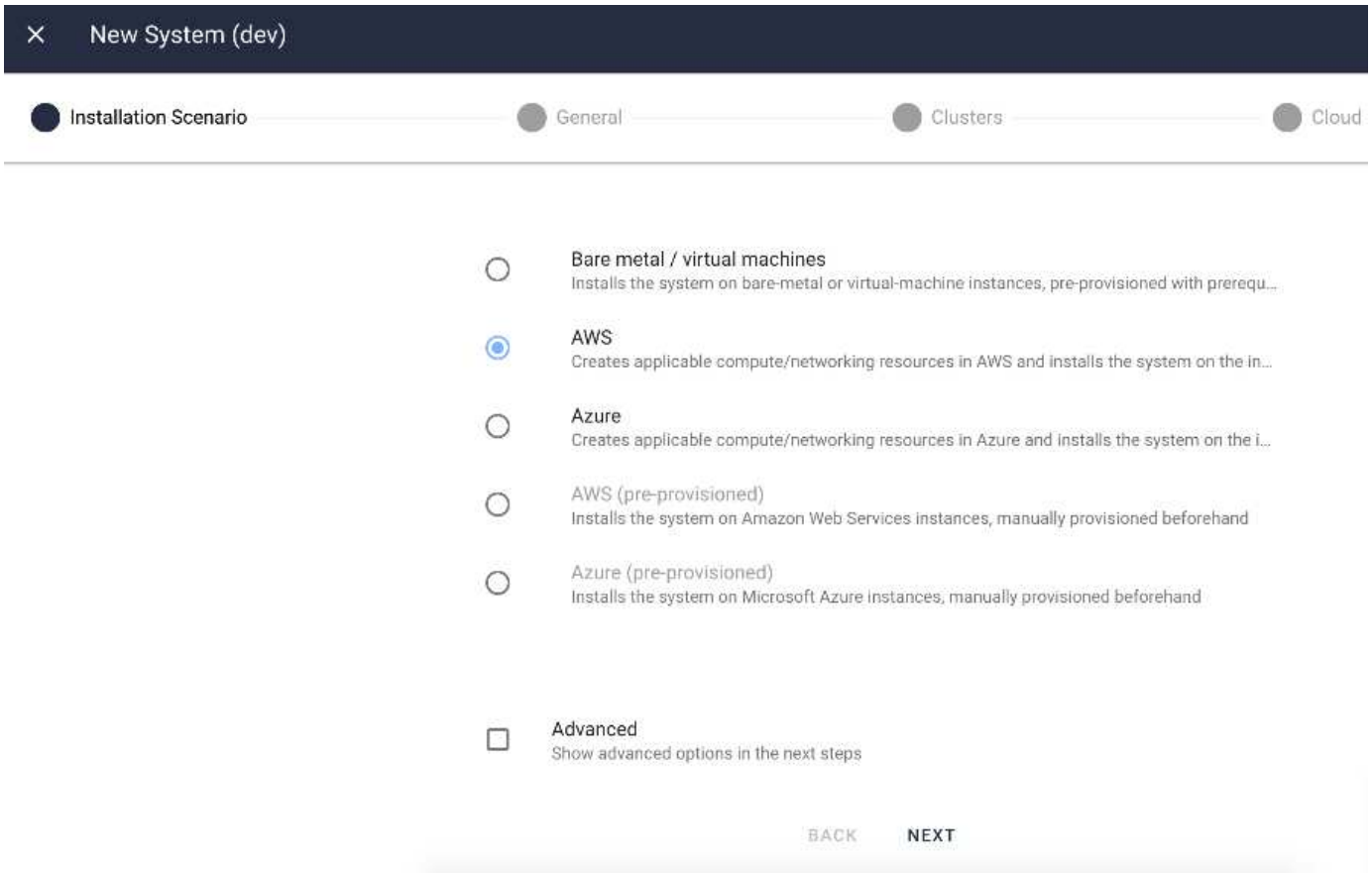
Descripción general de la configuración

Iguazio se puede instalar de forma local o en un proveedor de cloud.

Instalación de Iguazio

El aprovisionamiento puede realizarse como servicio y gestionado por Iguazio o por el cliente. En ambos casos, Iguazio ofrece una aplicación de despliegue (Proventio) para implementar y gestionar clústeres.

Para la instalación en las instalaciones, consulte "NVA-1121" para la configuración de computación, red y almacenamiento. La implantación en las instalaciones de Iguazio es proporcionada por Iguazio sin costes adicionales para el cliente. Consulte "esta página" Para configuraciones de servidores DNS y SMTP. La página de instalación de Proventio se muestra a continuación.



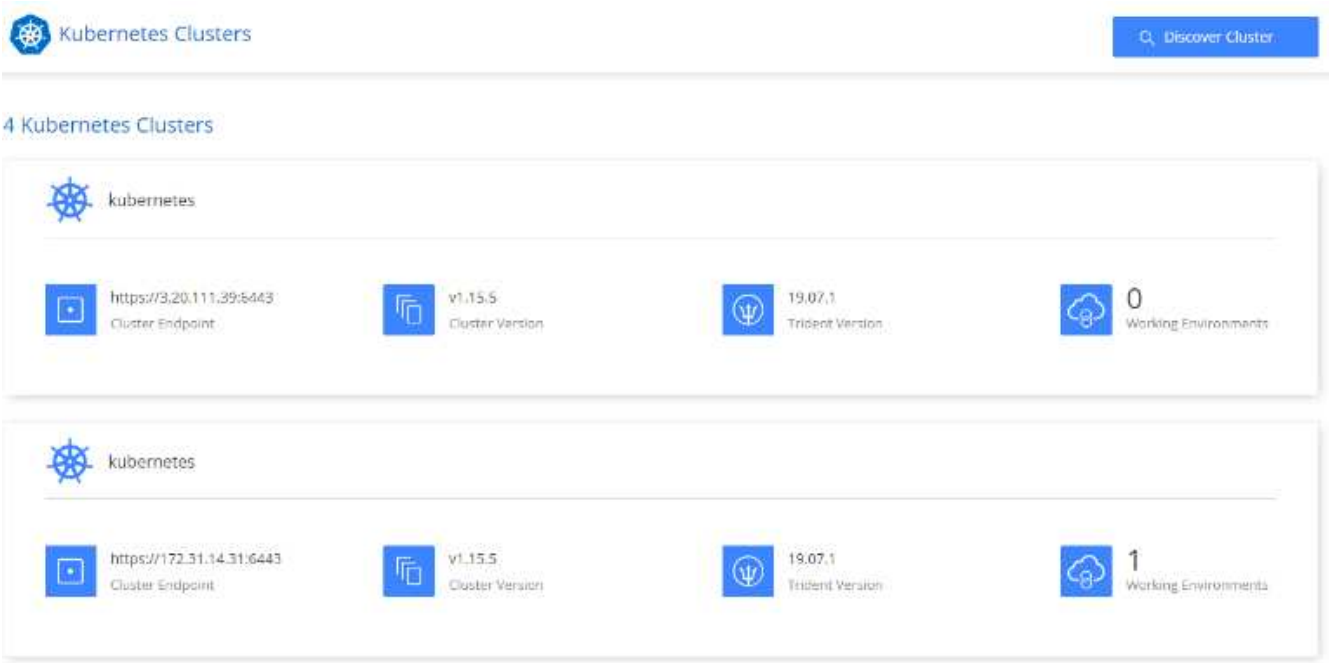
Configurar Kubernetes Cluster

Esta sección se divide en dos partes para la puesta en marcha en el cloud y en las instalaciones, respectivamente.

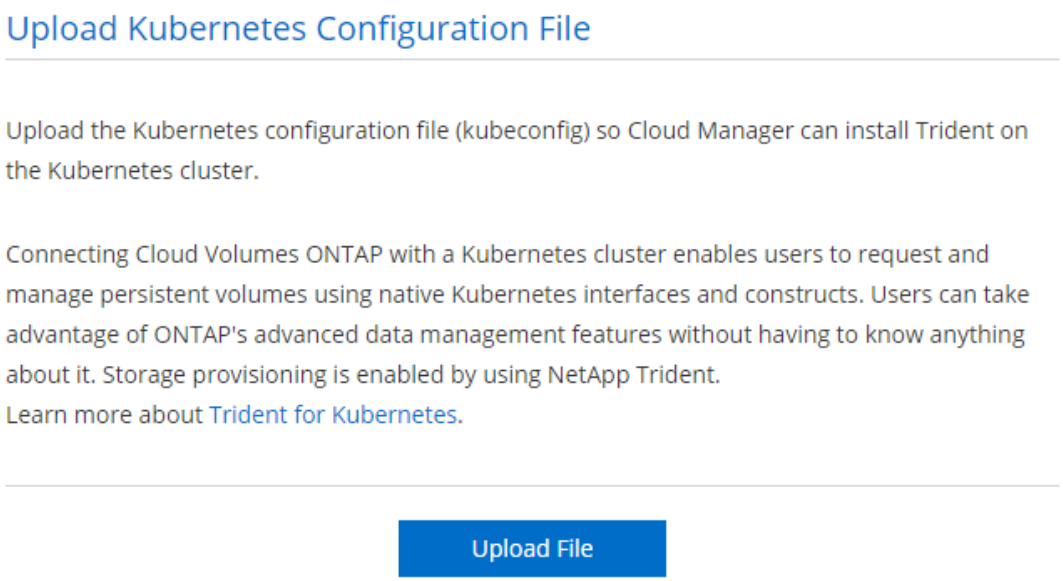
Configuración de Kubernetes de puesta en marcha del cloud

A través de Cloud Manager de NetApp, puede definir la conexión al clúster de Kubernetes del Iguazio. Trident requiere acceso a varios recursos en el clúster para que el volumen esté disponible.

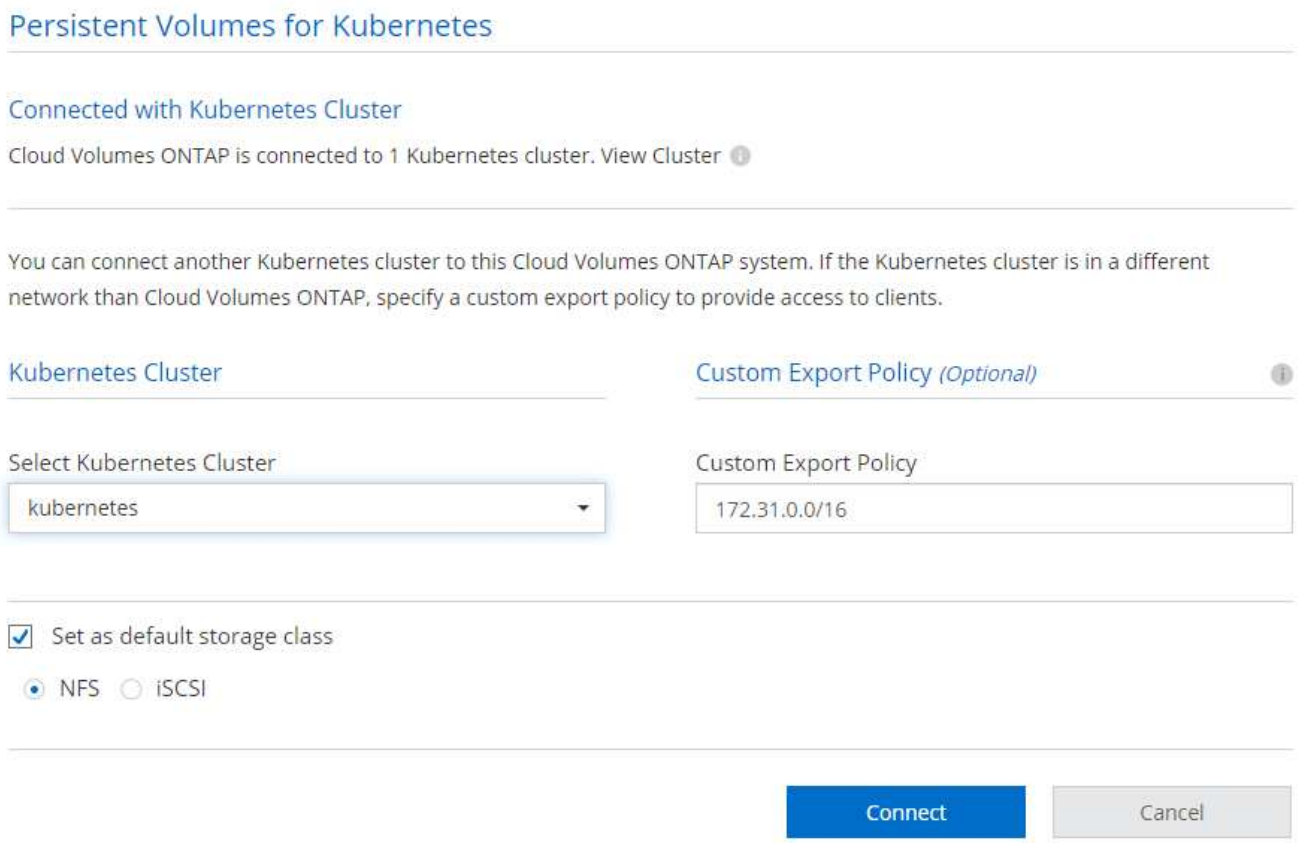
1. Para habilitar el acceso, obtenga el archivo de configuración de Kubernetes desde uno de los nodos de Iguazio. El archivo está ubicado en `/home/Iguazio/.kube/config`. Descargue este archivo en su escritorio.
2. Vaya a detectar clúster para configurar.



3. Cargue el archivo de configuración de Kubernetes. Consulte la siguiente imagen.



4. Ponga en marcha Trident y asocie un volumen con el clúster. Vea la siguiente imagen sobre la definición y asignación de un volumen persistente al clúster de Iguazio. Este proceso crea un volumen persistente (PV) en el clúster Kubernetes de Iguazio. Antes de poder usarlo, debe definir una solicitud de volumen persistente (PVC).



Configuración de Kubernetes de puesta en marcha en las instalaciones

Para ver la instalación local de Trident de NetApp, consulte "TR-4798" para obtener más detalles. Tras configurar su clúster de Kubernetes e instalar Trident de NetApp, puede conectar Trident al clúster Iguazio para habilitar las capacidades de gestión de datos de NetApp, como realizar copias Snapshot de sus datos y su modelo.

Defina una solicitud de volumen persistente

1. Guarde el siguiente YAML en un archivo para crear un PVC del tipo Basic.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
  storageClassName: netapp-file
```

2. Aplique el archivo YLMA al clúster de Kubernetes de Iguazio.

```
Kubectl -n default-tenant apply -f <your yaml file>
```

Adjunte el volumen de NetApp al ordenador portátil Jupyter

Iguazio ofrece varios servicios gestionados para proporcionar a los científicos de datos una pila completa para el desarrollo y la puesta en marcha de aplicaciones de IA/ML. Puede obtener más información sobre estos componentes en la "Visión General de Iguazio de Servicios y Herramientas de aplicación".

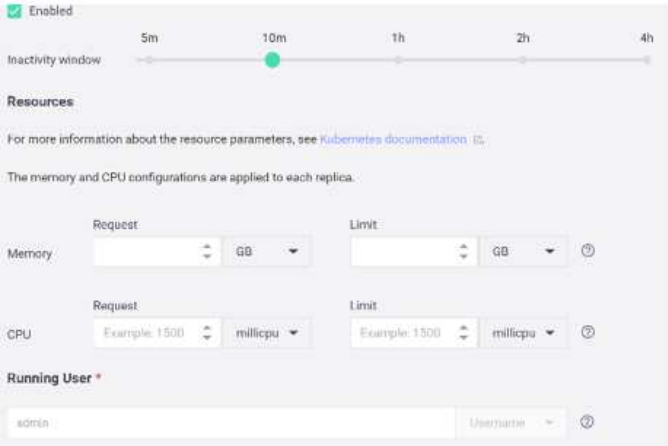
Uno de los servicios gestionados es el portátil Jupyter. Cada desarrollador obtiene su propia implementación de un contenedor de portátiles con los recursos que necesitan para el desarrollo. Para darles acceso a Cloud Volume de NetApp, puede asignar el volumen a su contenedor y la asignación de recursos, la configuración de usuario en ejecución y la variable de entorno para las reclamaciones de volumen persistente se presentan en la siguiente imagen.

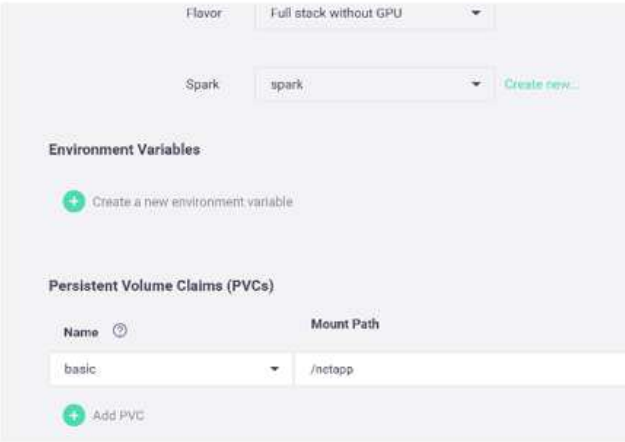
Para obtener información sobre una configuración en las instalaciones, puede consultar "TR-4798" En la configuración de Trident para habilitar las funcionalidades de gestión de datos de ONTAP de NetApp, como realizar copias Snapshot de sus datos o modelos para el control de versiones. Añada la siguiente línea en su archivo de configuración de back-end de Trident para que sean visibles los directorios de Snapshot:

```
{
    …
    "defaults": {
        "snapshotDir": "true"
    }
}
```

Debe crear un archivo de configuración back-end de Trident en formato JSON y, a continuación, ejecutar lo siguiente "Comando Trident" como referencia:

```
tridentctl create backend -f <backend-file>
```





Implementación de la aplicación

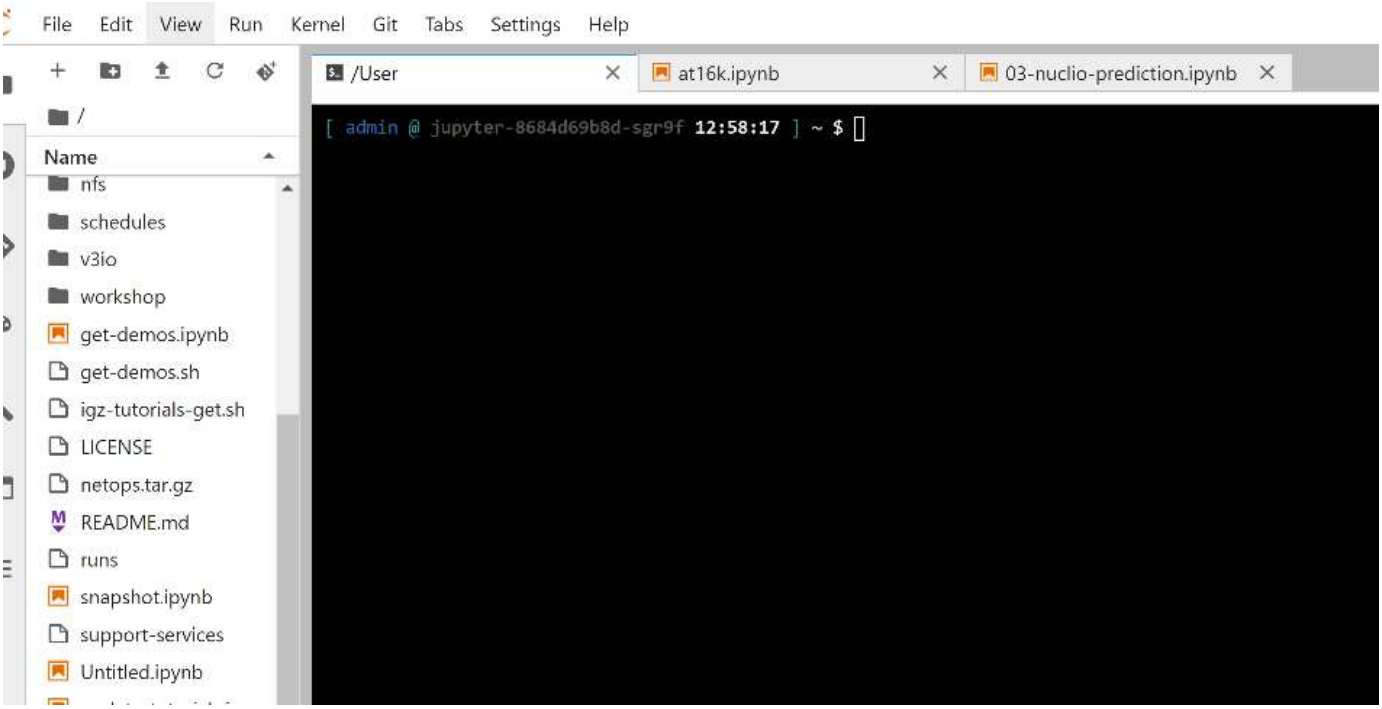
En las siguientes secciones se describe cómo instalar e implementar la aplicación.

Obtenga el código de GitHub

Ahora que el volumen de cloud de NetApp o el volumen de Trident de NetApp están disponibles para el clúster de Iguazio y el entorno de desarrolladores, puede comenzar a revisar la aplicación.

Los usuarios tienen su propio espacio de trabajo (directorio). En cada notebook, la ruta al directorio de usuario es `/User`. La plataforma Iguazio administra el directorio. Si sigue las instrucciones anteriores, el volumen de cloud de NetApp está disponible en `/netapp` directorio.

Obtenga el código de GitHub con una terminal Juppyter.



En el prompt de la terminal de Juppyter, clone el proyecto.

```
cd /User
git clone .
```

Ahora debería ver la `netops- netapp` Carpeta en el árbol de archivos del espacio de trabajo Juppyter.

Configurar entorno de trabajo

Copie el `Notebook set_env-Example.ipynb` como `set_env.ipynb`. Abrir y editar `set_env.ipynb`. Este cuaderno establece variables para las credenciales, ubicaciones de archivos y controladores de ejecución.

Si sigue las instrucciones anteriores, los pasos siguientes son los únicos cambios que se deben realizar:

1. Obtenga este valor desde el panel de servicios de Iguazio: `docker_registry`

   Ejemplo: `docker-registry.default-tenant.app.clusterq.iguaziodev.com:80`

2. Cambiar `admin` A su nombre de usuario de Iguazio:

   ```
   IGZ_CONTAINER_PATH = '/users/admin'
   ```

   A continuación, se muestran los detalles de conexión del sistema ONTAP. Incluya el nombre del volumen que se generó al instalar Trident. La siguiente configuración es para un clúster ONTAP en las instalaciones:

   ```
   ontapClusterMgmtHostname = '0.0.0.0'
   ontapClusterAdminUsername = 'USER'
   ontapClusterAdminPassword = 'PASSWORD'
   sourceVolumeName = 'SOURCE VOLUME'
   ```

   La siguiente configuración es para Cloud Volumes ONTAP:

```
MANAGER=ontapClusterMgmtHostname
svm='svm'
email='email'
password=ontapClusterAdminPassword
weid="weid"
volume=sourceVolumeName
```

Cree imágenes Docker básicas

Todo lo que necesita para construir un ducto ML está incluido en la plataforma Iguazio. El desarrollador puede definir las especificaciones de las imágenes Docker necesarias para ejecutar la canalización y ejecutar la creación de imágenes desde el portátil Jupyter. Abra el portátil `create- images.ipynb` Y ejecute todas las celdas.

Este cuaderno crea dos imágenes que utilizamos en la canalización.

- `iguazio/netapp.` Se utiliza para manejar tareas ML.

Create image for training pipeline

```
[4]: fn.build_config(image=docker_registry+'/iguazio/netapp', commands=['pip install \
     v3io_frames fsspec>=0.3.3 PyYAML==5.1.2 pyarrow==0.15.1 pandas==0.25.3 matplotlib seaborn yellowb
     fn.deploy()
```

- `netapp/pipeline`. Contiene utilidades para gestionar las copias snapshot de NetApp.

Create image for Ontap utilitites

```
[9]: fn.build_config(image=docker_registry + '/netapp/pipeline:latest',commands=['apt -y update','pip install v3io_frames netapp_ontap'
     fn.deploy()
```

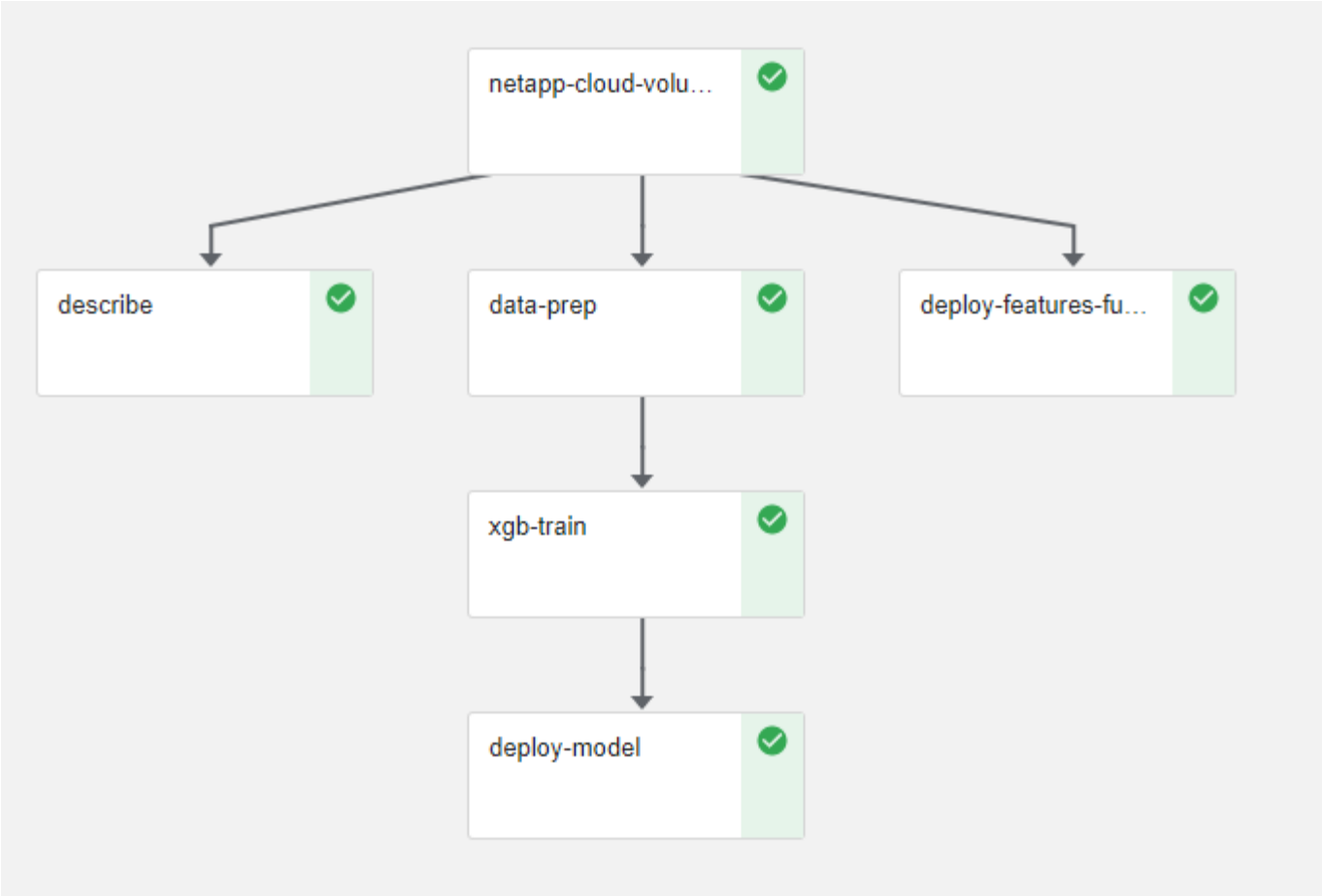
Revise los cuadernos Juppyter individuales

En la siguiente tabla se enumeran las bibliotecas y los marcos que utilizamos para crear esta tarea. Todos estos componentes se han integrado plenamente con los controles de seguridad y acceso basados en funciones de Iguazio.

Bibliotecas/Marco	Descripción
MLRun	Un gestionado por Iguazio para permitir el montaje, ejecución y supervisión de una canalización ML/IA.
Nuclio	Un marco de funciones sin servidor integrado con Iguazio. También disponible como proyecto de código abierto gestionado por Iguazio.
Kubeflow	Un marco basado en Kubernetes para poner en marcha la canalización. Este es también un proyecto de código abierto al que colabora Iguazio. Se integra con Iguazio para una mayor seguridad e integración con el resto de la infraestructura.
Docker	Un registro Docker se ejecuta como servicio en la plataforma Iguazio. También puede cambiar esta opción para conectarse al registro.
Cloud Volumes de NetApp	Cloud Volumes que se ejecuta en AWS nos proporciona acceso a grandes cantidades de datos y la capacidad de realizar copias de Snapshot para versiones de los conjuntos de datos utilizados para el entrenamiento.
Trident	Trident es un proyecto de código abierto gestionado por NetApp. Facilita la integración con los recursos de almacenamiento y computación en Kubernetes.

Hemos utilizado varios portátiles para construir la canalización DE ML. Cada portátil puede probarse individualmente antes de ser reunido en la tubería. Cubrimos cada portátil individualmente tras el flujo de despliegue de esta aplicación de demostración.

El resultado deseado es una canalización que entrena un modelo basado en una copia Snapshot de los datos y pone en marcha el modelo para la inferencia. En la siguiente imagen se muestra un diagrama de bloque de una canalización MLRun completada.



Ponga en marcha la función de generación de datos

En esta sección se describe cómo utilizamos las funciones de Nuclio serverless para generar datos de dispositivos de red. El caso de uso se adapta a partir de un cliente Iguazio que ha implementado la canalización y utiliza servicios Iguazio para supervisar y predecir los fallos del dispositivo de red.

Simulamos los datos procedentes de dispositivos de red. Ejecución del cuaderno Juppyter `data-generator.ipynb` Crea una función sin servidor que se ejecuta cada 10 minutos y genera un archivo Parquet con nuevos datos. Para implementar la función, ejecute todas las celdas de este portátil. Consulte "Sitio Web de Nuclio" para revisar cualquier componente desconocido en este cuaderno.

Al generar la función, se ignora una celda con el siguiente comentario. Se asume que todas las celdas del portátil forman parte de la función. Importe el módulo Nuclio para activarlo `%nuclio magic`.

```
# nuclio: ignore
import nuclio
```

En la especificación para la función, definimos el entorno en el que se ejecuta la función, cómo se activa y los recursos que consume.

```
spec = nuclio.ConfigSpec(config={"spec.triggers.inference.kind":"cron",

"spec.triggers.inference.attributes.interval" :"10m",
                                "spec.readinessTimeoutSeconds" : 60,
                                "spec.minReplicas" : 1},……
```

La `init_context` La función es invocada por el marco Nuclio tras la inicialización de la función.

```
def init_context(context):
    ….
```

Cualquier código que no esté en una función se invoca cuando se inicializa la función. Cuando lo invoca, se ejecuta una función de controlador. Puede cambiar el nombre del controlador y especificarlo en la especificación de función.

```
def handler(context, event):
            …
```

Puede probar la función desde el portátil antes de la implementación.

```
%%time
# nuclio: ignore
init_context(context)
event = nuclio.Event(body='')
output = handler(context, event)
output
```

La función puede desplegarse desde el portátil o puede desplegarse a partir de una canalización CI/CD (adaptando este código).

```
addr = nuclio.deploy_file(name='generator',project='netops',spec=spec,
tag='v1.1')
```

Cuadernos de pipeline

Estos cuadernos no están diseñados para ejecutarse individualmente para esta configuración. Esto es sólo una revisión de cada bloc de notas. Los invocamos como parte del proyecto. Para ejecutarlas individualmente, revise la documentación MLRun para ejecutarlas como trabajos de Kubernetes.

snap_cv.ipynb

Este portátil gestiona las copias snapshot de Cloud Volume al principio de la canalización. Pasa el nombre del volumen al contexto de la canalización. Este cuaderno invoca un script de shell para manejar la copia

Snapshot. Mientras se ejecuta en la canalización, el contexto de ejecución contiene variables que ayudan a localizar todos los archivos necesarios para su ejecución. Mientras escribe este código, el desarrollador no tiene que preocuparse por la ubicación del archivo en el contenedor que lo ejecuta. Como se describe más tarde, esta aplicación se implementa con todas sus dependencias y es la definición de los parámetros de canalización que proporciona el contexto de ejecución.

```
command = os.path.join(context.get_param('APP_DIR'),"snap_cv.sh")
```

La ubicación de la copia Snapshot creada se coloca en el contexto de MLRun que consumirán los pasos de la canalización.

```
context.log_result('snapVolumeDetails',snap_path)
```

Los siguientes tres portátiles se ejecutan en paralelo.

data-prep.ipynb

Las métricas sin formato deben convertirse en funciones para permitir el entrenamiento de modelos. Este cuaderno lee las métricas sin formato del directorio Snapshot y escribe las funciones de entrenamiento de modelos en el volumen de NetApp.

Cuando se ejecuta en el contexto de la canalización, la entrada `DATA_DIR` Contiene la ubicación de la copia Snapshot.

```
metrics_table = os.path.join(str(mlruncontext.get_input('DATA_DIR',
os.getenv('DATA_DIR','/netpp'))),
                            mlruncontext.get_param('metrics_table',
os.getenv('metrics_table','netops_metrics_parquet')))
```

describa.ipynb

Para visualizar las métricas entrantes, implementamos un paso de canalización que proporciona gráficos y gráficos disponibles a través de las interfaces de usuario de Kubeflow y MLRun. Cada ejecución tiene su propia versión de esta herramienta de visualización.

```
ax.set_title("features correlation")
plt.savefig(os.path.join(base_path, "plots/corr.png"))
context.log_artifact(PlotArtifact("correlation",  body=plt.gcf()),
local_path="plots/corr.html")
```

deploy-feature-function.ipynb

Supervisamos continuamente las métricas en busca de anomalías. Este bloc de notas crea una función sin servidor que genera las funciones que necesitan ejecutar la predicción en las métricas entrantes. Este cuaderno invoca la creación de la función. El código de función se encuentra en el portátil `data-prep.ipynb`. Observe que utilizamos el mismo bloc de notas como un paso en la tubería para este propósito.

training.ipynb

Una vez que creamos las funciones, activamos la formación del modelo. El resultado de este paso es el modelo que se va a utilizar para la inferencia. También recopilamos estadísticas para realizar un seguimiento de cada ejecución (experimento).

Por ejemplo, el siguiente comando introduce la puntuación de precisión en el contexto de ese experimento. Este valor es visible en Kubeflow y MLRun.

```
context.log_result('accuracy',score)
```

despliegue-inferencia-function.ipynb

El último paso de la canalización es poner en marcha el modelo como una función sin servidor para la inferencia continua. Este cuaderno invoca la creación de la función sin servidor definida en `nuclio-inference- function.ipynb`.

Revisar y crear el canalización

La combinación de ejecutar todos los portátiles en una tubería permite que la ejecución continua de experimentos reevalúe la precisión del modelo con las nuevas métricas. En primer lugar, abra la `pipeline.ipynb` portátil. Le repasamos por los detalles que muestran cómo NetApp y Iguazio simplifican la puesta en marcha de esta canalización DE ML.

Utilizamos MLRun para proporcionar contexto y manejar la asignación de recursos a cada paso de la canalización. El servicio de API MLRun se ejecuta en la plataforma Iguazio y es el punto de interacción con los recursos de Kubernetes. Cada desarrollador no puede solicitar recursos directamente; la API gestiona las solicitudes y habilita los controles de acceso.

```
# MLRun API connection definition
mlconf.dbpath = 'http://mlrun-api:8080'
```

La canalización puede funcionar con Cloud Volumes de NetApp y los volúmenes en las instalaciones. Hemos creado esta demostración para usar Cloud Volumes, pero puede ver en el código la opción para ejecutarse en las instalaciones.

```
# Initialize the NetApp snap fucntion once for all functions in a notebook
if [ NETAPP_CLOUD_VOLUME ]:
    snapfn =
code_to_function('snap',project='NetApp',kind='job',filename="snap_cv.ipyn
b").apply(mount_v3io())
    snap_params = {
    "metrics_table" : metrics_table,
    "NETAPP_MOUNT_PATH" : NETAPP_MOUNT_PATH,
    'MANAGER' : MANAGER,
    'svm' : svm,
    'email': email,
    'password': password ,
    'weid': weid,
    'volume': volume,
    "APP_DIR" : APP_DIR
       }
else:
    snapfn =
code_to_function('snap',project='NetApp',kind='job',filename="snapshot.ipy
nb").apply(mount_v3io())
….
snapfn.spec.image = docker_registry + '/netapp/pipeline:latest'
snapfn.spec.volume_mounts =
[snapfn.spec.volume_mounts[0],netapp_volume_mounts]
      snapfn.spec.volumes = [ snapfn.spec.volumes[0],netapp_volumes]
```

La primera acción necesaria para convertir un cuaderno Juppyter en un paso de Kubeflow es convertir el código en una función. Una función tiene todas las especificaciones necesarias para ejecutar ese portátil. A medida que se desplaza hacia abajo por el bloc de notas, puede ver que definimos una función para cada paso de la canalización.

Parte del portátil	Descripción
<code_to_function> (parte del módulo MLRun)	Nombre de la función: Nombre del proyecto. se utiliza para organizar todos los artefactos del proyecto. Esto es visible en la interfaz de usuario de MLRun. Amable. En este caso, un trabajo de Kubernetes. Esto podría ser DASK, mpi, sparkk8s, y más. Consulte la documentación de MLRun para obtener más detalles. Archivo. El nombre del portátil. También puede ser una ubicación en Git (HTTP).
imagen	El nombre de la imagen de Docker que estamos utilizando para este paso. Lo hemos creado anteriormente con el bloc de notas create-image.ipynb.
montajes_volúmenes y volúmenes	Detalles para montar el Cloud Volume de NetApp en tiempo de ejecución.

También definimos parámetros para los pasos.

```
params={   "FEATURES_TABLE":FEATURES_TABLE,
           "SAVE_TO" : SAVE_TO,
           "metrics_table" : metrics_table,
           'FROM_TSDB': 0,
           'PREDICTIONS_TABLE': PREDICTIONS_TABLE,
           'TRAIN_ON_LAST': '1d',
           'TRAIN_SIZE':0.7,
           'NUMBER_OF_SHARDS' : 4,
           'MODEL_FILENAME' : 'netops.v3.model.pickle',
           'APP_DIR' : APP_DIR,
           'FUNCTION_NAME' : 'netops-inference',
           'PROJECT_NAME' : 'netops',
           'NETAPP_SIM' : NETAPP_SIM,
           'NETAPP_MOUNT_PATH': NETAPP_MOUNT_PATH,
           'NETAPP_PVC_CLAIM' : NETAPP_PVC_CLAIM,
           'IGZ_CONTAINER_PATH' : IGZ_CONTAINER_PATH,
           'IGZ_MOUNT_PATH' : IGZ_MOUNT_PATH
            }
```

Después de tener la definición de función para todos los pasos, puede construir la canalización. Utilizamos la `kfp` módulo para realizar esta definición. La diferencia entre el uso de MLRun y la construcción por su cuenta es la simplificación y el acortamiento de la codificación.

Las funciones definidas se convierten en componentes de pasos mediante el `as_step` Función de MLRun.

Definición de paso de instantánea

Inicie una función Snapshot, la salida y el montaje v3io como fuente:

```
snap = snapfn.as_step(NewTask(handler='handler',params=snap_params),
name='NetApp_Cloud_Volume_Snapshot',outputs=['snapVolumeDetails','training
_parquet_file']).apply(mount_v3io())
```

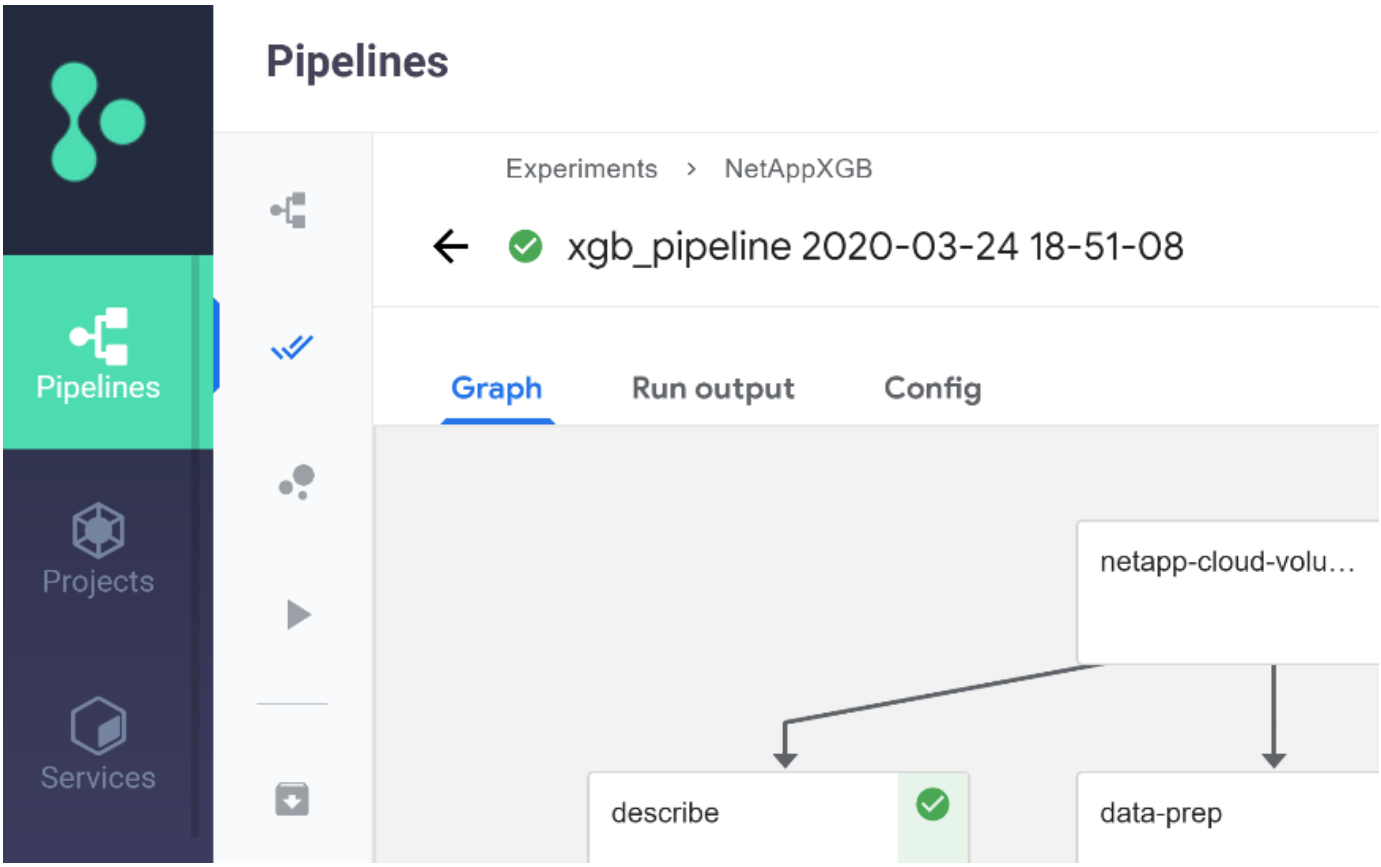
Parámetros	Detalles
Nueva tarea	Newtask es la definición de la ejecución de la función.
(Módulo MLRun)	Manipulador. Nombre de la función Python que se va a invocar. Utilizamos el controlador de nombres en el portátil, pero no es necesario. parámetros. Los parámetros que pasamos a la ejecución. Dentro de nuestro código, utilizamos context.get_param («PARAMETER») para obtener los valores.

Parámetros	Detalles
paso_as	Nombre. Nombre del paso de la canalización de Kubeflow. salidas. Estos son los valores que el paso agrega al diccionario al terminar. Eche un vistazo al portátil SNAP_cv.ipynb. mount_v3io(). Esto configura el paso para montar /User para el usuario que ejecuta la canalización.

```
prep = data_prep.as_step(name='data-prep',
handler='handler',params=params,
                          inputs = {'DATA_DIR':
snap.outputs['snapVolumeDetails']} ,

out_path=artifacts_path).apply(mount_v3io()).after(snap)
```

Parámetros	Detalles
entradas	Puede pasar a un paso las salidas de un paso anterior. En este caso, snap.outsits[napVolumeDetails] es el nombre de la copia Snapshot que creamos en el paso snap.
ruta de salida	Ubicación para colocar artefactos que generan utilizando el módulo MLRun log_Artifacts.

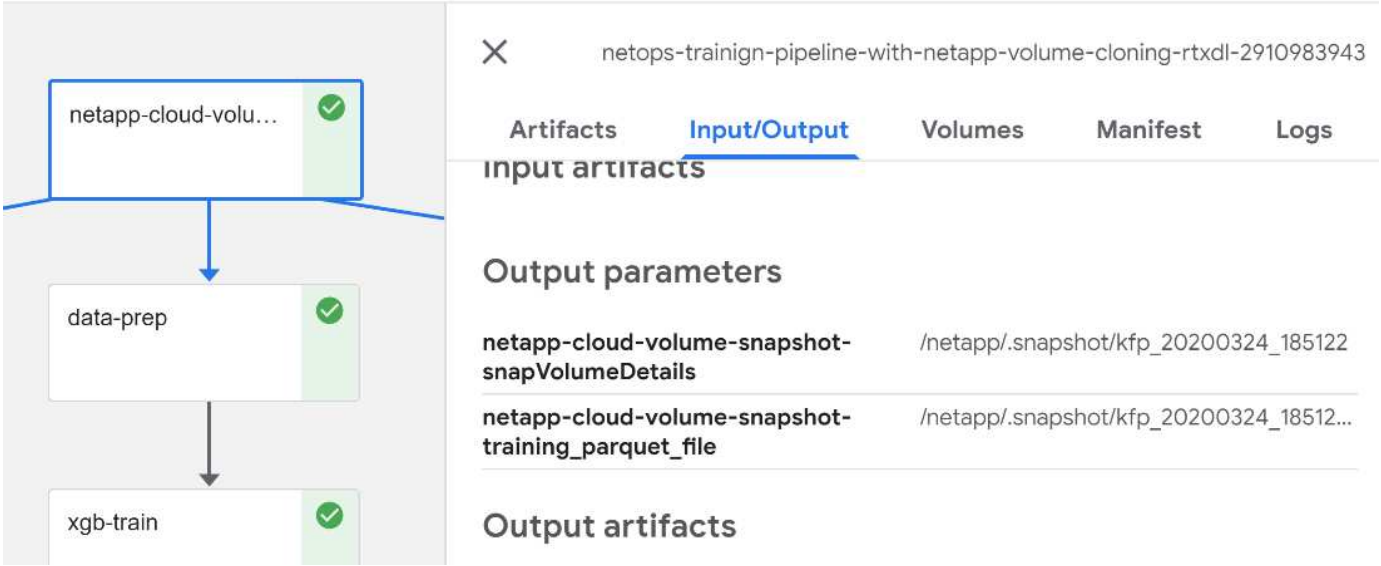
Puede ejecutar `pipeline.ipynb` de arriba a abajo. A continuación, puede ir a la pestaña tuberías desde el panel de control de Iguazio para supervisar el progreso tal y como se ve en la pestaña tuberías del panel de control de Iguazio.



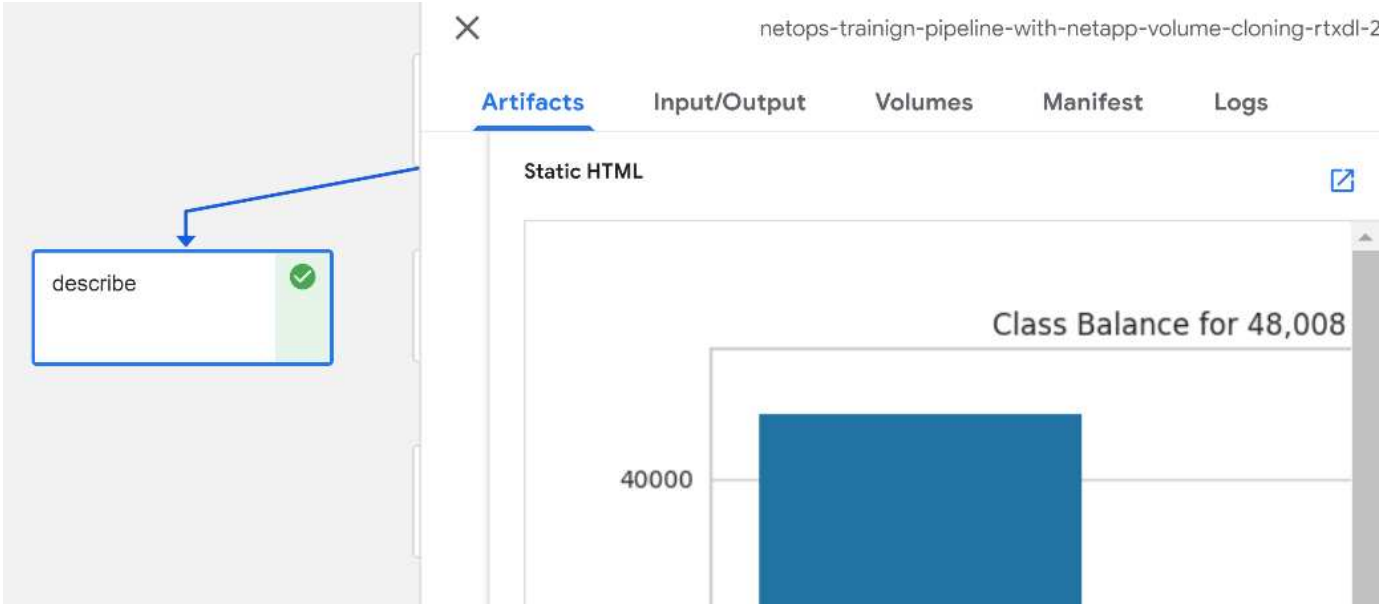
Debido a que hemos registrado la precisión del paso de entrenamiento en cada carrera, tenemos un registro de precisión para cada experimento, como se ve en el registro de precisión de entrenamiento.

Run name	Status	Duration	Pipeline Version	Recurring ...	Start time	accuracy
xgb_pipeline 2020-03-24 18-51-...	✓	0:08:43	[View pipeline]	-	3/24/2020, 2:51:09 PM	0.985
xgb_pipeline 2020-03-19 13-31-...	✓	0:08:14	[View pipeline]	-	3/19/2020, 9:31:19 AM	0.980
xgb_pipeline 2020-03-18 12-56-...	✓	0:08:11	[View pipeline]	-	3/18/2020, 8:56:08 AM	0.990
xgb_pipeline 2020-03-17 19-49-...	✓	0:08:03	[View pipeline]	-	3/17/2020, 3:49:31 PM	0.985
xgb_pipeline 2020-03-17 18-34-...	✓	0:05:54	[View pipeline]	-	3/17/2020, 2:34:56 PM	0.980
xgb_pipeline 2020-03-17 17-34-...	✓	0:04:48	[View pipeline]	-	3/17/2020, 1:34:16 PM	0.982
xgb_pipeline 2020-03-17 17-01-...	✓	0:05:25	[View pipeline]	-	3/17/2020, 1:01:58 PM	0.987
xgb_pipeline 2020-03-16 16-47-...	✓	0:06:08	[View pipeline]	-	3/16/2020, 12:47:19 ...	0.983
xgb_pipeline 2020-03-16 13-57-...	✓	0:05:18	[View pipeline]	-	3/16/2020, 9:57:03 AM	0.980

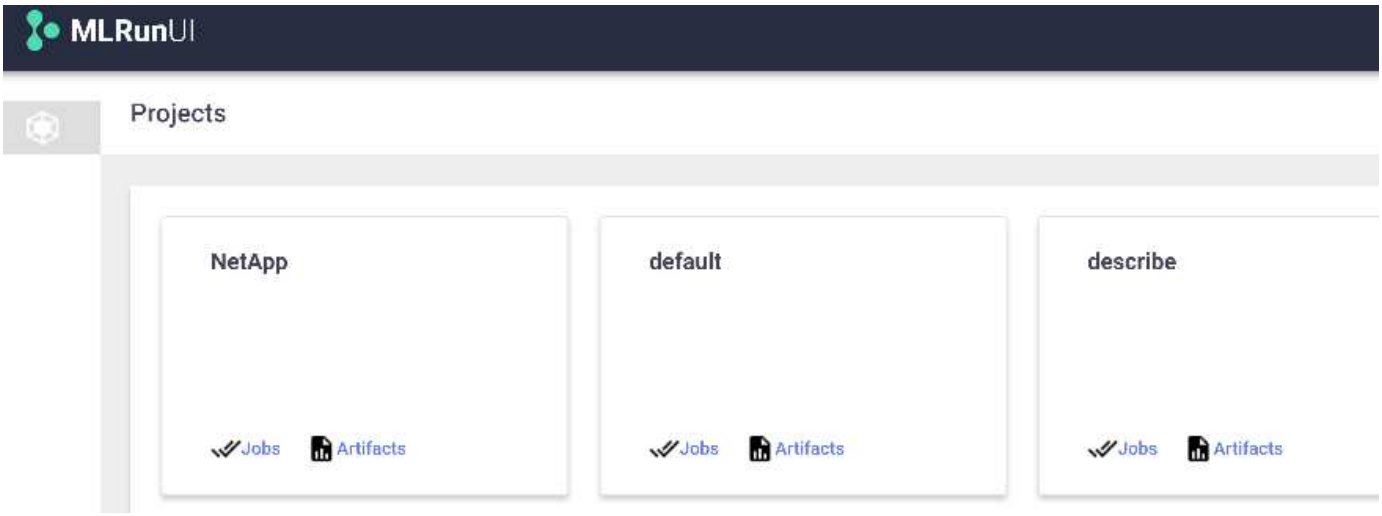
Si selecciona el paso Snapshot, puede ver el nombre de la copia Snapshot que se utilizó para ejecutar este experimento.



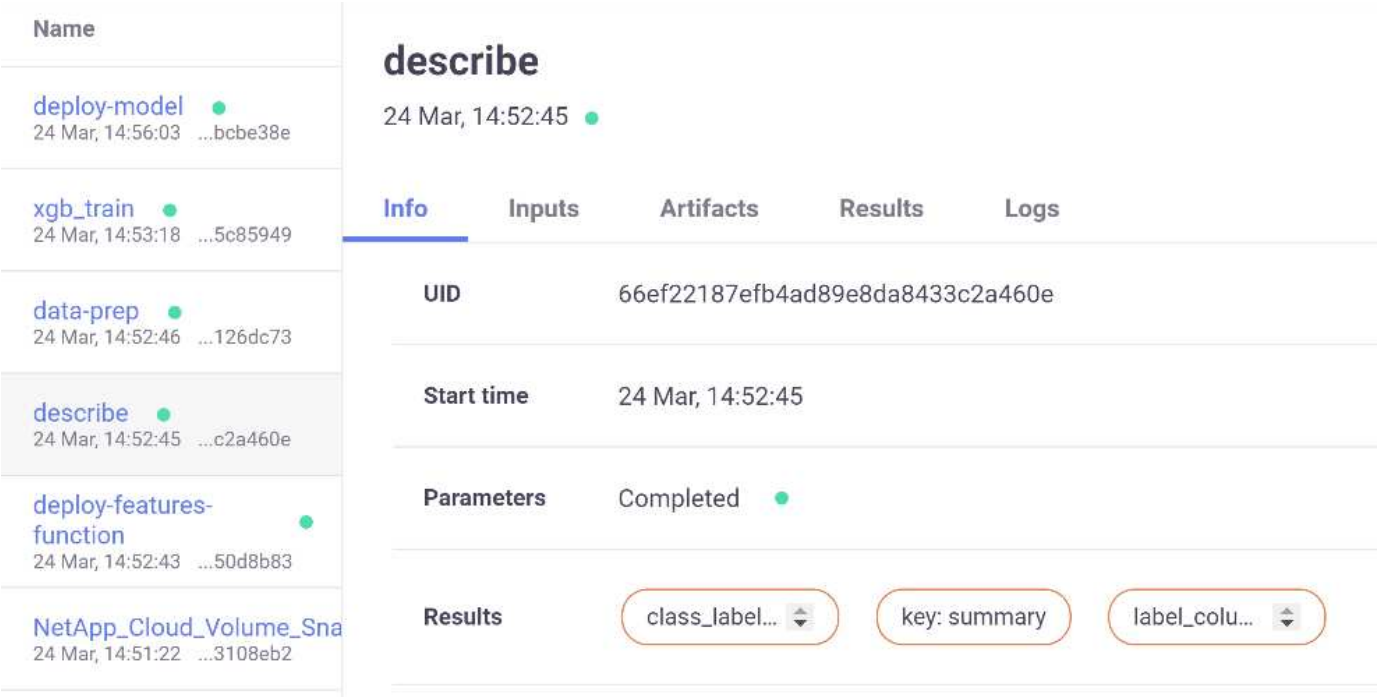
El paso descrito tiene artefactos visuales para explorar las métricas que utilizamos. Puede expandir para ver el trazado completo como se ve en la siguiente imagen.



La base de datos de la API MLRun también realiza un seguimiento de las entradas, salidas y artefactos de cada ejecución organizada por el proyecto. En la siguiente imagen se puede ver un ejemplo de entradas, salidas y artefactos para cada secuencia.



Para cada trabajo, almacenamos detalles adicionales.



Hay más información sobre MLRun que podemos cubrir en este documento. Los artefactos de al, incluida la definición de los pasos y las funciones, se pueden guardar en la base de datos de API, con versiones e invocados individualmente o como un proyecto completo. Los proyectos también se pueden guardar e insertar en Git para su uso posterior. Le animamos a obtener más información en la "Sitio de MLRun GitHub".

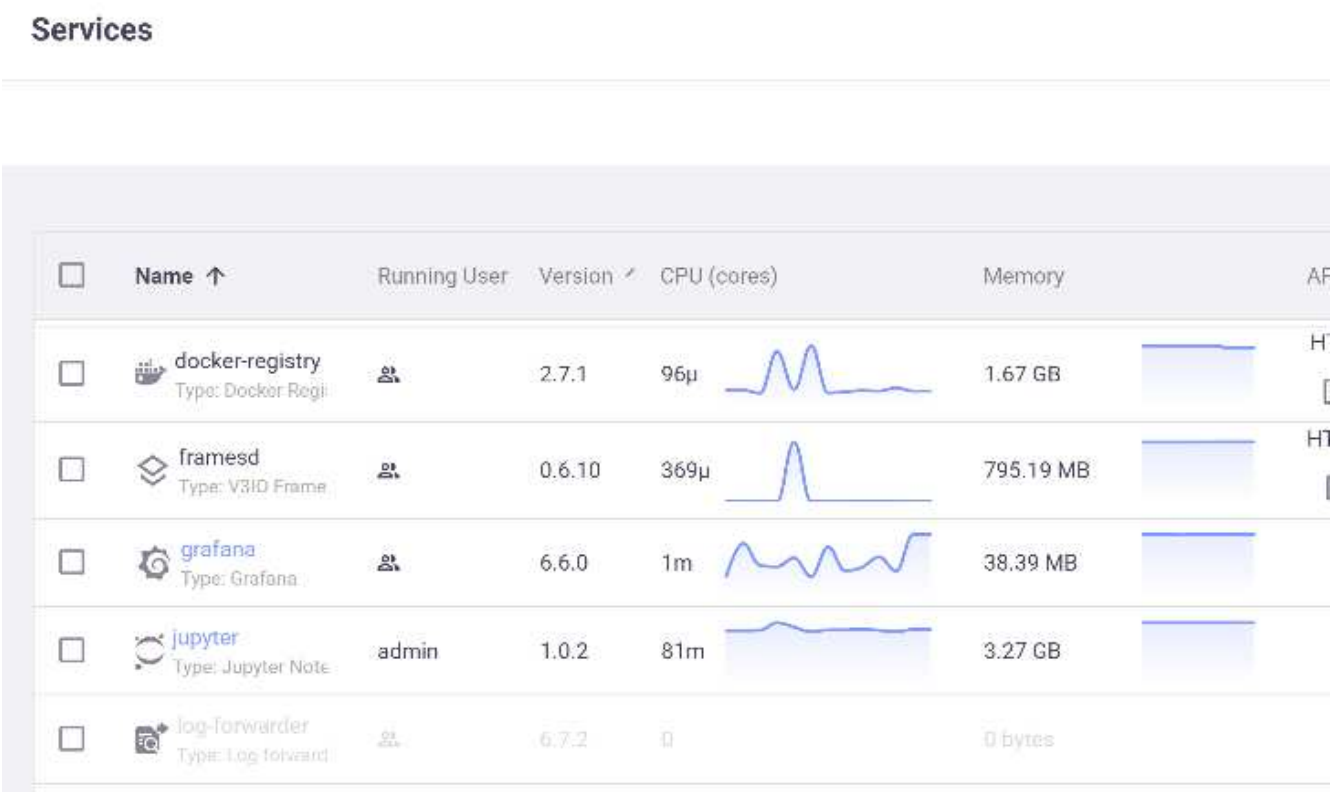
Implemente el panel de Grafana

Una vez que todo se pone en marcha, ejecutamos inferencias sobre nuevos datos. Los modelos predicen fallos en el equipo de dispositivo de red. Los resultados de la predicción se almacenan en una tabla de timbres de Iguazio. Puede visualizar los resultados con Grafana en la plataforma integrada con la política de acceso a datos y seguridad de Iguazio.

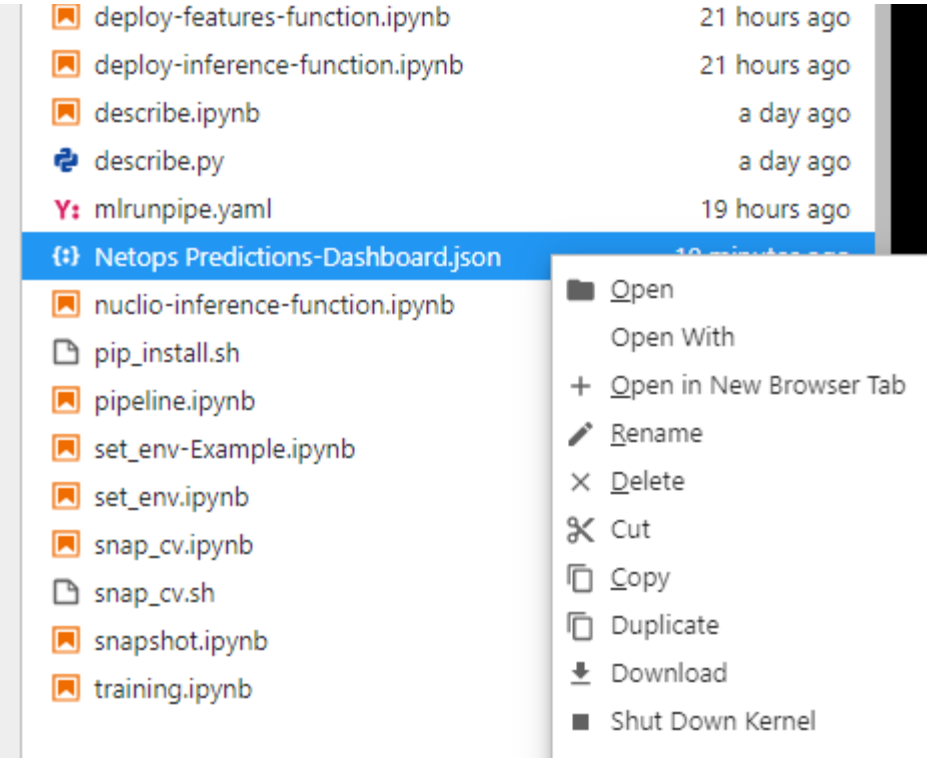
Puede implementar la consola importando el archivo JSON proporcionado en las interfaces de Grafana en el

clúster.

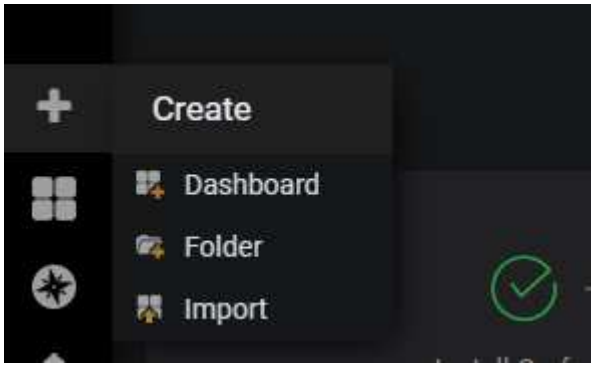
1. Para verificar que el servicio Grafana se está ejecutando, busque Servicios.



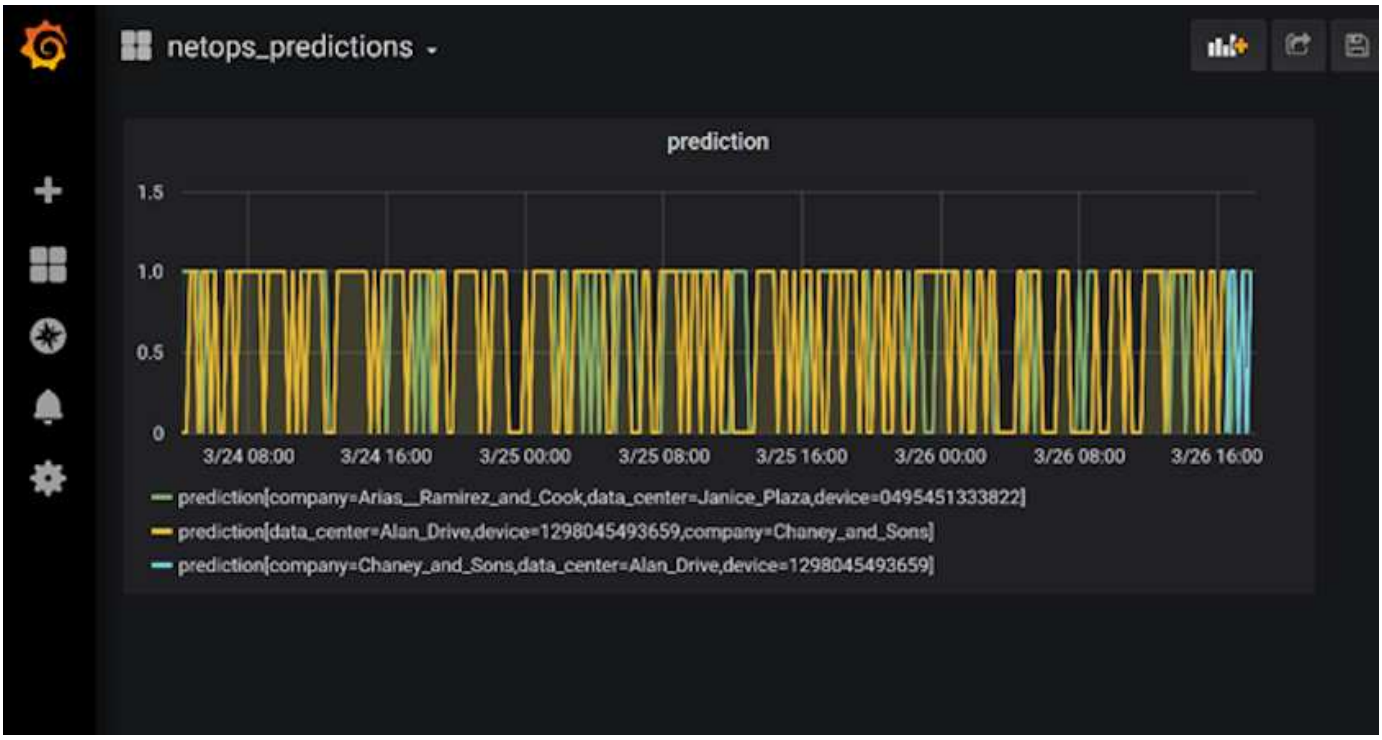
2. Si no está presente, implemente una instancia en la sección Servicios:
   a. Haga clic en Nuevo servicio.
   b. Seleccione Grafana de la lista.
   c. Acepte los valores predeterminados.
   d. Haga clic en Siguiente paso.
   e. Introduzca su ID de usuario.
   f. Haga clic en Guardar servicio.
   g. Haga clic en aplicar cambios en la parte superior.
3. Para implementar el panel de control, descargue el archivo `NetopsPredictions-Dashboard.json` A través de la interfaz Juppyter.



4. Abra Grafana en la sección Servicios e importe el panel de control.



5. Haga clic en Upload `*.json` File (Archivo) y seleccione el archivo que descargó anteriormente (`NetopsPredictions-Dashboard.json`). El panel se muestra una vez finalizada la carga.



Despliegue la función de limpieza

Cuando genera una gran cantidad de datos, es importante mantener las cosas limpias y organizadas. Para ello, implemente la función de limpieza con `cleanup.ipynb` portátil.

Beneficios

NetApp y Iguazio aceleran y simplifican la puesta en marcha de aplicaciones de IA y ML mediante la creación de marcos esenciales como Kubeflow, Apache Spark y TensorFlow, junto con herramientas de orquestación como Docker y Kubernetes. Al unificar la canalización de datos completa, NetApp y Iguazio reducen la latencia y la complejidad inherentes a muchas cargas de trabajo informáticas avanzadas, y esta brecha entre el desarrollo y las operaciones. Los científicos de datos pueden ejecutar consultas en grandes conjuntos de datos y compartir de forma segura datos y modelos algorítmicos con usuarios autorizados durante la fase de entrenamiento. Después de que los modelos en contenedores están listos para la producción, puede moverlos fácilmente desde entornos de desarrollo a entornos operativos.

Conclusión

Cuando cree sus propias canalizaciones de IA/ML, configurar la integración, la gestión, la seguridad y la accesibilidad de los componentes en una arquitectura es una tarea ardua. Dar a los desarrolladores acceso y control de su entorno presenta otro conjunto de retos.

La combinación de NetApp e Iguazio aúna estas tecnologías como servicios gestionados para acelerar la adopción de tecnologías y mejorar los plazos de comercialización de las nuevas aplicaciones DE IA/ML.

TR-4915: Movimiento de datos con E-Series y BeeGFS para flujos de trabajo de análisis e IA

Cody Harryman y Ryan Rodine, NetApp

En TR-4915 se describe cómo mover datos desde cualquier repositorio de datos a un sistema de archivos BeeGFS respaldado por el almacenamiento SAN E-Series de NetApp. En el caso de aplicaciones de inteligencia artificial (IA) y aprendizaje automático (ML), es posible que los clientes deban mover de manera rutinaria grandes conjuntos de datos que superen muchos petabytes de datos en sus clústeres BeeGFS para el desarrollo de modelos. En este documento se explora cómo llevar a cabo este proceso mediante el XCP de NetApp y las herramientas de copia y sincronización de BlueXP de NetApp.

"TR-4915: Movimiento de datos con E-Series y BeeGFS para flujos de trabajo de análisis e IA"

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 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.