Pipeline di dati, data Lake e gestione

NetApp Solutions

NetApp
April 26, 2024

Sommario

Pipeline di dati, data Lake e gestione

AWS FSX per NetApp ONTAP (FSxN) per MLOps

Autore(i):
Jian Jian (Ken), Senior Data & Applied Scientist, NetApp

In questa sezione viene illustrata l'applicazione pratica dello sviluppo di infrastrutture ai, fornendo una panoramica end-to-end della costruzione di una pipeline MLOps utilizzando FSxN. Con tre esempi completi, ti guida a soddisfare le tue esigenze MLOps tramite questa potente piattaforma per la gestione dei dati.

Questi articoli si concentrano su:

1. "Parte 1 - integrazione di AWS FSX per NetApp ONTAP (FSxN) come bucket S3 privato in AWS SageMaker"
2. "Parte 2 - utilizzo di AWS FSX for NetApp ONTAP (FSxN) come origine dati per il training sui modelli in SageMaker"
3. "Parte 3 - creazione di Una pipeline MLOps semplificata (ci/CT/CD)"

Al termine di questa sezione, avrete acquisito una solida comprensione di come utilizzare FSxN per ottimizzare i processi MLOps.

Parte 1 - integrazione di AWS FSX per NetApp ONTAP (FSxN) come bucket S3 privato in AWS SageMaker

Autore(i):
Jian Jian (Ken), Senior Data & Applied Scientist, NetApp

Introduzione

Utilizzando SageMaker come esempio, questa pagina fornisce istruzioni sulla configurazione di FSxN come bucket S3 privato.

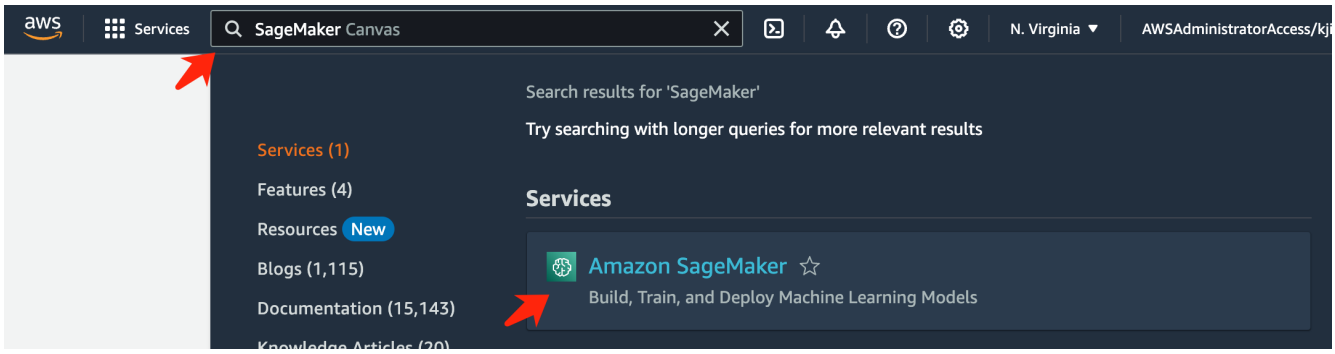
Per ulteriori informazioni su FSxN, si prega di dare un'occhiata a questa presentazione (."Collegamento video")

Guida dell'utente

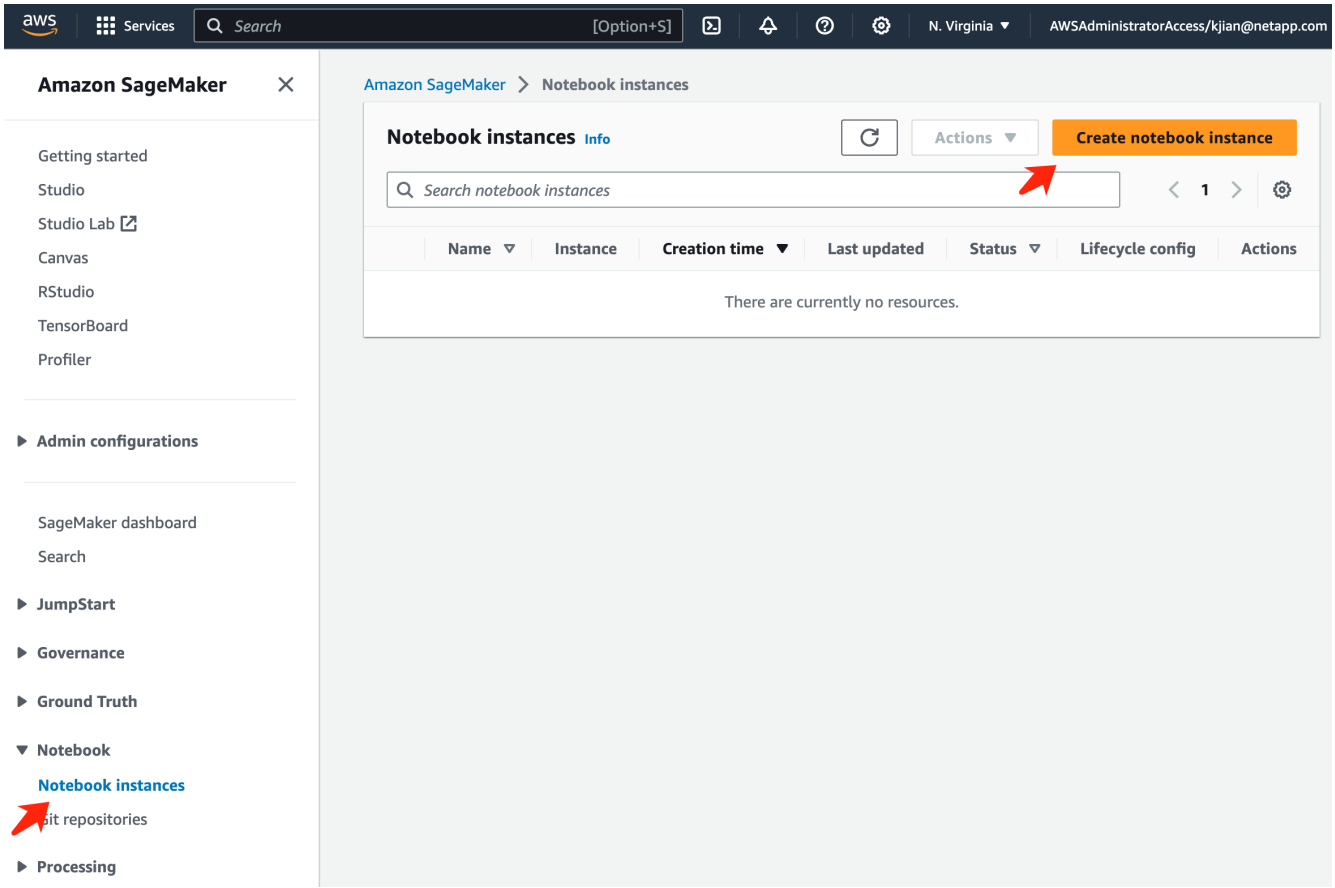
Creazione server

Creare un'istanza di notebook SageMaker

1. Apri la console AWS. Nel pannello di ricerca, cerca SageMaker e fai clic sul servizio **Amazon SageMaker**.



2. Aprire **istanze notebook** nella scheda notebook, fare clic sul pulsante arancione **Crea istanza notebook**.



3. Nella pagina di creazione,
   Immettere il **nome istanza notebook**
   Espandere il pannello **rete**
   Lasciare le altre voci predefinite e selezionare i gruppi **VPC**, **Subnet** e **protezione**. (Questa **VPC** e **sottorete** verranno utilizzate per creare il file system FSxN in un secondo momento)
   Fare clic sul pulsante arancione **Crea istanza notebook** in basso a destra.

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

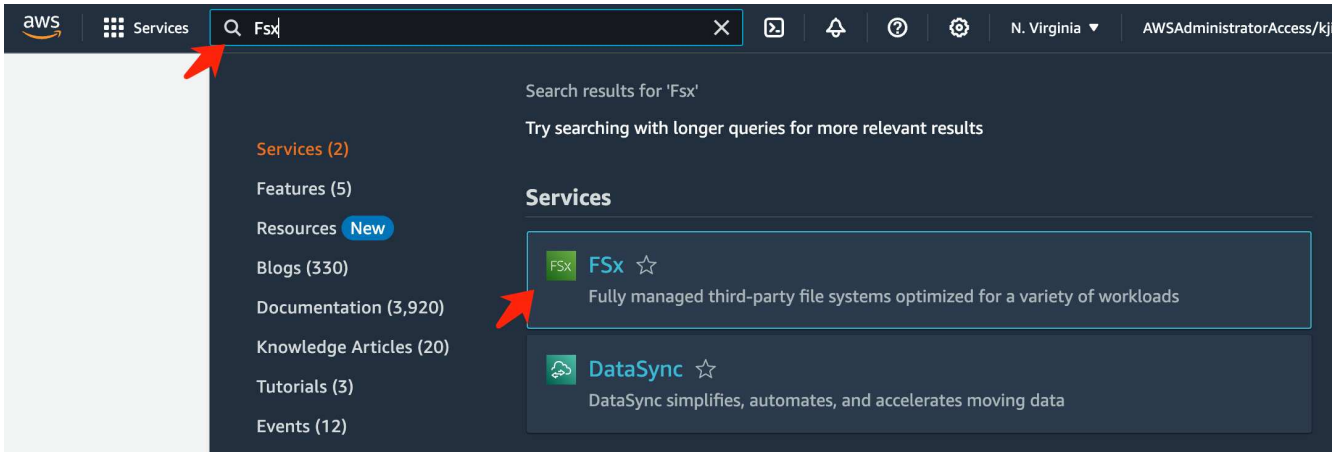
▶ Git repositories- *optional*

▶ Tags - *optional*

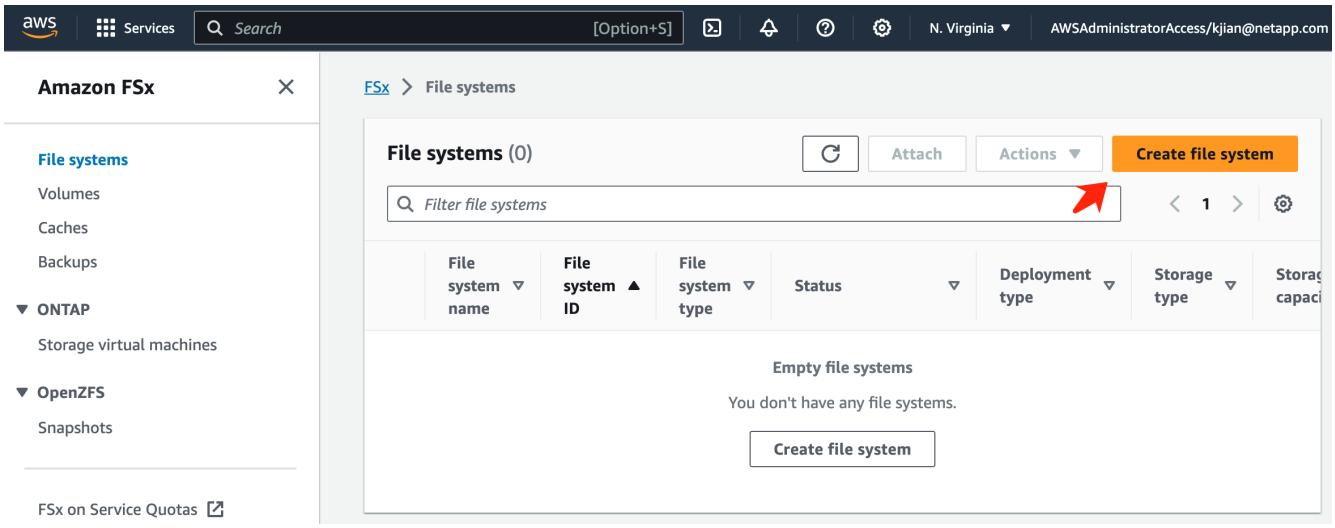
Cancel | Create notebook instance

Creare un file system FSxN

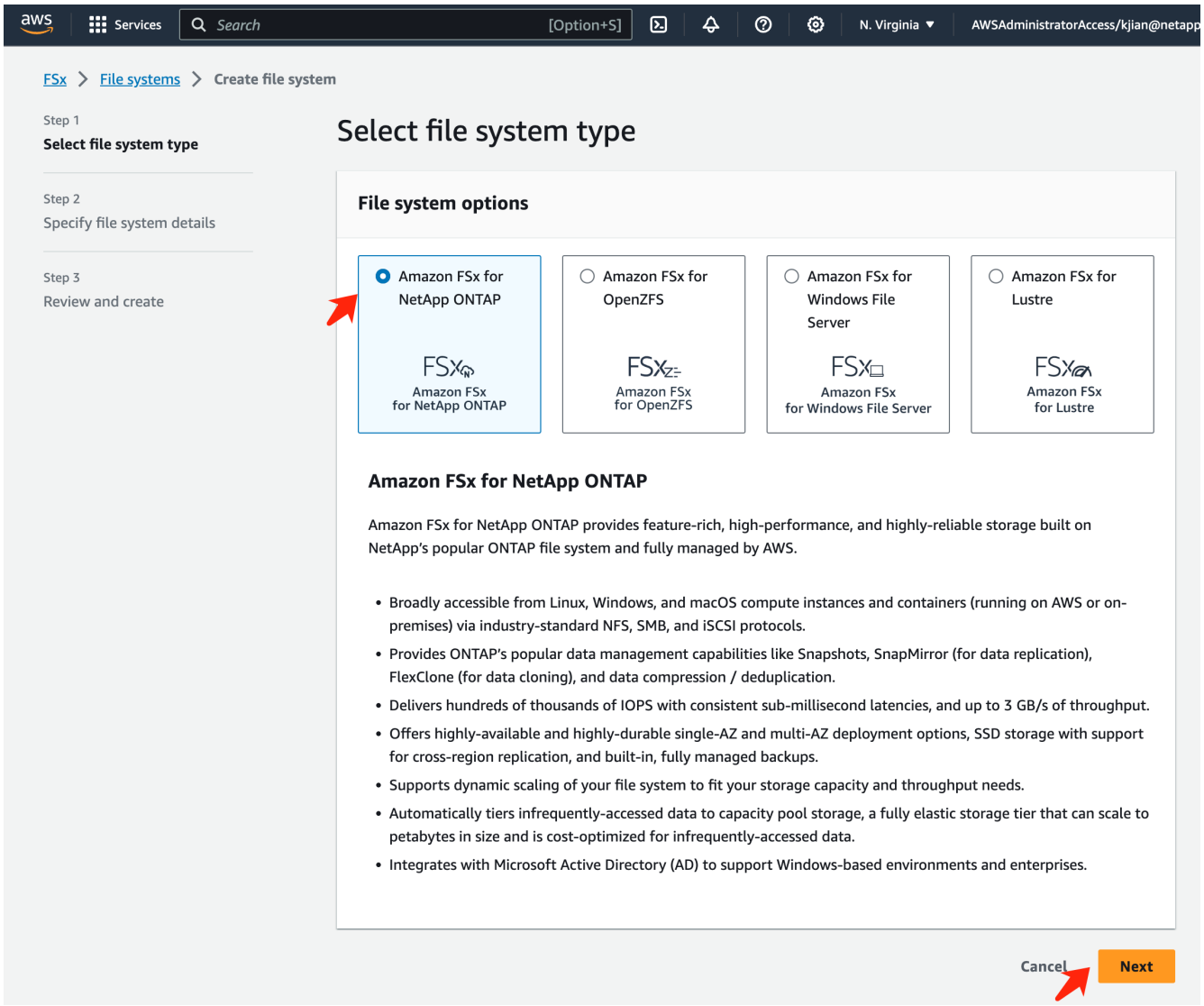
1. Apri la console AWS. Nel pannello di ricerca, cercate FSX e fate clic sul servizio **FSX**.



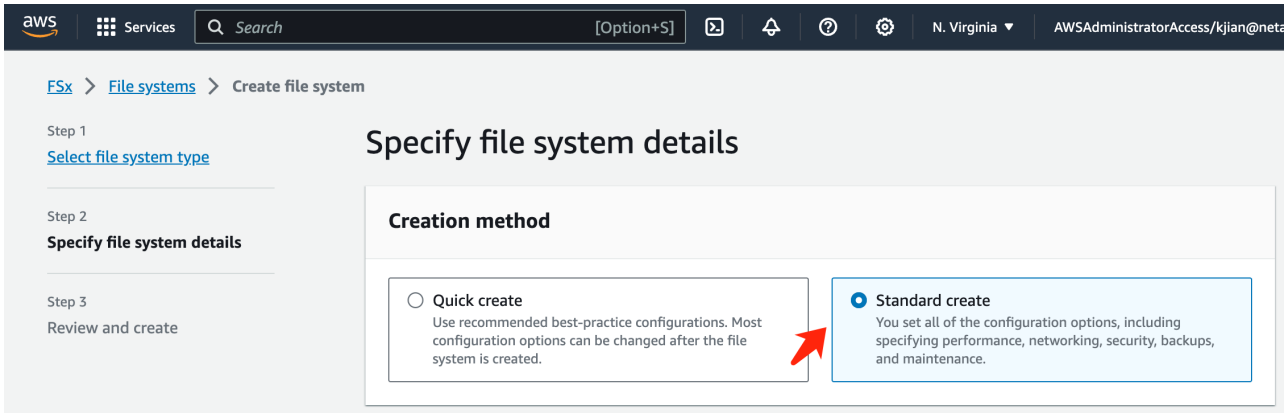
2. Fare clic su **Crea file system**.



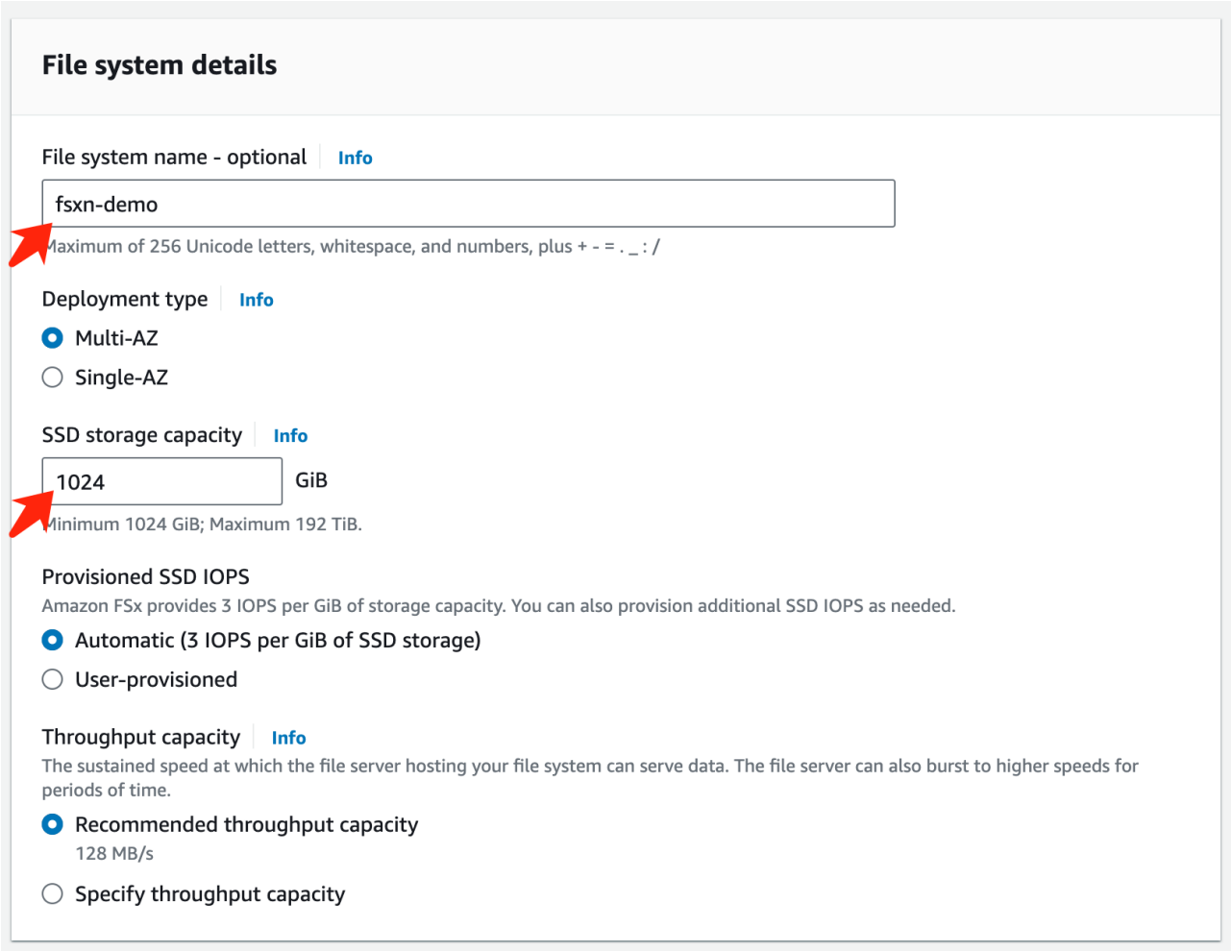
3. Selezionare la prima scheda **FSX per NetApp ONTAP** e fare clic su **Avanti**.



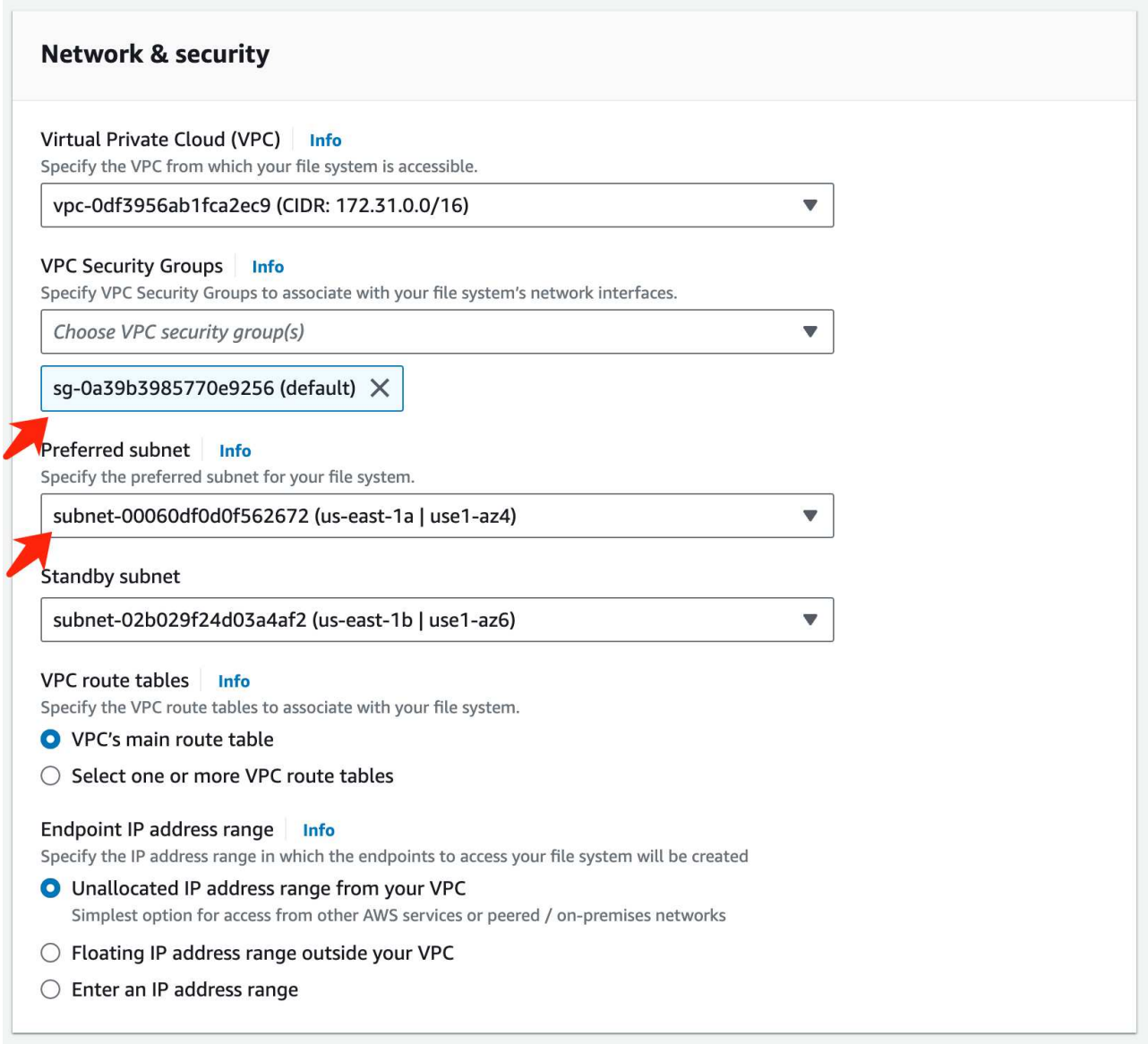
4. Nella pagina di configurazione dei dettagli.
   a. Selezionare l'opzione **creazione standard**.



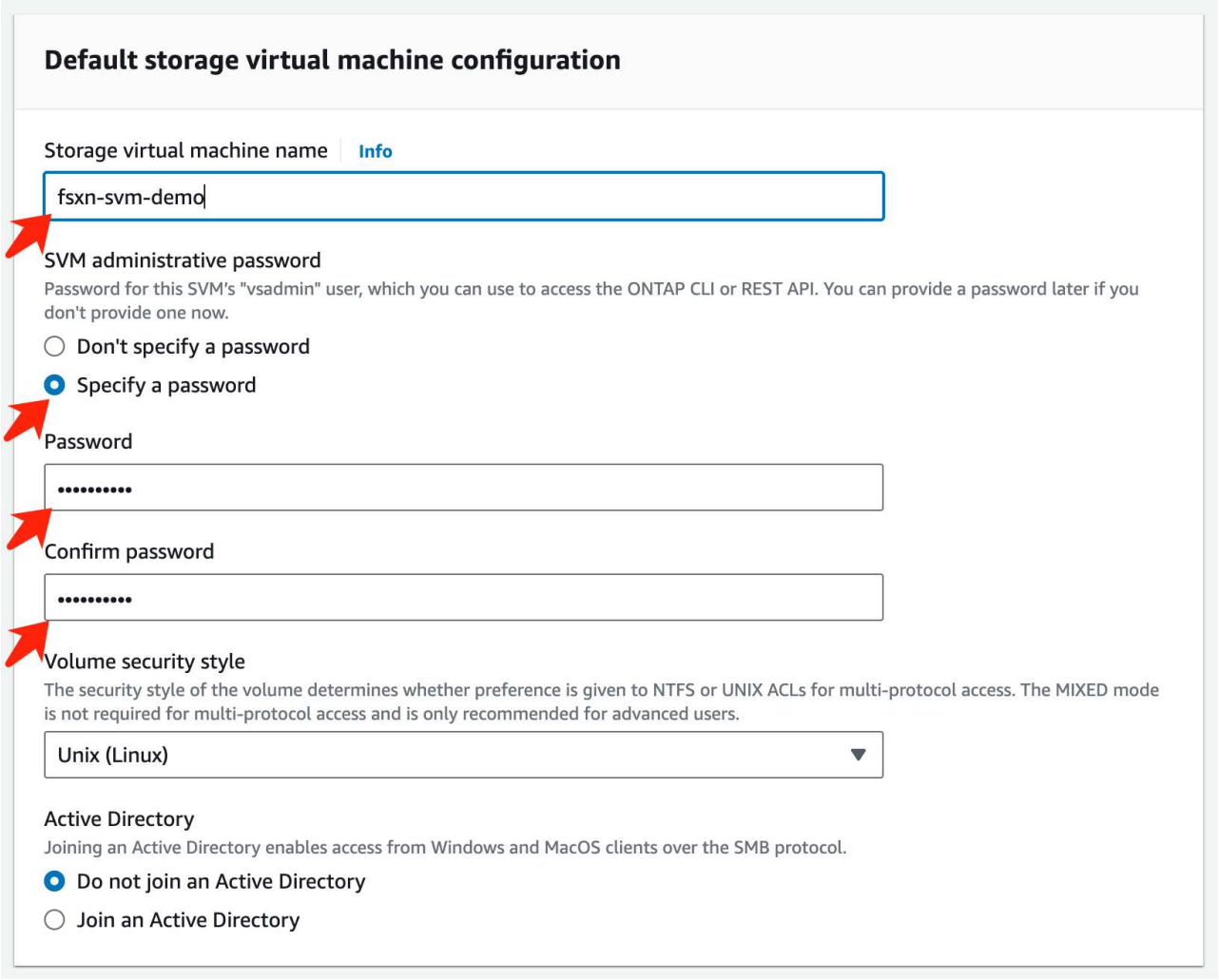
   b. Immettere il **nome del file system** e la **capacità di archiviazione SSD**.



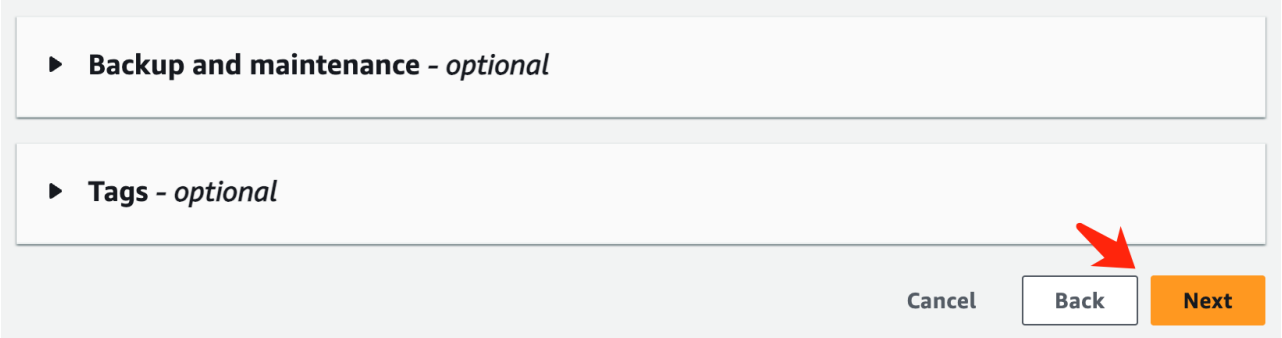
c. Assicurarsi di utilizzare **VPC** e **subnet** uguali all'istanza **SageMaker notebook**.



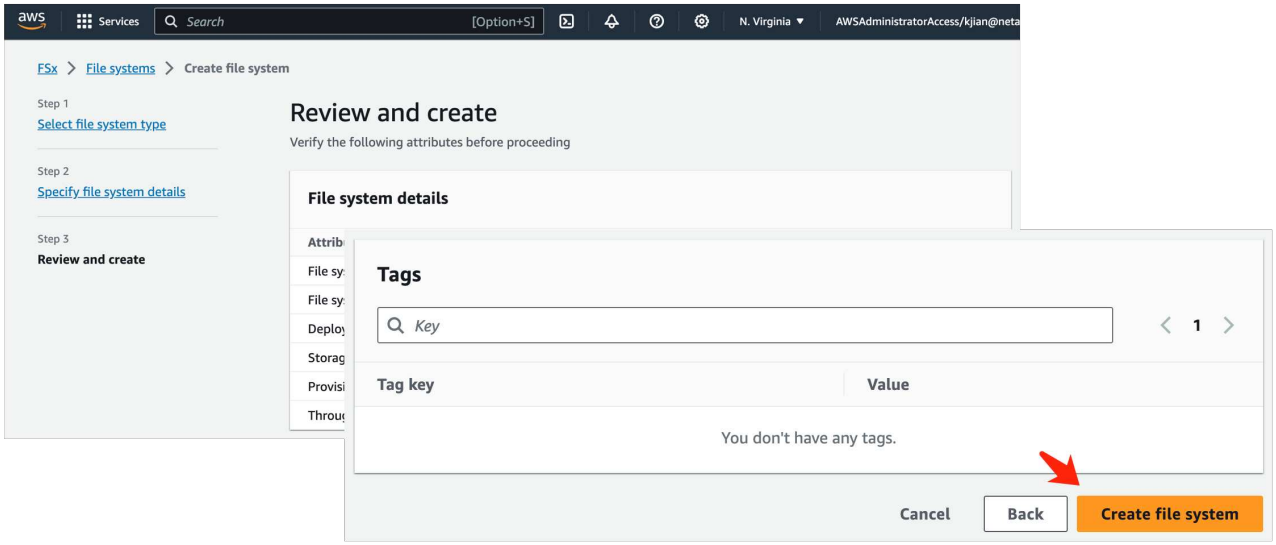
d. Immettere il nome **Storage Virtual Machine** e **specificare una password** per la SVM (Storage Virtual Machine).



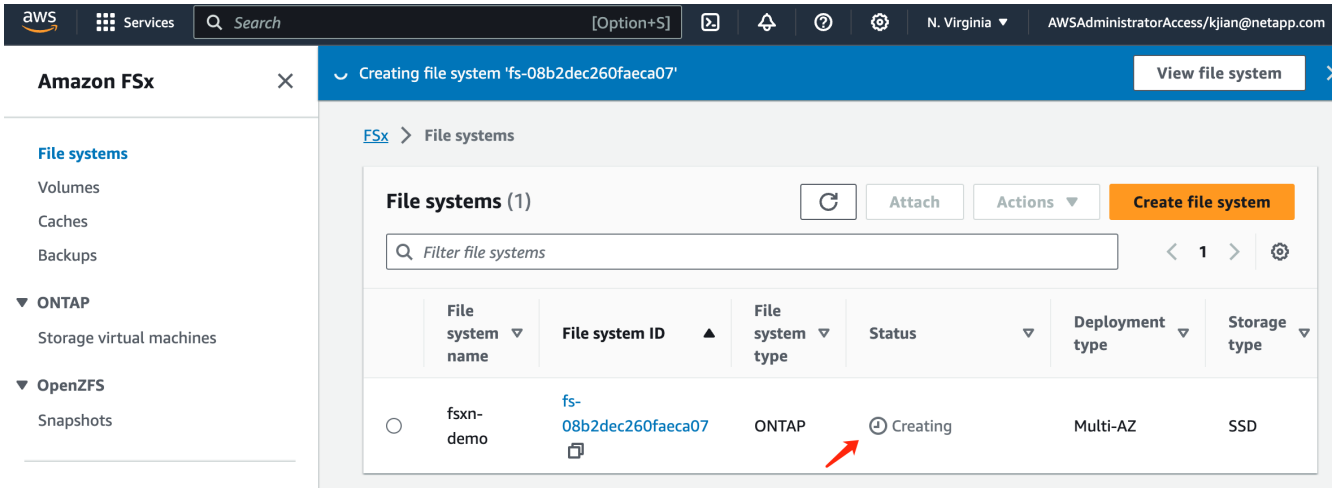
e. Lasciare le altre voci predefinite e fare clic sul pulsante arancione **Avanti** in basso a destra.



f. Fare clic sul pulsante arancione **Crea file system** in basso a destra nella pagina di revisione.



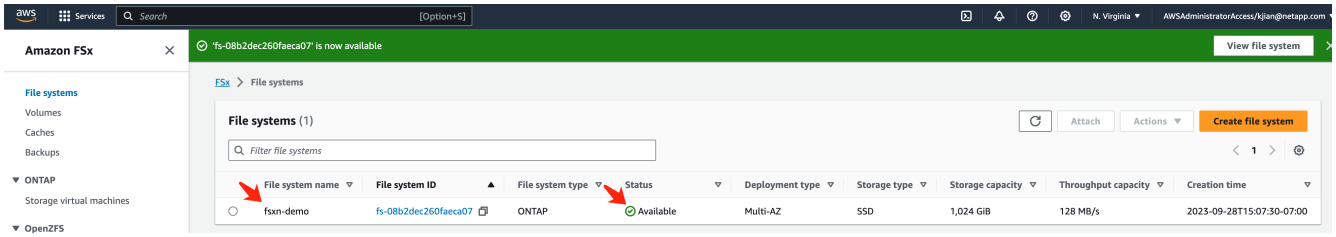
5. L'accelerazione del file system FSX può richiedere circa **20-40 minuti**.



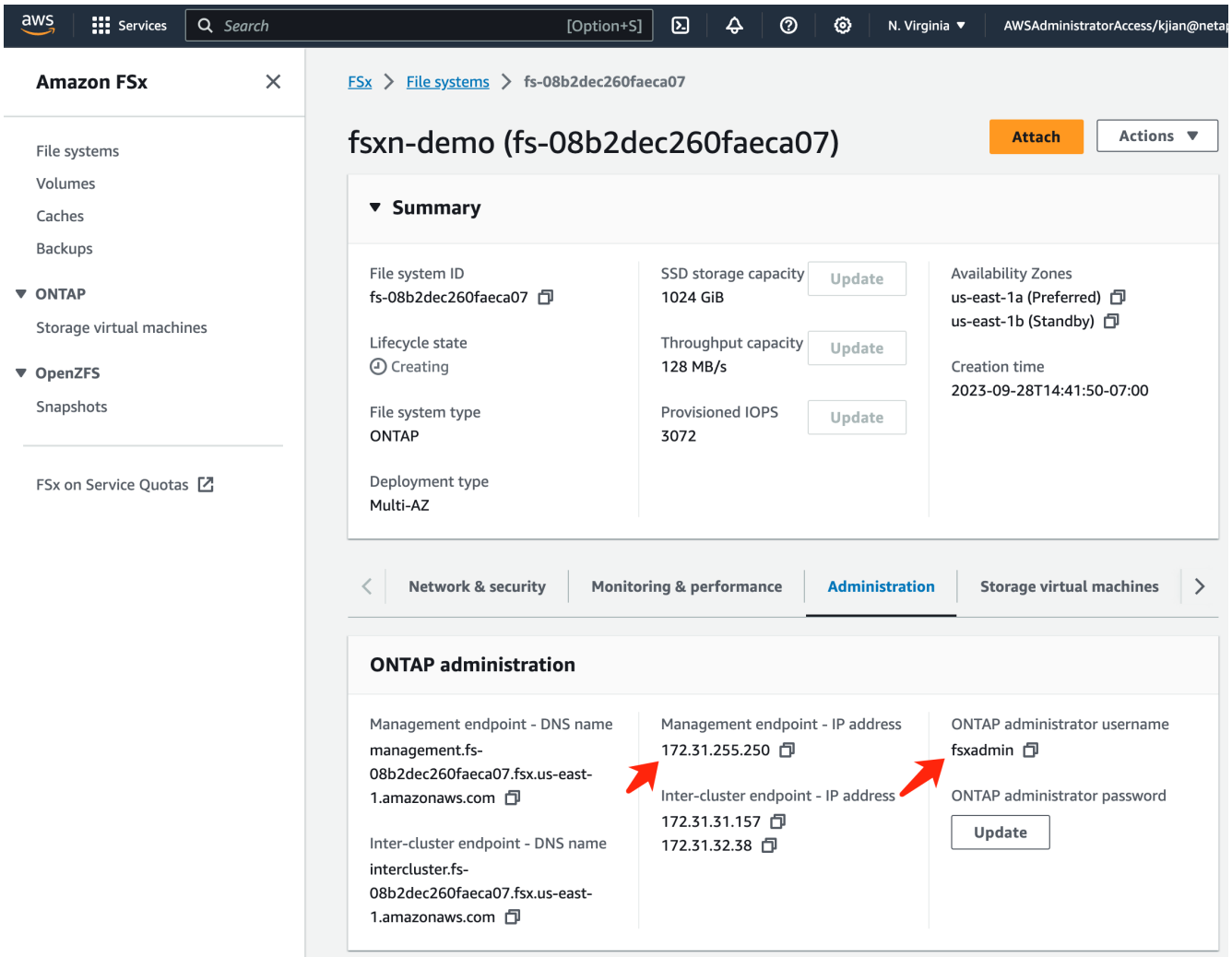
Server Configuration (Configurazione server)

Configurazione ONTAP

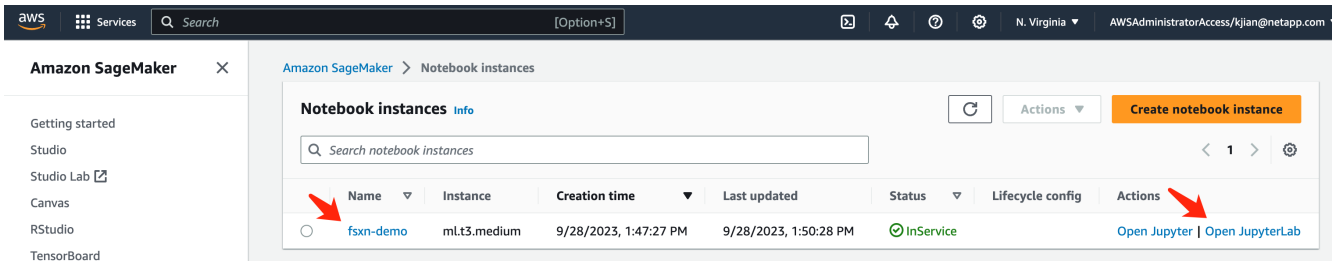
1. Aprire il file system FSX creato. Assicurarsi che lo stato sia **disponibile**.



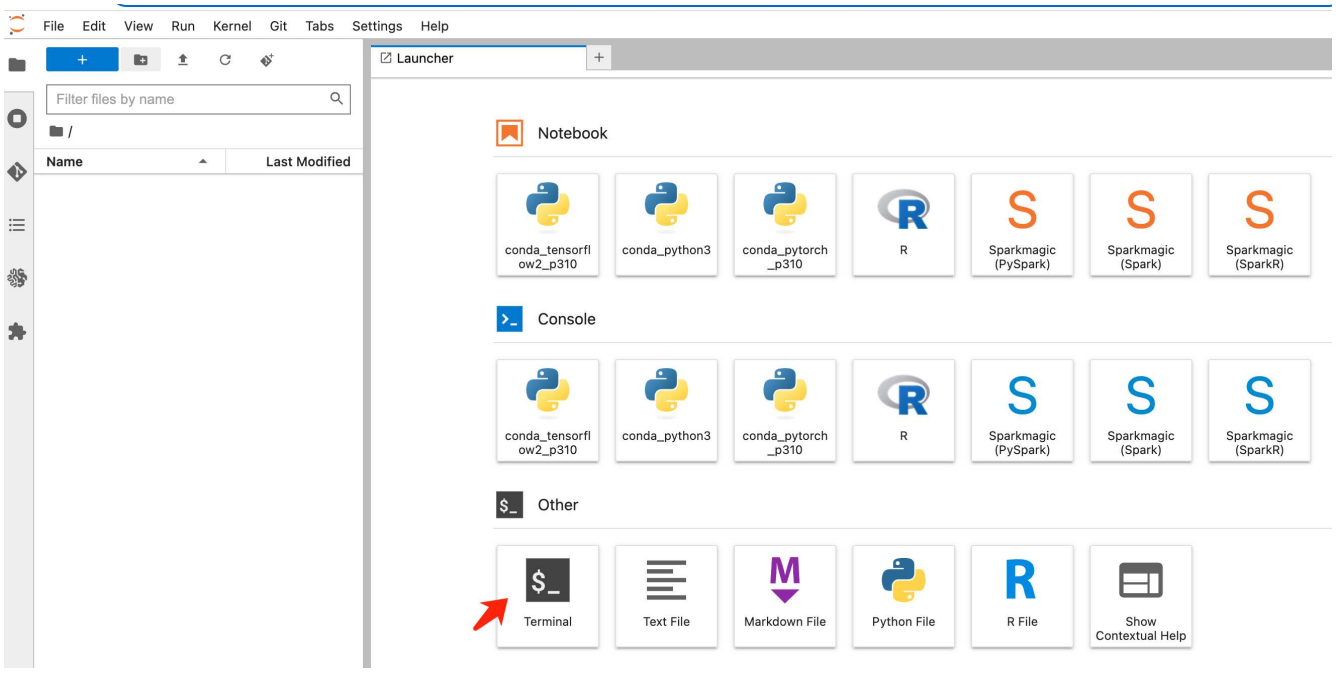
2. Selezionare la scheda **Amministrazione** e mantenere **endpoint di gestione - indirizzo IP** e **nome utente amministratore ONTAP**.



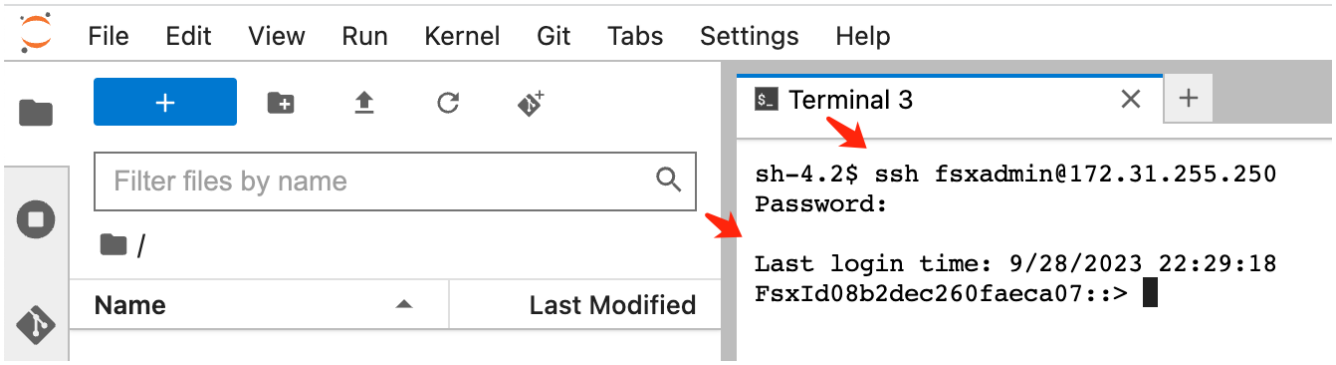
3. Aprire l'istanza creata **SageMaker notebook** e fare clic su **Apri JupyterLab**.



4. Nella pagina Jupyter Lab, aprire un nuovo **terminale**.

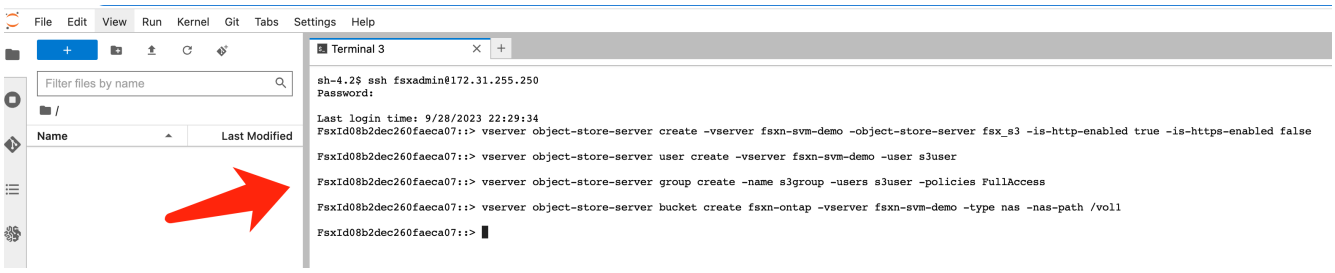


5. Inserisci il comando ssh ssh <nome utente admin>@<IP server ONTAP> per accedere al file system FSxN ONTAP. (Il nome utente e l'indirizzo IP sono recuperati dalla fase 2)
   Utilizzare la password utilizzata durante la creazione della **Storage Virtual Machine**.



6. Eseguire i comandi nel seguente ordine.
   Utilizziamo **fsxn-ontap** come nome per il **nome bucket S3 privato FSxN**.
   Utilizzare **storage virtual machine name** per l'argomento **-vserver**.

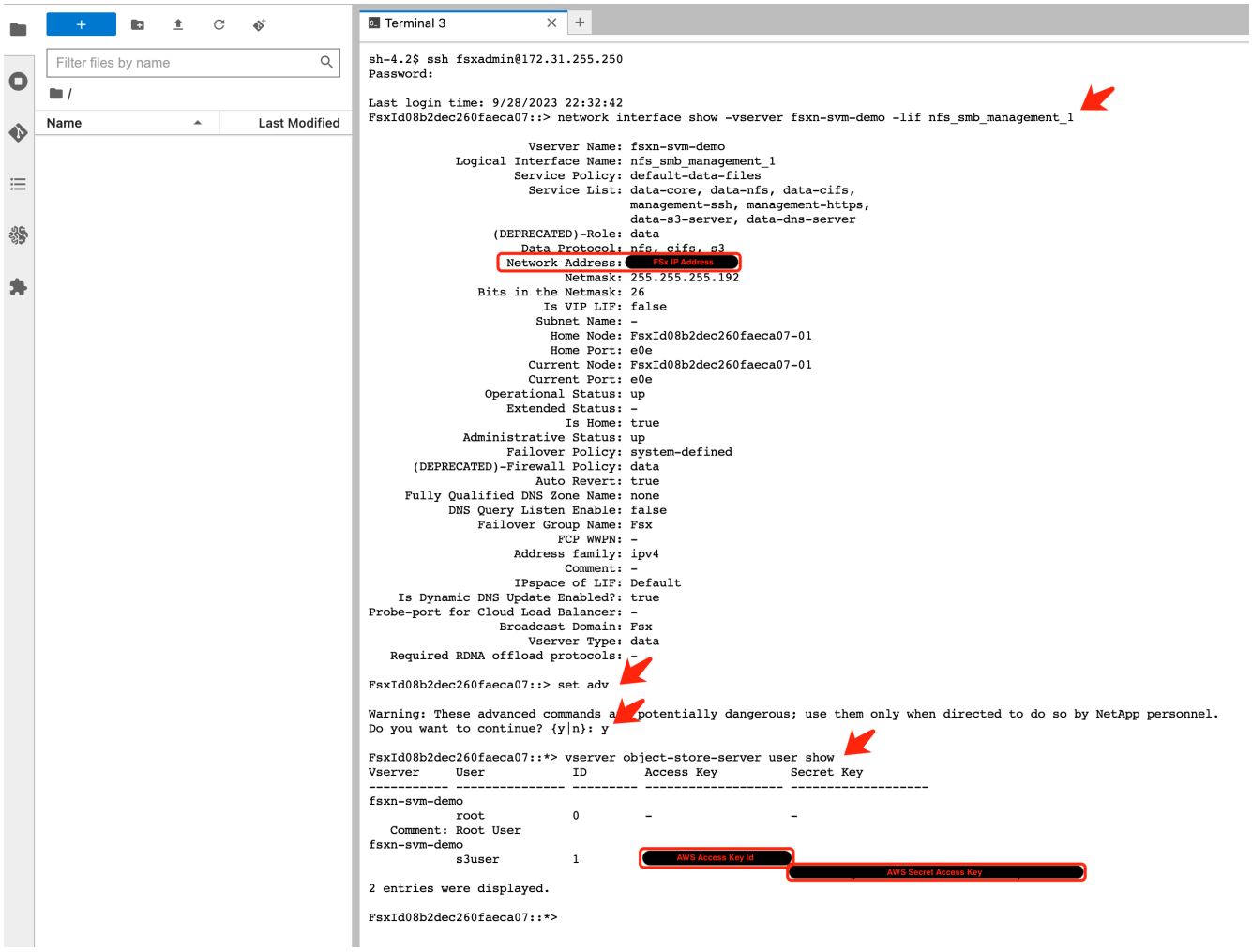
```
vserver object-store-server create -vserver fsxn-svm-demo -object-store
-server fsx_s3 -is-http-enabled true -is-https-enabled false

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. Eseguire i seguenti comandi per recuperare l'IP dell'endpoint e le credenziali per FSxN private S3.

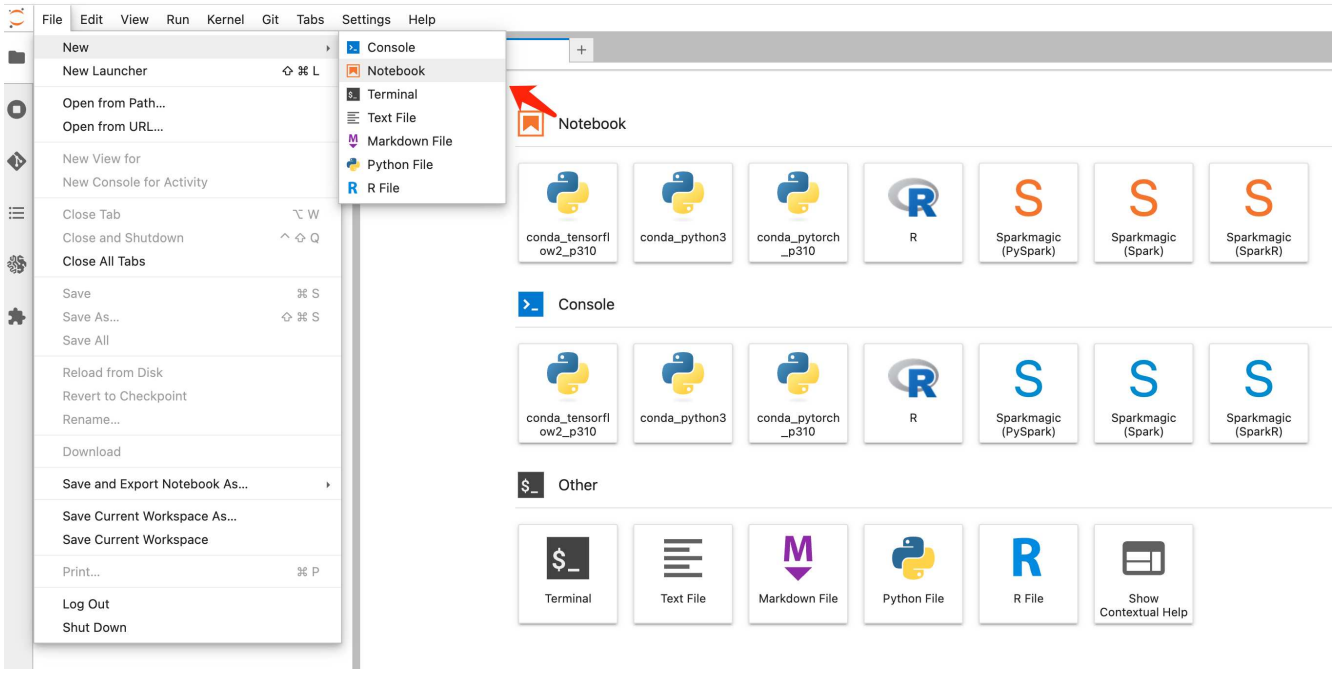
```
network interface show -vserver fsxn-svm-demo -lif nfs_smb_management_1

set adv

vserver object-store-server user show
```

8. Conservare l'IP dell'endpoint e le credenziali per un utilizzo futuro.



Client Configuration (Configurazione client)

1. Nell'istanza di notebook SageMaker, creare un nuovo notebook Jupyter.



2. Utilizzare il codice riportato di seguito come soluzione alternativa per caricare i file nel bucket S3 privato di FSxN.
   Per un esempio di codice completo, fare riferimento a questo notebook.
   "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']
```

Si conclude così l'integrazione tra FSxN e l'istanza SageMaker.

Utile elenco di controllo per il debug

- Verificare che l'istanza di SageMaker notebook e il file system FSxN si trovino nello stesso VPC.
- Ricordarsi di eseguire il comando **set dev** su ONTAP per impostare il livello di privilegio su **dev**.

FAQ (al 27 settembre 2023)

D: Perché viene visualizzato l'errore "**si è verificato un errore (NotImplemented) quando si chiama l'operazione CreateMultipartUpload: Il comando S3 richiesto non è implementato**" quando si caricano i file su FSxN?

R: Come bucket S3 privato, FSxN supporta il caricamento di file fino a 100MB MB. Quando si utilizza il protocollo S3, i file di dimensioni superiori a 100MB KB vengono divisi in 100MB blocchi e viene richiamata la funzione "CreateMultipartUpload". Tuttavia, l'attuale implementazione di FSxN private S3 non supporta questa funzione.

D: Perché ricevo l'errore "**si è verificato un errore (AccessDenied) quando si chiamano le operazioni PutObject: Access Denied**" quando si caricano i file su FSxN?

R: Per accedere al bucket S3 privato FSxN da un'istanza di SageMaker notebook, passare le credenziali AWS alle credenziali FSxN. Tuttavia, la concessione del permesso di scrittura all'istanza richiede una soluzione alternativa che implica il montaggio del bucket e l'esecuzione del comando shell 'chmod' per modificare le autorizzazioni.

D: Come posso integrare il bucket S3 privato di FSxN con altri servizi ML di SageMaker?

R: Purtroppo, SageMaker Services SDK non fornisce un modo per specificare l'endpoint per il bucket S3 privato. Di conseguenza, FSxN S3 non è compatibile con i servizi SageMaker come Sagemaker Data Wrangler, Sagemaker Clarify, Sagemaker Glue, Sagemaker Athena, Sagemaker AutoML, e altri.

Parte 2 - utilizzo di AWS FSX for NetApp ONTAP (FSxN) come origine dati per il training sui modelli in SageMaker

Autore(i):

Jian Jian (Ken), Senior Data & Applied Scientist, NetApp

Introduzione

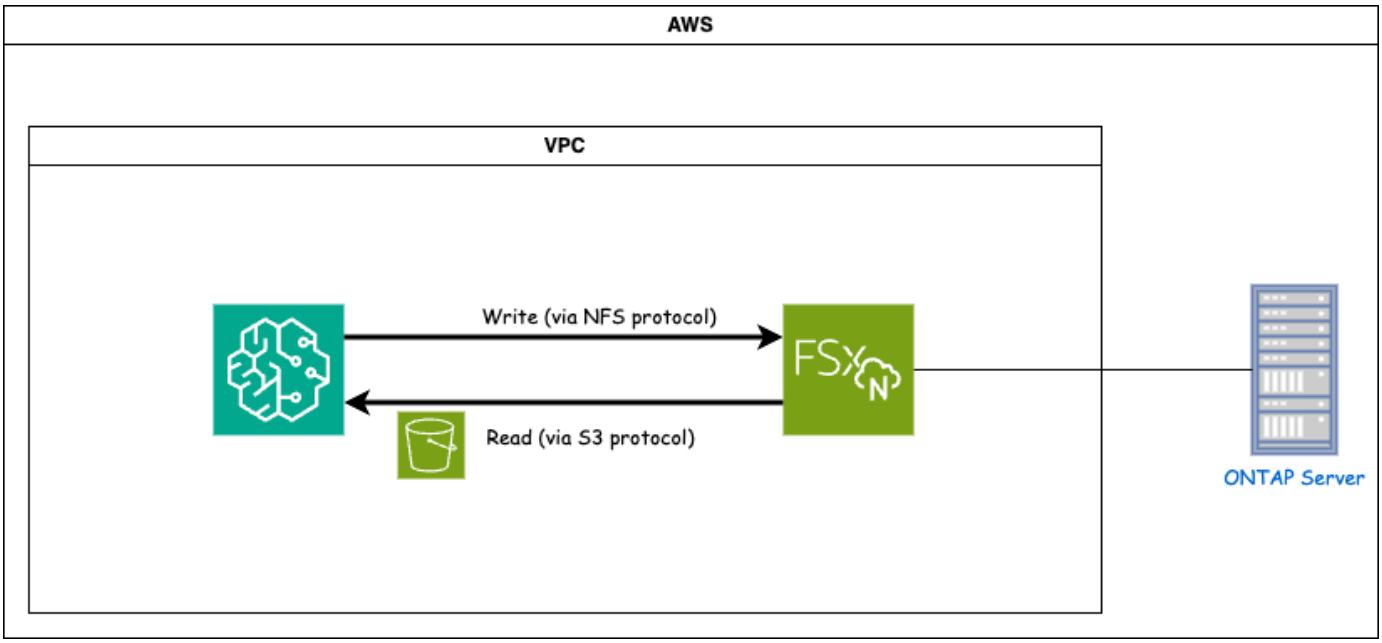
Questo tutorial offre un esempio pratico di un progetto di classificazione della computer vision, che fornisce esperienza pratica nella creazione di modelli ML che utilizzano FSxN come origine dati all'interno dell'ambiente SageMaker. Il progetto si concentra sull'utilizzo di PyTorch, un framework di apprendimento approfondito, per classificare la qualità degli pneumatici in base alle immagini degli pneumatici. Enfatizza lo sviluppo di modelli di machine learning utilizzando FSxN come fonte di dati in Amazon SageMaker.

Che cos'è FSxN

Amazon FSX per NetApp ONTAP è in realtà una soluzione di storage completamente gestita offerta da AWS. Sfrutta il file system ONTAP di NetApp per fornire storage affidabile e dalle performance elevate. Grazie al supporto per protocolli come NFS, SMB e iSCSI, permette l'accesso perfetto da diversi container e istanze di calcolo. Il servizio è progettato per offrire performance eccezionali, garantendo operazioni sui dati rapide ed efficienti. Inoltre, offre high Availability e durata elevata per garantire che i tuoi dati rimangano accessibili e protetti. Inoltre, la capacità storage di Amazon FSX per NetApp ONTAP è scalabile e ti permette di regolarla facilmente in base alle tue esigenze.

Prerequisito

Ambiente di rete



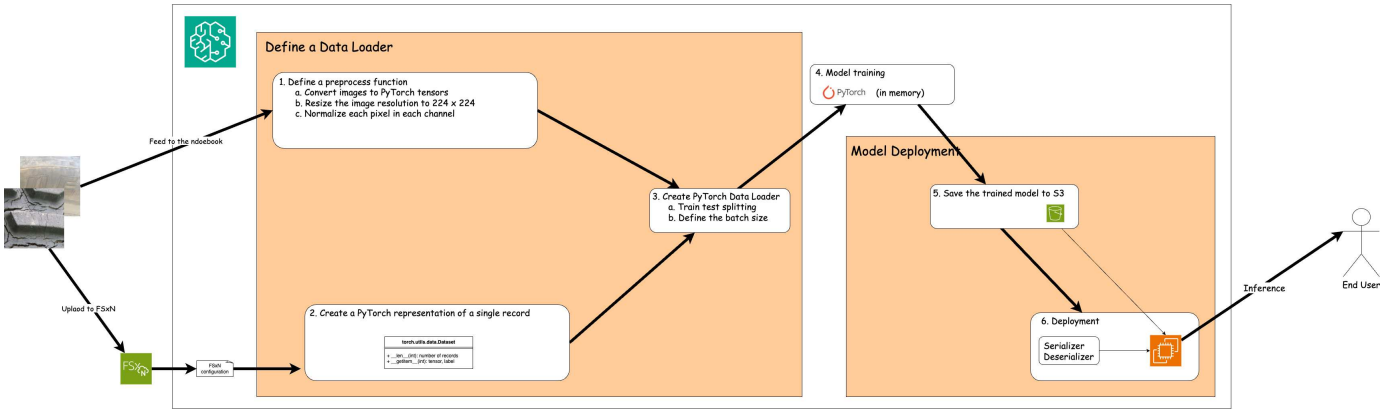
FSxN (Amazon FSX per NetApp ONTAP) è un servizio di storage AWS. Include un file system in esecuzione sul sistema NetApp ONTAP e una SVM (System Virtual Machine) gestita da AWS che si connette all'IT. Nel diagramma fornito, il server NetApp ONTAP gestito da AWS si trova all'esterno del VPC. La SVM funge da intermediario tra SageMaker e il sistema NetApp ONTAP, ricevendo le richieste operative da SageMaker e inoltrandole allo storage sottostante. Per accedere a FSxN, SageMaker deve essere collocato nello stesso VPC della distribuzione di FSxN. Questa configurazione garantisce la comunicazione e l'accesso ai dati tra SageMaker e FSxN.

Accesso ai dati

Negli scenari reali, i data scientist utilizzano in genere i dati esistenti memorizzati in FSxN per costruire i propri modelli di machine learning. Tuttavia, per scopi dimostrativi, poiché il file system FSxN è inizialmente vuoto dopo la creazione, è necessario caricare manualmente i dati di addestramento. Questo può essere ottenuto montando FSxN come volume a SageMaker. Una volta montato correttamente il file system, è possibile caricare il set di dati nella posizione montata, rendendolo accessibile per l'addestramento dei modelli all'interno dell'ambiente SageMaker. Questo approccio consente di sfruttare la capacità di storage e le funzionalità di FSxN lavorando con SageMaker per lo sviluppo e la formazione dei modelli.

Il processo di lettura dei dati prevede la configurazione di FSxN come bucket S3 privato. Per istruzioni dettagliate sulla configurazione, fare riferimento alla "Parte 1 - integrazione di AWS FSX per NetApp ONTAP (FSxN) come bucket S3 privato in AWS SageMaker"

Panoramica sull'integrazione



Il flusso di lavoro di utilizzo dei dati di formazione in FSxN per creare un modello di apprendimento approfondito in SageMaker può essere riassunto in tre fasi principali: Definizione del caricatore dati, formazione del modello e distribuzione. Ad alto livello, questi passaggi costituiscono la base di una pipeline MLOps. Tuttavia, ogni fase prevede diverse fasi secondarie dettagliate per un'implementazione completa. Queste fasi secondarie comprendono varie attività come la pre-elaborazione dei dati, la suddivisione del dataset, la configurazione del modello, la regolazione dell'iperparametro, la valutazione del modello, e distribuzione dei modelli. Questi passaggi garantiscono un processo completo ed efficace per la creazione e l'implementazione di modelli di apprendimento approfondito utilizzando i dati di formazione di FSxN all'interno dell'ambiente SageMaker.

Integrazione step-by-step

Caricatore dati

Per addestrare una rete di apprendimento profondo PyTorch con i dati, viene creato un caricatore dati per facilitare l'alimentazione dei dati. Il caricatore dati non solo definisce la dimensione del batch, ma determina anche la procedura di lettura e pre-elaborazione di ciascun record all'interno del batch. Configurando il data loader, possiamo gestire l'elaborazione dei dati in batch, consentendo la formazione della rete di deep learning.

Il caricatore dati è composto da 3 parti.

Funzione di pre-elaborazione

```
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]
    )
])
```

Il frammento di codice riportato sopra illustra la definizione delle trasformazioni di pre-elaborazione delle immagini utilizzando il modulo **torchvision.Transforms**. In questa turtorial, l'oggetto di pre-elaborazione viene creato per applicare una serie di trasformazioni. In primo luogo, la trasformazione **ToTensor()** converte

l'immagine in una rappresentazione tensoriale. Successivamente, la trasformazione **Ridimensiona((224.224)** ridimensiona l'immagine a una dimensione fissa di 224x224 pixel. Infine, la trasformazione **Normalize()** normalizza i valori del tensore sottraendo la media e dividendo per la deviazione standard lungo ciascun canale. I valori di deviazione media e standard utilizzati per la normalizzazione sono comunemente impiegati in modelli di rete neurale pre-addestrati. Nel complesso, questo codice prepara i dati dell'immagine per un'ulteriore elaborazione o immissione in un modello pre-addestrato convertendoli in un tensore, ridimensionandoli e normalizzando i valori dei pixel.

Classe dataset 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
```

Questa classe fornisce funzionalità per ottenere il numero totale di record nell'insieme di dati e definisce il metodo di lettura dei dati per ogni record. All'interno della funzione ***getitem***, il codice utilizza l'oggetto bucket boto3 S3 per recuperare i dati binari da FSxN. Lo stile del codice per accedere ai dati da FSxN è simile alla lettura dei dati da Amazon S3. La spiegazione successiva si sofferma sul processo di creazione dell'oggetto S3 privato **bucket**.

FSxN come repository S3 privato

```
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 ---
```

Per leggere i dati da FSxN in SageMaker, viene creato un gestore che punta allo storage FSxN utilizzando il protocollo S3. Ciò consente a FSxN di essere trattato come un bucket S3 privato. La configurazione del gestore include l'indicazione dell'indirizzo IP della SVM FSxN, del nome del bucket e delle credenziali necessarie. Per una spiegazione completa su come ottenere questi elementi di configurazione, fare riferimento al documento all'indirizzo "Parte 1 - integrazione di AWS FSX per NetApp ONTAP (FSxN) come bucket S3 privato in AWS SageMaker".

Nell'esempio sopra menzionato, l'oggetto bucket viene utilizzato per creare un'istanza dell'oggetto dataset PyTorch. L'oggetto dataset verrà ulteriormente spiegato nella sezione successiva.

Il caricatore dati 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)
```

Nell'esempio fornito, viene specificata una dimensione batch di 64, che indica che ogni batch conterrà 64 record. Combinando la classe PyTorch **dataset**, la funzione di pre-elaborazione e la dimensione del batch di training, otteniamo il caricatore dati per la formazione. Questo caricatore dati facilita il processo di iterazione del set di dati in batch durante la fase di training.

Training sui modelli

```
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')
```

Questo codice implementa un processo di formazione PyTorch standard. Definisce un modello di rete neurale chiamato **TyreQualityClassifier** utilizzando strati convoluzionali e uno strato lineare per classificare la qualità dei pneumatici. Il ciclo di training itera i batch di dati, calcola la perdita e aggiorna i parametri del modello utilizzando la backpropagation e l'ottimizzazione. Inoltre, stampa l'ora corrente, l'epoca, il batch e la perdita a scopo di monitoraggio.

Implementazione dei modelli

Implementazione

```
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())
```

Il codice salva il modello PyTorch in **Amazon S3** perché SageMaker richiede che il modello venga memorizzato in S3 per la distribuzione. Caricando il modello su **Amazon S3**, diventa accessibile a SageMaker, consentendo la distribuzione e l'inferenza sul modello distribuito.

```
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
)
```

Questo codice facilita la distribuzione di un modello PyTorch su SageMaker. Definisce un serializzatore personalizzato, **TyreQualitySerializer**, che preelabora e serializza i dati di input come un tensor PyTorch. La classe **TyreQualityPredictor** è un predittore personalizzato che utilizza il serializzatore definito e un **JSONDeserializer**. Il codice crea inoltre un oggetto **PyTorchModel** per specificare la posizione S3 del modello, il ruolo IAM, la versione del framework e il punto di ingresso per l'inferenza. Il codice genera un

indicatore data e ora e costruisce un nome endpoint in base al modello e all'indicatore data e ora. Infine, il modello viene distribuito utilizzando il metodo Deploy, specificando il numero di istanze, il tipo di istanza e il nome dell'endpoint generato. In questo modo, il modello PyTorch può essere distribuito e accessibile per l'inferenza su SageMaker.

Inferenza

```
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)
```

Questo è l'esempio di utilizzo dell'endpoint distribuito per l'inferenza.

Parte 3 - creazione di Una pipeline MLOps semplificata (ci/CT/CD)

Autore(i):
Jian Jian (Ken), Senior Data & Applied Scientist, NetApp

Introduzione

In questo tutorial scoprirai come sfruttare i vari servizi AWS per costruire una semplice pipeline MLOps che comprende Continuous Integration (ci), Continuous Training (CT) e Continuous Deployment (CD). A differenza delle tradizionali pipeline DevOps, gli MLOps richiedono ulteriori considerazioni per completare il ciclo operativo. Seguendo questo tutorial, si acquisiranno informazioni sull'integrazione della TC nel loop MLOps, consentendo una formazione continua dei modelli e una distribuzione perfetta per l'inferenza. Il tutorial ti guiderà attraverso il processo di utilizzo dei servizi AWS per creare questa pipeline MLOps end-to-end.

Manifesto

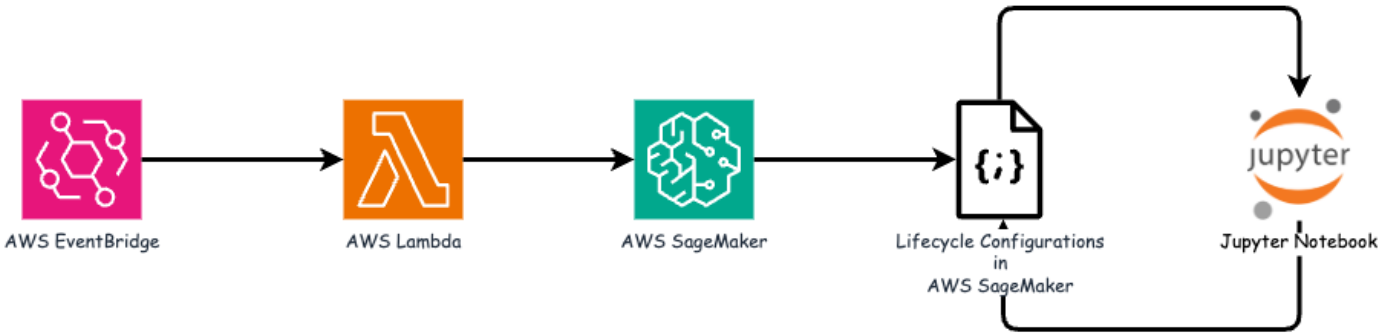
Funzionalità	Nome	Commento
Storage dei dati	AWS FSxN	Fare riferimento a. "Parte 1 - integrazione di AWS FSX per NetApp ONTAP (FSxN) come bucket S3 privato in AWS SageMaker".
IDE di data science	AWS SageMaker	Questa esercitazione si basa sul notebook Jupyter presentato in "Parte 2 - utilizzo di AWS FSX for NetApp ONTAP (FSxN) come origine dati per il training sui modelli in SageMaker".
Funzione per attivare la pipeline MLOps	Funzione AWS Lambda	-
Trigger di job cron	AWS EventBridge	-

Funzionalità	Nome	Commento
Framework di deep learning	PyTorch	-
SDK AWS Python	boto3	-
Linguaggio di programmazione	Python	v3,10

Prerequisito

- Un file system FSxN preconfigurato. Questa esercitazione utilizza i dati memorizzati in FSxN per il processo di formazione.
- Un'istanza **SageMaker notebook** configurata per condividere lo stesso VPC del file system FSxN menzionato sopra.
- Prima di attivare la funzione **AWS Lambda**, assicurarsi che l'istanza **SageMaker notebook** sia nello stato **Stopped**.
- Il tipo di istanza **ml.g4dn.xlarge** è necessario per sfruttare l'accelerazione GPU necessaria per i calcoli delle reti neurali profonde.

Architettura



Questa pipeline MLOps è un'implementazione pratica che utilizza un job cron per attivare una funzione senza server, che a sua volta esegue un servizio AWS registrato con una funzione di callback del ciclo di vita. Il **AWS EventBridge** agisce come job cron. Richiama periodicamente una funzione **AWS Lambda** responsabile del riaddestramento e della ridistribuzione del modello. Questo processo comporta la creazione dell'istanza **AWS SageMaker notebook** per eseguire le attività necessarie.

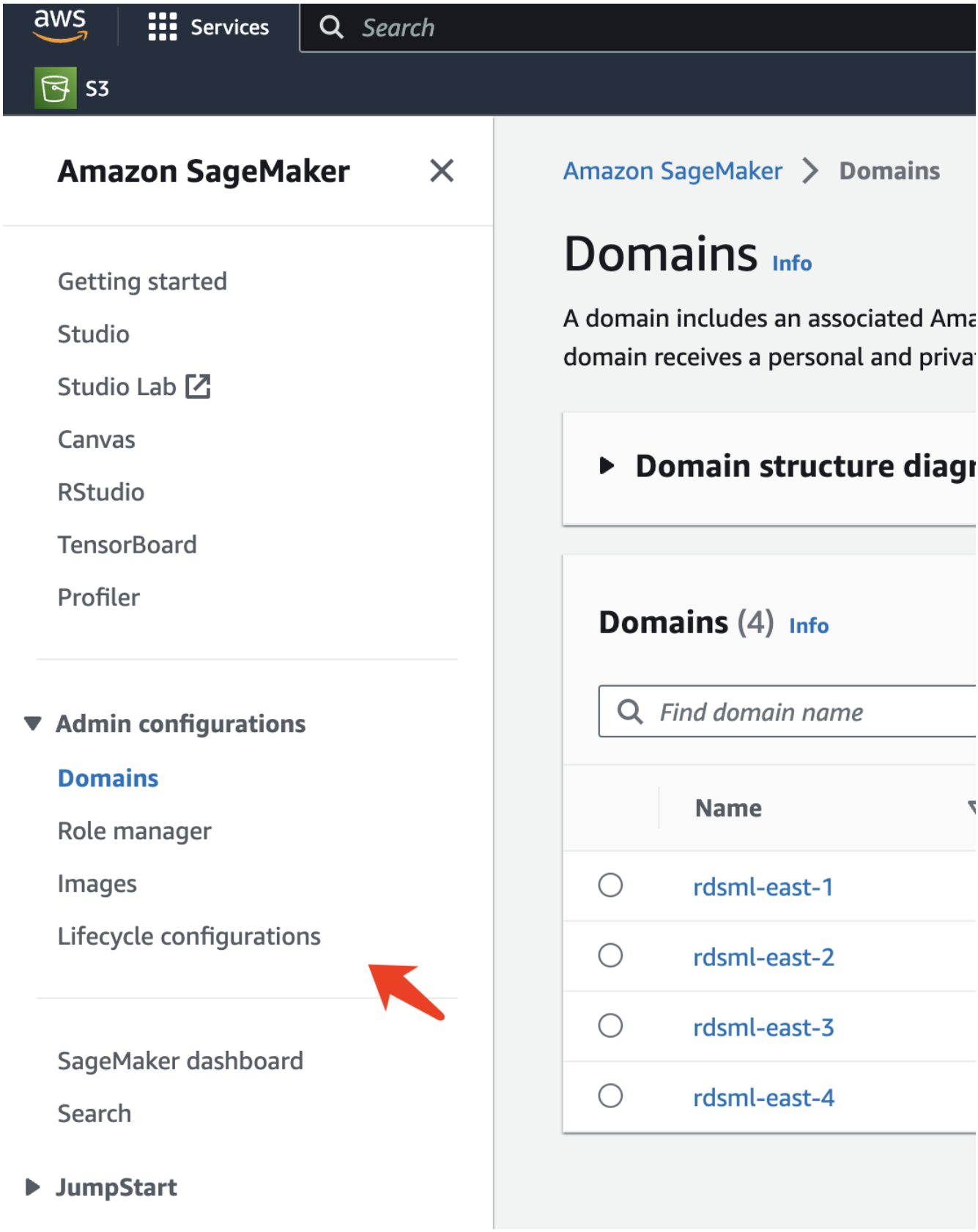
Configurazione dettagliata

Configurazioni del ciclo di vita

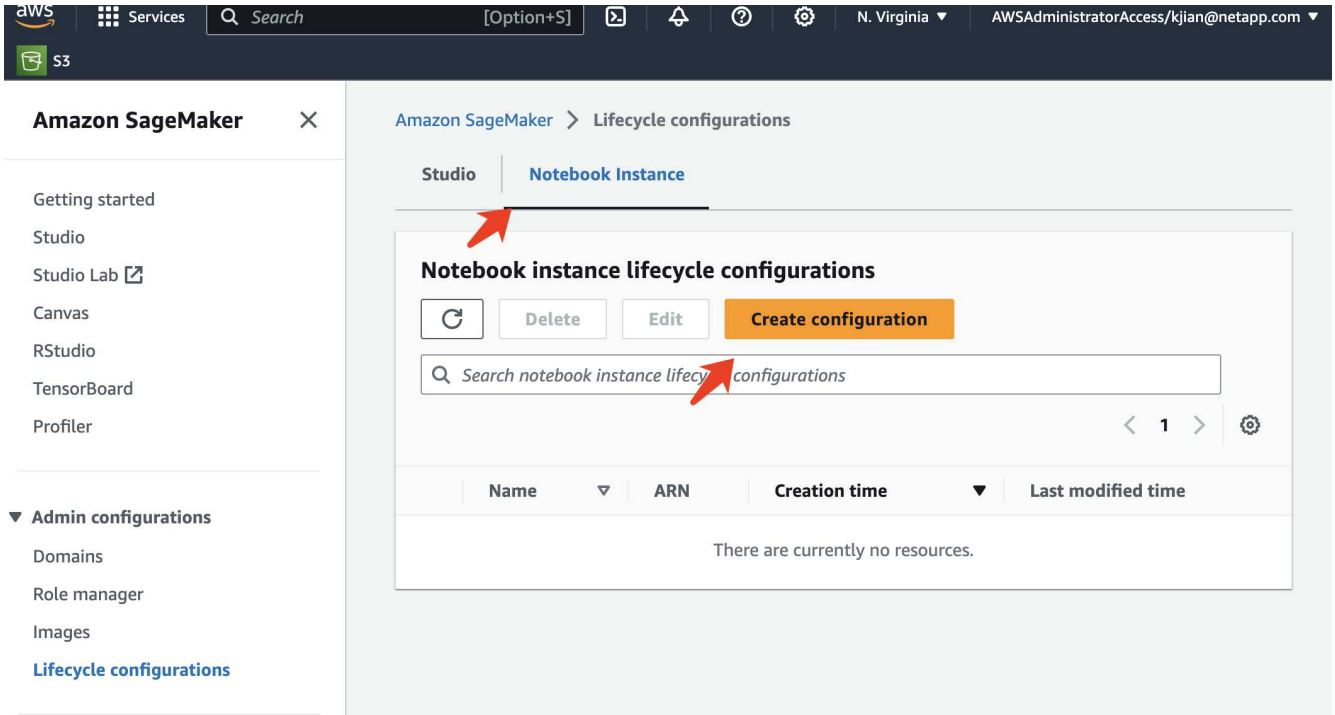
Per configurare la funzione di callback del ciclo di vita per l'istanza del notebook AWS SageMaker, utilizzare **configurazioni del ciclo di vita**. Questo servizio consente di definire le azioni necessarie da eseguire durante la rotazione dell'istanza del notebook. In particolare, è possibile implementare uno script della shell all'interno delle **configurazioni del ciclo di vita** per arrestare automaticamente l'istanza del notebook una volta completati i processi di formazione e distribuzione. Si tratta di una configurazione richiesta, in quanto il costo è una delle considerazioni principali di MLOps.

È importante notare che la configurazione per **configurazioni del ciclo di vita** deve essere impostata in anticipo. Pertanto, si consiglia di assegnare una priorità alla configurazione di questo aspetto prima di procedere con l'altra impostazione della pipeline MLOps.

1. Per impostare una configurazione del ciclo di vita, aprire il pannello **Sagemaker** e passare a **configurazioni del ciclo di vita** nella sezione **configurazioni amministratore**.



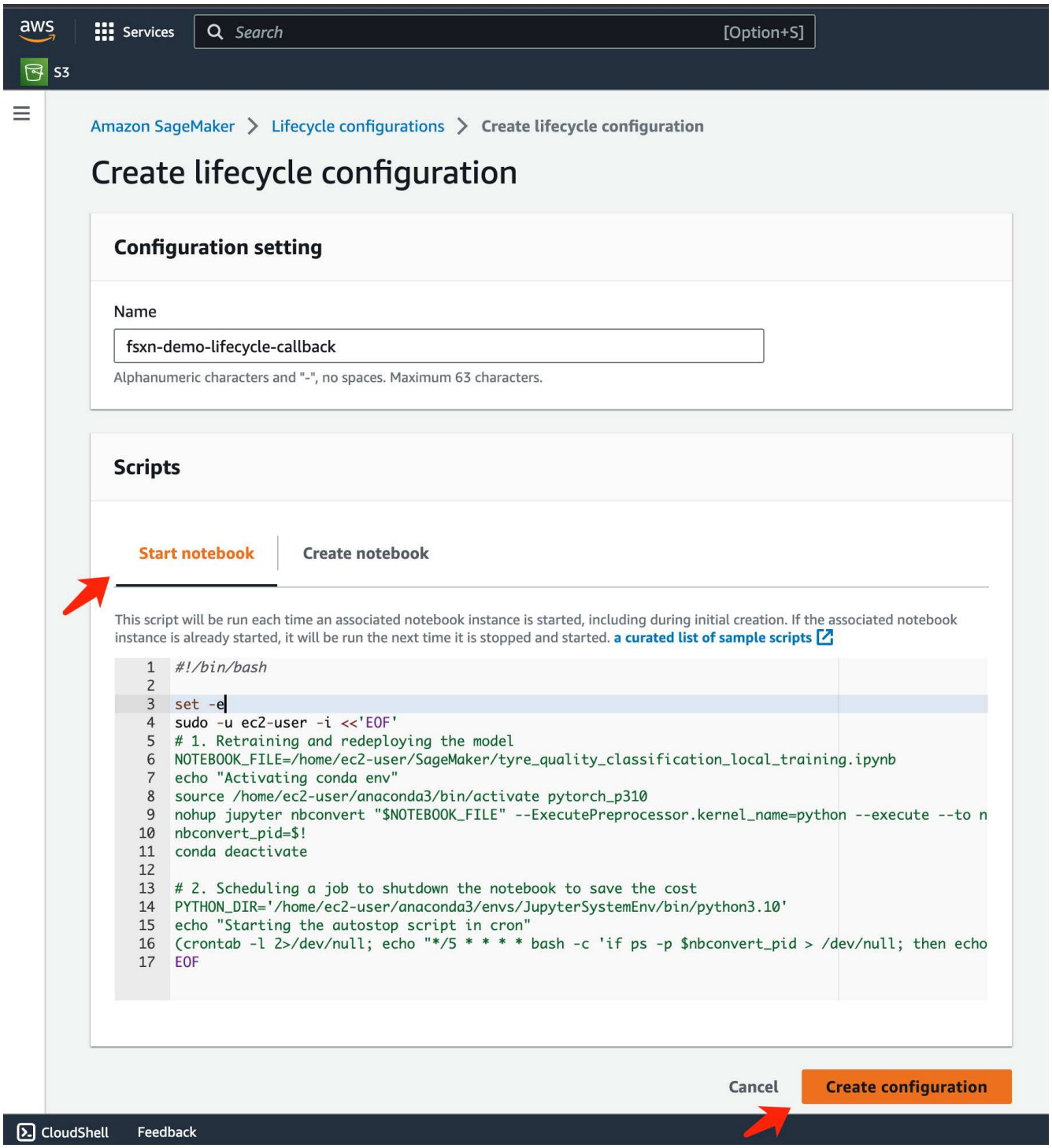
2. Selezionare la scheda **istanza notebook** e fare clic sul pulsante **Crea configurazione**



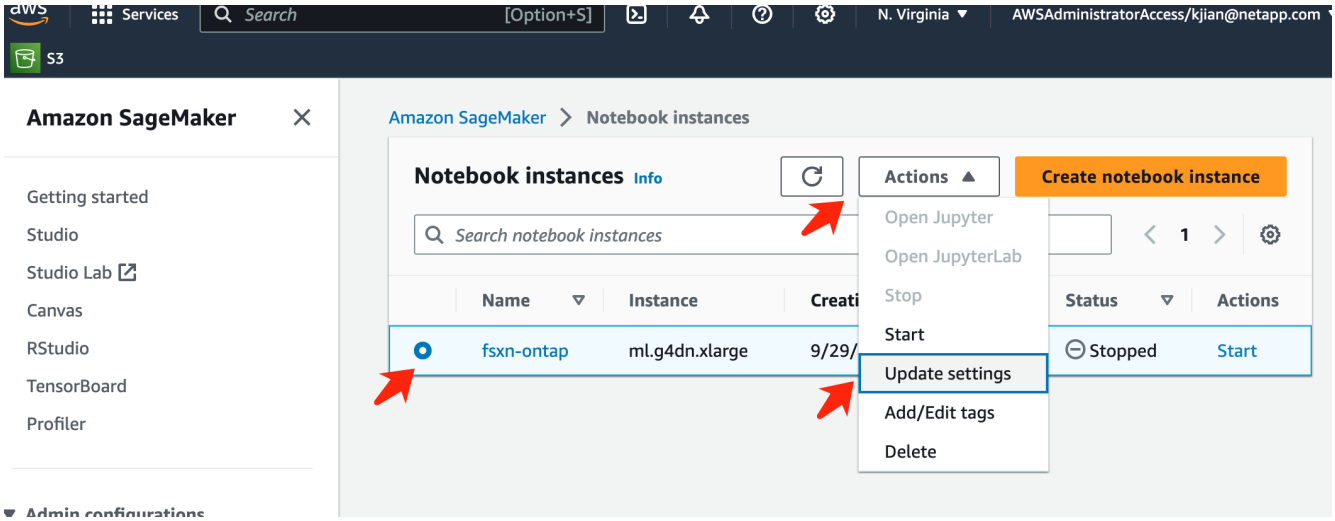
3. Incollare il codice riportato di seguito nell'area di immissione.

```
#!/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
```

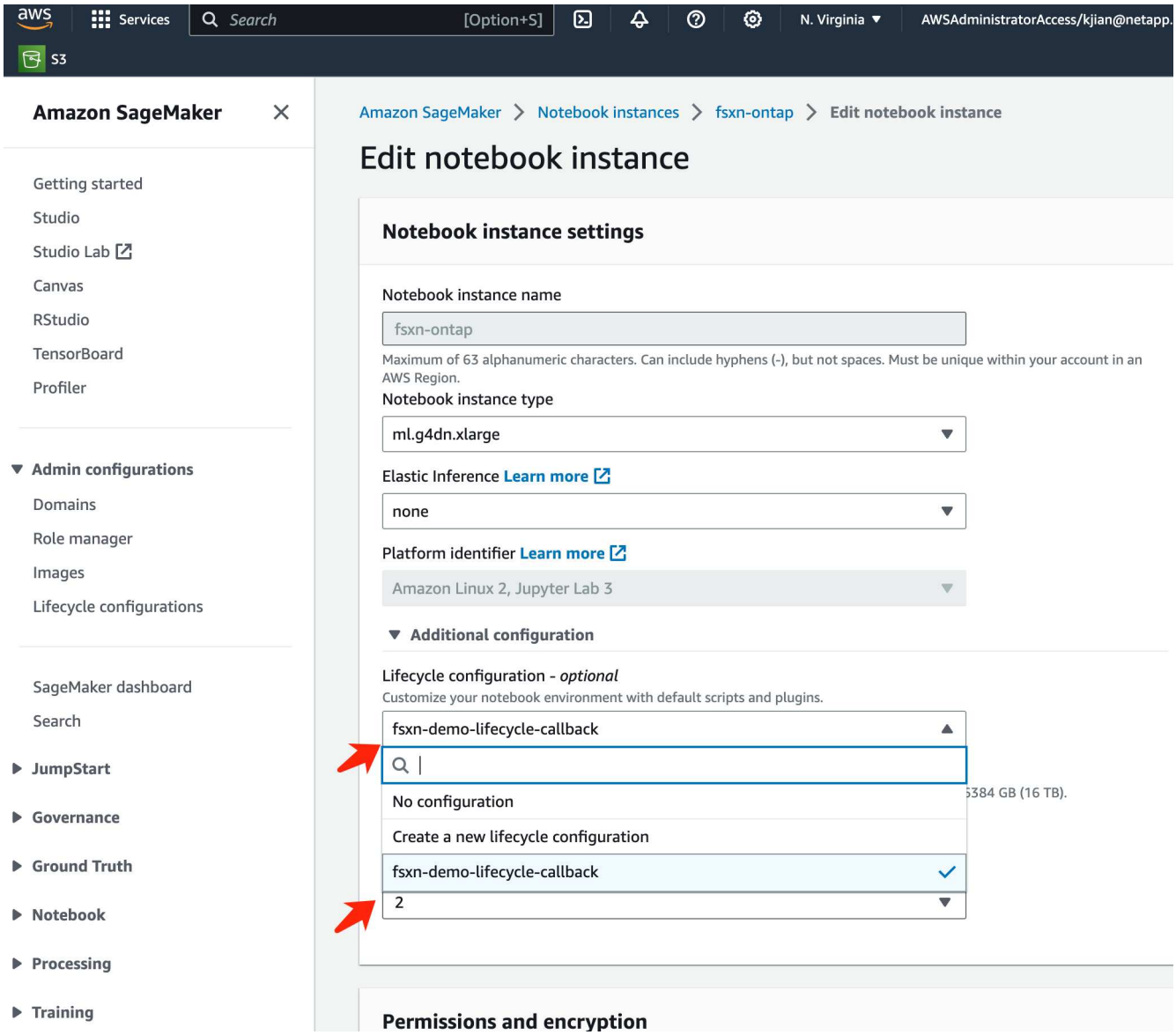
4. Questo script esegue il notebook Jupyter, che gestisce il riaddestramento e la ridistribuzione del modello per l'inferenza. Al termine dell'esecuzione, il notebook si spegne automaticamente entro 5 minuti. Per ulteriori informazioni sull'istruzione Problem e sull'implementazione del codice, fare riferimento a. "Parte 2 - utilizzo di AWS FSX for NetApp ONTAP (FSxN) come origine dati per il training sui modelli in SageMaker".



5. Dopo la creazione, accedere a istanze notebook, selezionare l'istanza di destinazione e fare clic su **Aggiorna impostazioni** nel menu a discesa azioni.



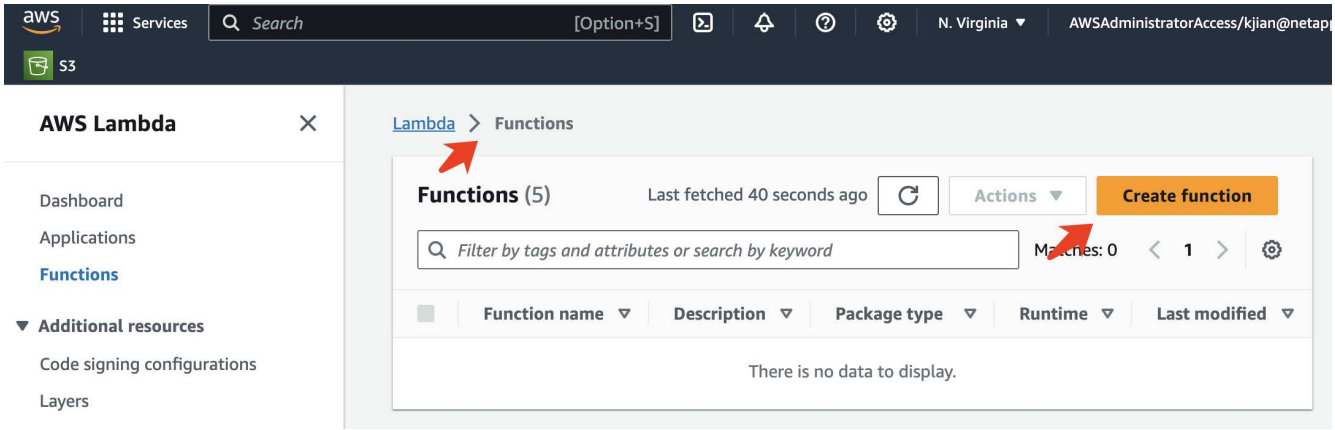
6. Selezionare la configurazione **Lifecycle** creata e fare clic su **Aggiorna istanza notebook**.



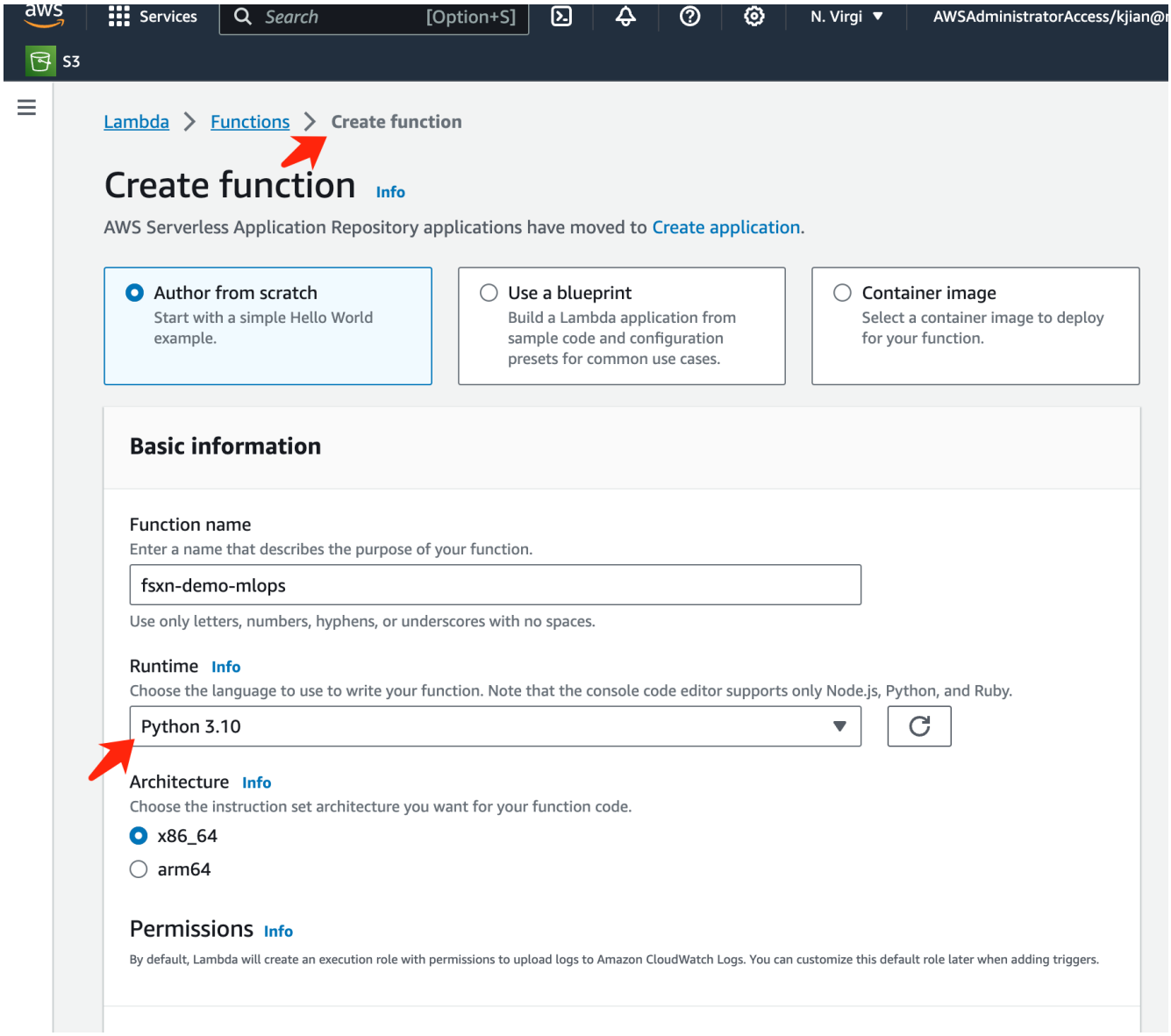
Funzione serverless di AWS Lambda

Come accennato in precedenza, la funzione **AWS Lambda** è responsabile della creazione dell'istanza **AWS SageMaker notebook**.

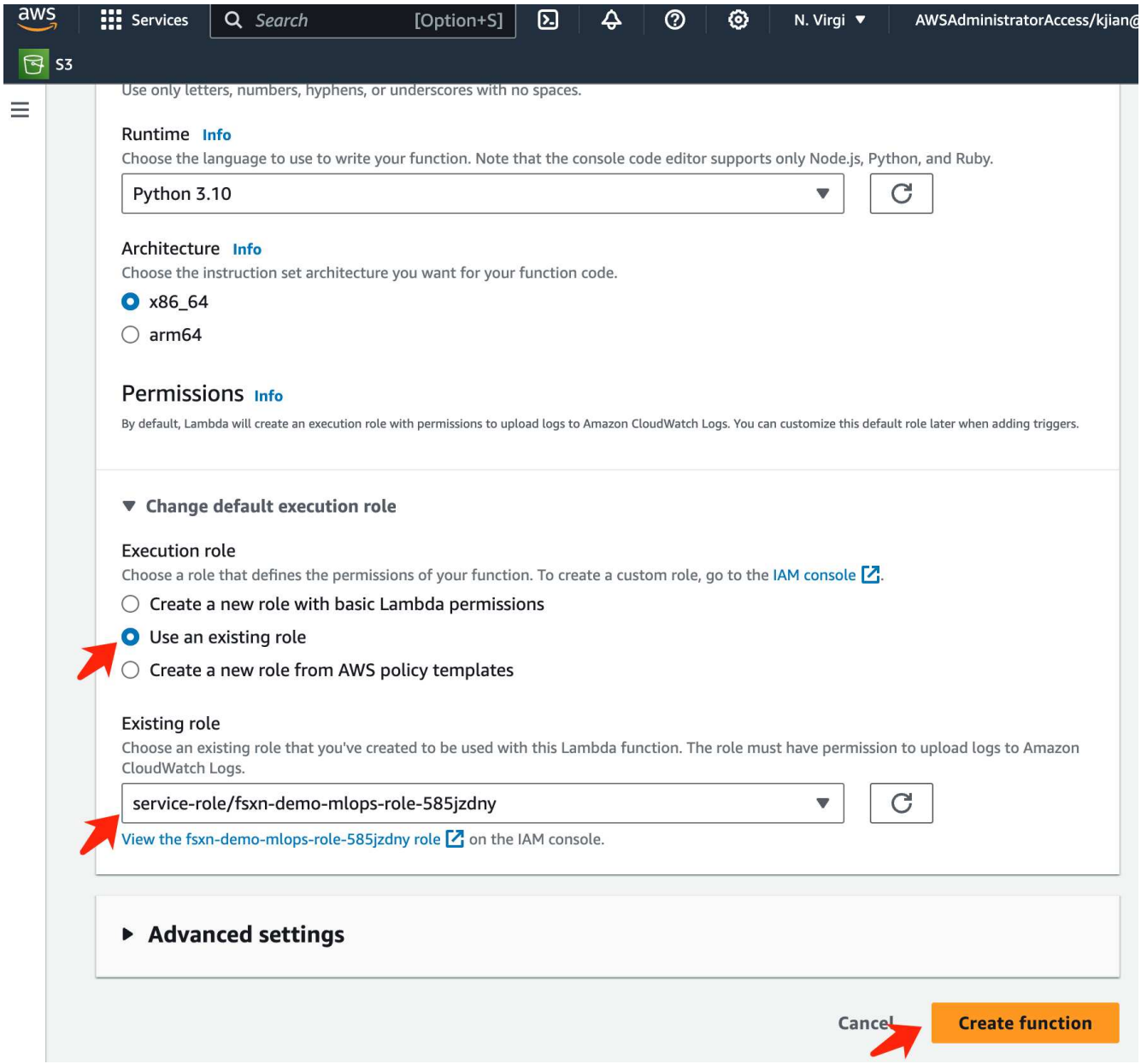
1. Per creare una funzione **AWS Lambda**, accedere al pannello corrispondente, passare alla scheda **funzioni** e fare clic su **Crea funzione**.



2. Si prega di archiviare tutte le voci necessarie nella pagina e ricordarsi di cambiare il Runtime a **Python 3,10**.



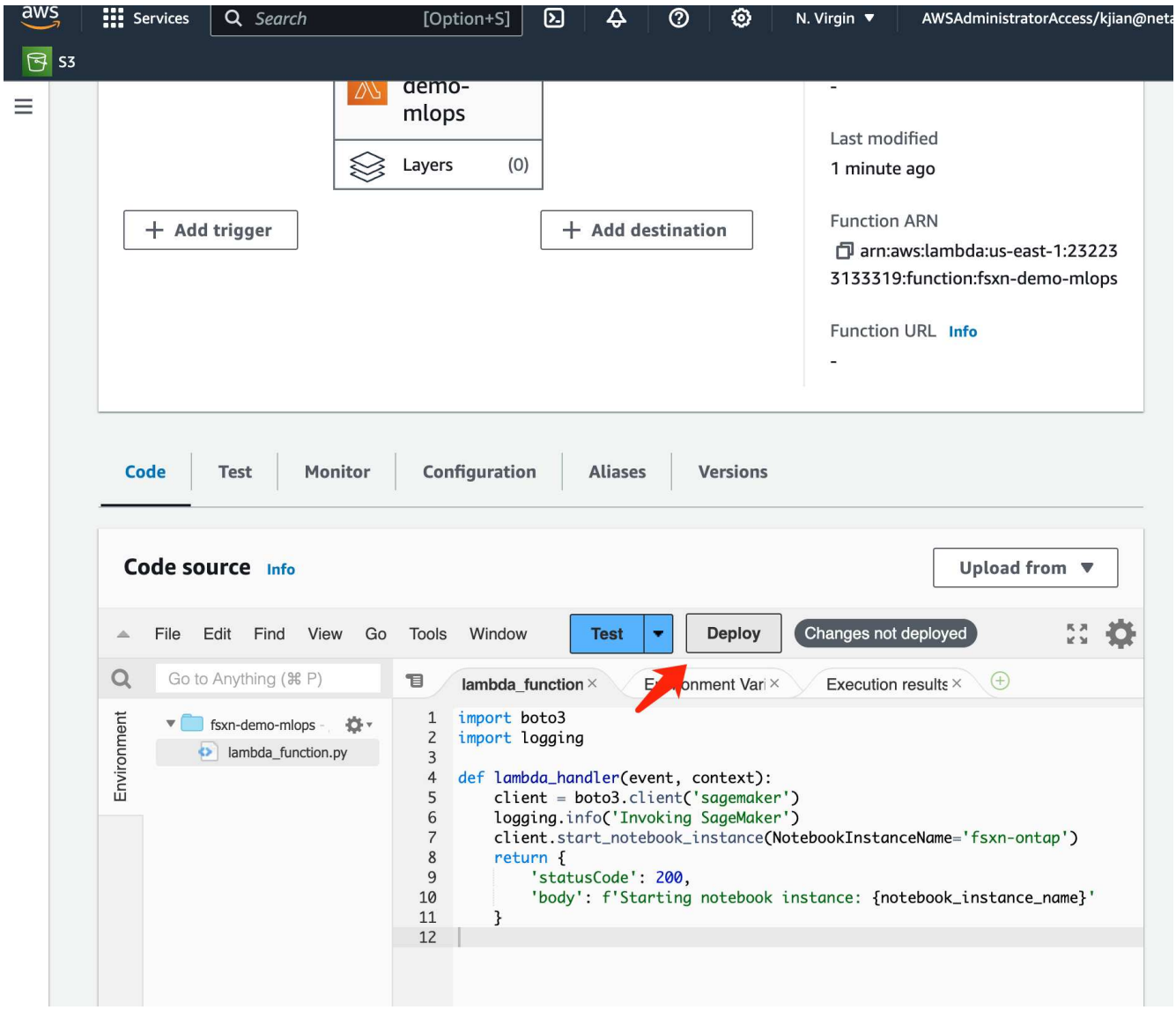
3. Verificare che il ruolo designato disponga dell'autorizzazione richiesta **AmazonSageMakerFullAccess** e fare clic sul pulsante **Crea funzione**.



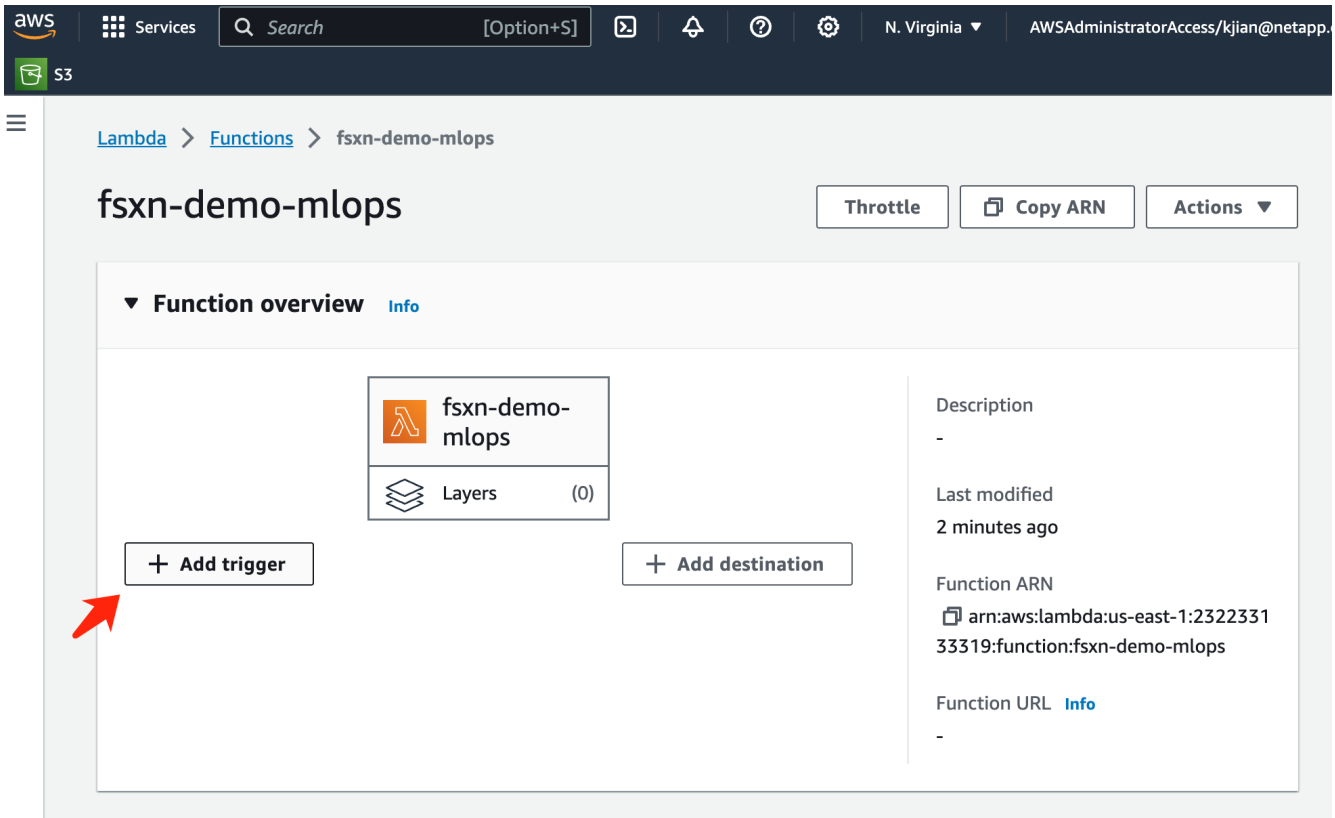
4. Selezionare la funzione Lambda creata. Nella scheda Codice, copiare e incollare il seguente codice nell'area di testo. Questo codice avvia l'istanza del notebook denominata **fsxn-ontap**.

```
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}'
    }
```

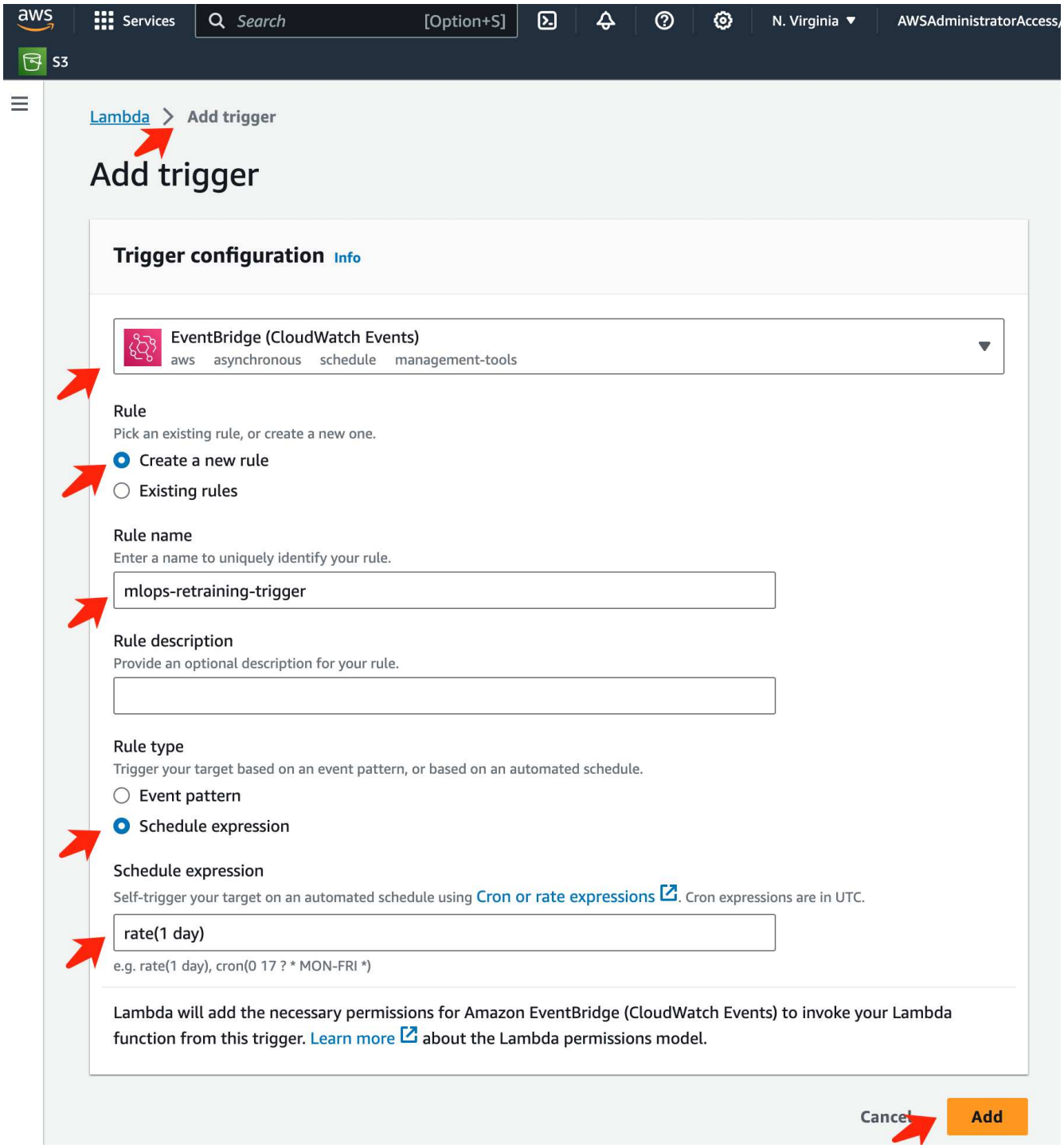
5. Fare clic sul pulsante **Deploy** per applicare questa modifica di codice.



6. Per specificare come attivare questa funzione AWS Lambda, fare clic sul pulsante Add Trigger (Aggiungi trigger).



7. Selezionare EventBridge dal menu a discesa, quindi fare clic sul pulsante di opzione Crea una nuova regola. Nel campo espressione pianificazione, immettere `rate(1 day)`, Quindi fare clic sul pulsante Aggiungi per creare e applicare questa nuova regola del job cron alla funzione AWS Lambda.



Dopo aver completato la configurazione in due fasi, su base giornaliera, la funzione **AWS Lambda** avvierà il notebook **SageMaker**, eseguirà il riaddestramento del modello utilizzando i dati del repository **FSxN**, ridistribuirà il modello aggiornato nell'ambiente di produzione e spegnerà automaticamente l'istanza **SageMaker notebook** per ottimizzare i costi. In questo modo, il modello rimane aggiornato.

Questo conclude il tutorial per lo sviluppo di una pipeline MLOps.

MLOps multicloud ibrido con Domino Data Lab e NetApp

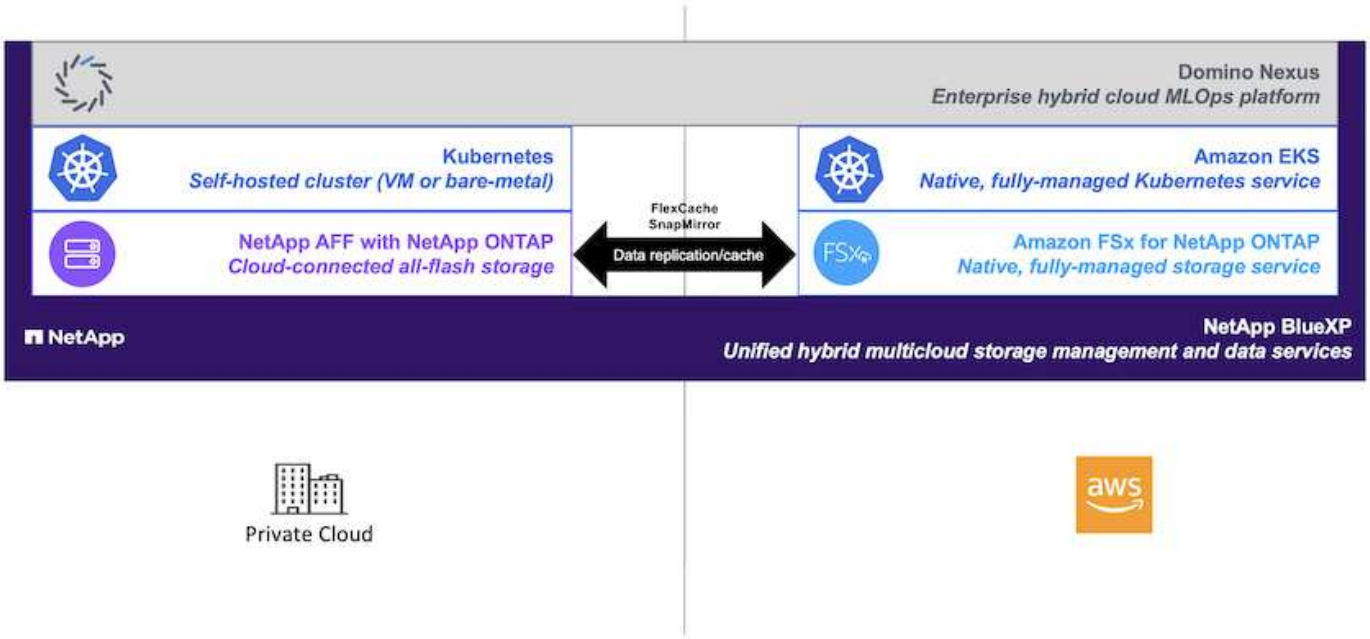
MLOps multicloud ibrido con Domino Data Lab e NetApp

Mike Oglesby, NetApp

Le organizzazioni di tutto il mondo stanno attualmente adottando l'ai per trasformare i propri business e processi. Per questo motivo, l'infrastruttura di calcolo ai-ready è spesso poco disponibile. Le aziende stanno adottando architetture MLOps di multicloud ibrido al fine di sfruttare gli ambienti di calcolo disponibili in regioni, data center e cloud diversi, bilanciando costi, disponibilità e performance.

Domino Nexus, di Domino Data Lab, è un piano di controllo MLOps unificato che consente di eseguire carichi di lavoro di data science e machine learning su qualsiasi cluster di calcolo, in qualsiasi cloud, regione o on-premise. Unifica i silos di data science in tutta l'azienda, in modo da avere un unico posto per creare, implementare e monitorare i modelli. Allo stesso modo, le funzionalità di gestione dei dati nel cloud ibrido di NetApp ti permettono di portare i dati nelle tue posizioni lavorative e in spazi di lavoro, indipendentemente da dove siano in esecuzione. Abbinando Domino Nexus a NetApp, puoi pianificare i carichi di lavoro in più ambienti senza doverti preoccupare della disponibilità dei dati. In altre parole, hai la possibilità di inviare i tuoi carichi di lavoro e i tuoi dati all'ambiente di calcolo appropriato, consentendoti di accelerare le implementazioni di IA durante la navigazione nelle normative relative a privacy e sovranità dei dati.

Questa soluzione dimostra l'implementazione di un piano di controllo MLOps unificato, che incorpora un cluster Kubernetes on-premise e un cluster EKS (Elastic Kubernetes Service) in esecuzione in Amazon Web Services (AWS).



Panoramica sulla tecnologia

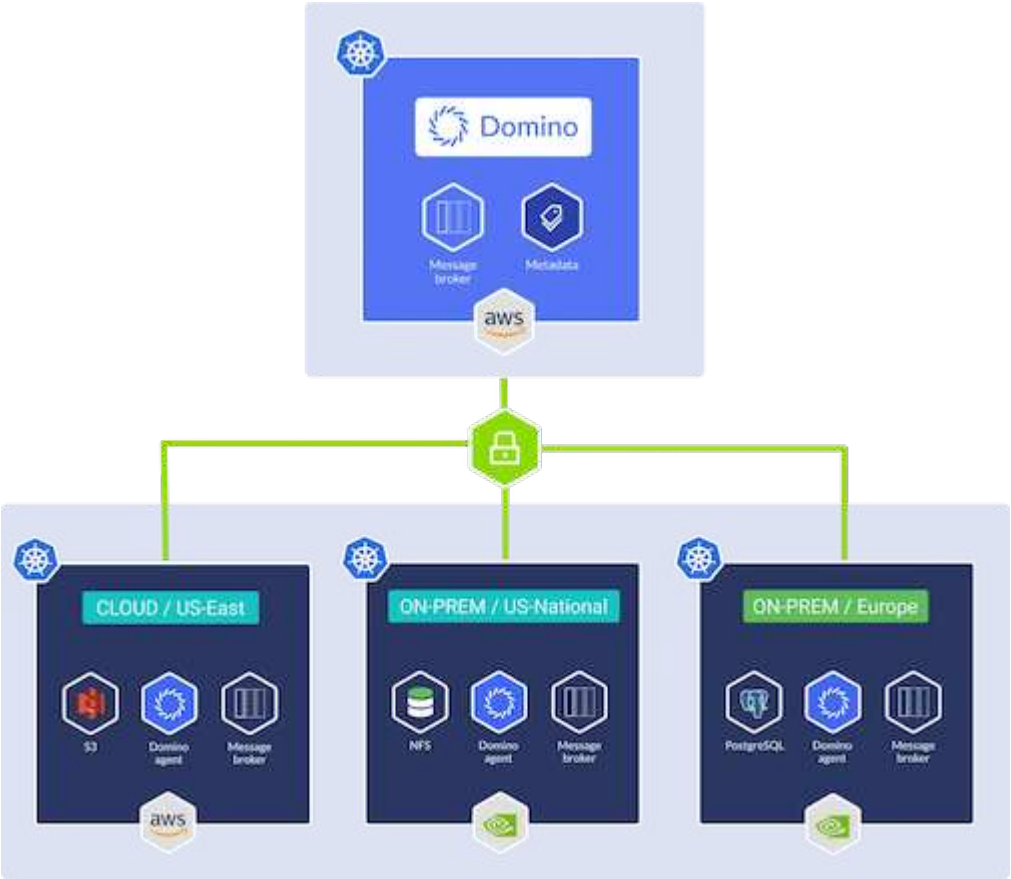
Domino Data Lab

Domino Data Lab potenzia le aziende basate su modelli con la sua piattaforma Enterprise ai leader del settore, scelta da oltre il 20% delle aziende Fortune 100. Domino accelera lo sviluppo e l'implementazione del lavoro di data science, aumentando al contempo la collaborazione e la governance. Con Domino, le aziende di tutto il mondo possono sviluppare farmaci migliori, coltivare colture più produttive, costruire auto migliori e molto altro. Fondata nel 2013, Domino è sostenuta da Coatue Management, Great Hill Partners, Highland Capital,

Sequoia Capital e da altri importanti investitori.

Domino consente alle aziende e ai loro data scientist di creare, implementare e gestire l'intelligenza artificiale su una piattaforma end-to-end unificata, rapida, responsabile e conveniente. I team possono accedere a tutti i dati, gli strumenti, il calcolo, i modelli e i progetti di cui hanno bisogno in qualsiasi ambiente, in modo da collaborare, riutilizzare il lavoro passato, tenere traccia dei modelli in produzione per migliorare l'accuratezza, standardizzare con le Best practice e rendere l'ai responsabile e governata.

- **Aperto e flessibile:** accedete al più ampio ecosistema di strumenti open source e commerciali, e infrastrutture, per le migliori innovazioni e nessun dipendenza dal vendor.
- **System of Record:** hub centrale per le operazioni e la conoscenza dell'intelligenza artificiale in tutta l'azienda, che consente Best practice, collaborazione interfunzionale, innovazione più rapida ed efficienza.
- **Integrato:** flussi di lavoro e automazione integrati — costruiti per processi, controlli e governance aziendali — soddisfano le vostre esigenze di conformità e normative.
- **Multicloud ibrido:** Esegui i workload ai vicino ai tuoi dati ovunque, on-premise, ibrido, qualsiasi cloud o multicloud, per costi inferiori, performance ottimali e compliance.



Domino Nexus

Domino Nexus è un singolo pannello di controllo che consente di eseguire carichi di lavoro di data science e machine learning su qualsiasi cluster di calcolo, in qualsiasi cloud, regione o on-premise. Unifica i silos di data science in tutta l'azienda, in modo da avere un unico posto per creare, implementare e monitorare i modelli.

NetApp BlueXP

NetApp BlueXP unifica tutti i servizi dati e storage di NetApp in un singolo tool che ti consente di creare, proteggere e gestire il tuo ambiente dati multicloud ibrido. Offre un'esperienza unificata per lo storage e i

servizi dati in ambienti on-premise e cloud e abilita la semplicità operativa attraverso la potenza di AIOps, con i parametri di consumo flessibili e la protezione integrata richiesti per il mondo di oggi basato sul cloud.

NetApp ONTAP

ONTAP 9, l'ultima generazione di software per la gestione dello storage NetApp, consente alle aziende di modernizzare l'infrastruttura e passare a un data center predisposto per il cloud. Sfruttando le funzionalità di gestione dei dati leader del settore, ONTAP consente la gestione e la protezione dei dati con un singolo set di strumenti, indipendentemente dalla posizione dei dati. Puoi anche spostare liberamente i dati ovunque siano necessari: Edge, core o cloud. ONTAP 9 include numerose funzionalità che semplificano la gestione dei dati, accelerano e proteggono i dati critici e abilitano le funzionalità dell'infrastruttura di nuova generazione nelle architetture di cloud ibrido.

Semplifica la gestione dei dati

La gestione dei dati è fondamentale per le operazioni IT aziendali e per i data scientist, in modo che le risorse appropriate vengano utilizzate per le applicazioni ai e per la formazione dei set di dati ai/ML. Le seguenti informazioni aggiuntive sulle tecnologie NetApp non rientrano nell'ambito di questa convalida, ma potrebbero essere rilevanti a seconda dell'implementazione.

Il software per la gestione dei dati ONTAP include le seguenti funzionalità per ottimizzare e semplificare le operazioni e ridurre il costo totale delle operazioni:

- Compaction dei dati inline e deduplica estesa. La compattazione dei dati riduce lo spazio sprecato all'interno dei blocchi di storage e la deduplica aumenta significativamente la capacità effettiva. Ciò vale per i dati memorizzati localmente e per i dati a più livelli nel cloud.
- Qualità del servizio (AQoS) minima, massima e adattativa. I controlli granulari della qualità del servizio (QoS) aiutano a mantenere i livelli di performance per le applicazioni critiche in ambienti altamente condivisi.
- NetApp FabricPool. Offre il tiering automatico dei dati cold per le opzioni di cloud storage pubblico e privato, tra cui Amazon Web Services (AWS), Azure e la soluzione di storage NetApp StorageGRID. Per ulteriori informazioni su FabricPool, vedere "TR-4598: Best practice FabricPool".

Accelera e proteggi i dati

ONTAP offre livelli superiori di performance e protezione dei dati ed estende queste funzionalità nei seguenti modi:

- Performance e latenza ridotta. ONTAP offre il throughput più elevato possibile con la latenza più bassa possibile.
- Protezione dei dati. ONTAP offre funzionalità di protezione dei dati integrate con gestione comune su tutte le piattaforme.
- NetApp Volume Encryption (NVE). ONTAP offre crittografia nativa a livello di volume con supporto per la gestione delle chiavi sia integrata che esterna.
- Multi-tenancy e autenticazione a più fattori. ONTAP consente la condivisione delle risorse dell'infrastruttura con i massimi livelli di sicurezza.

Infrastruttura a prova di futuro

ONTAP aiuta a soddisfare le esigenze di business esigenti e in continua evoluzione con le seguenti funzionalità:

- Scalabilità perfetta e operazioni senza interruzioni. ONTAP supporta l'aggiunta senza interruzioni di

capacità ai controller esistenti e ai cluster scale-out. I clienti possono eseguire l'upgrade alle tecnologie più recenti, come NVMe e 32GB FC, senza costose migrazioni dei dati o interruzioni.

- Connessione al cloud. ONTAP è il software per la gestione dello storage più connesso al cloud, con opzioni per storage software-defined e istanze native per il cloud in tutti i cloud pubblici.
- Integrazione con le applicazioni emergenti. ONTAP offre servizi dati di livello Enterprise per piattaforme e applicazioni di prossima generazione, come veicoli autonomi, città intelligenti e industria 4.0, utilizzando la stessa infrastruttura che supporta le applicazioni aziendali esistenti.

Amazon FSX per NetApp ONTAP

Amazon FSX per NetApp ONTAP è un servizio AWS completamente gestito e first-party che offre un file storage altamente affidabile, scalabile, dalle performance elevate e ricco di funzionalità, costruito sul popolare file system ONTAP di NetApp. FSX per ONTAP combina le funzionalità, le performance, le funzionalità e le operazioni API dei file system NetApp con l'agilità, la scalabilità e la semplicità di un servizio AWS completamente gestito.

NetApp Astra Trident

Astra Trident permette il consumo e la gestione delle risorse di storage in tutte le più apprezzate piattaforme di storage NetApp, nel cloud pubblico o on-premise, incluso ONTAP (AFF, FAS, Select, cloud, Amazon FSX per NetApp ONTAP), software Element (NetApp HCI, SolidFire), servizio Azure NetApp Files e Cloud Volumes Service su Google Cloud. Astra Trident è un orchestrator di storage dinamico conforme a Container Storage Interface (CSI) che si integra in modo nativo con Kubernetes.

Kubernetes

Kubernetes è una piattaforma open source, distribuita e di orchestrazione dei container, originariamente progettata da Google e ora gestita dalla Cloud Native Computing Foundation (CNCF). Kubernetes offre l'automazione delle funzioni di implementazione, gestione e scalabilità per le applicazioni in container ed è la piattaforma di orchestrazione dei container dominante negli ambienti Enterprise.

Amazon Elastic Kubernetes Service (EKS)

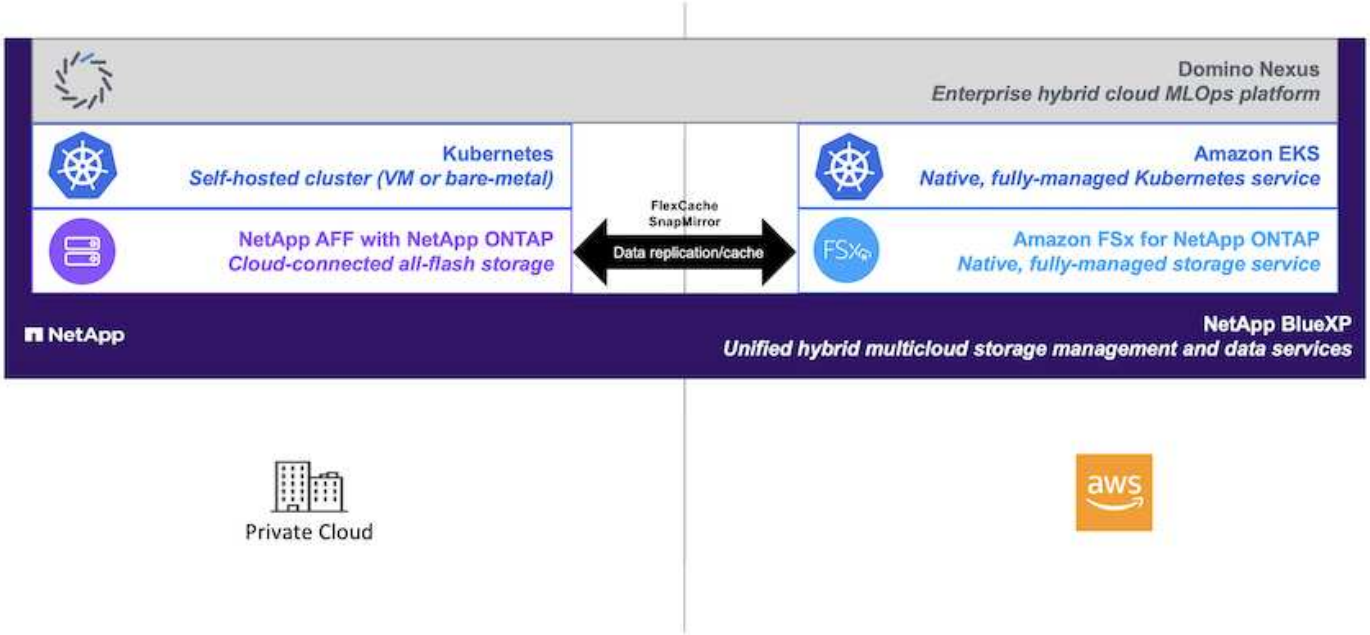
Amazon Elastic Kubernetes Service (Amazon EKS) è un servizio Kubernetes gestito nel cloud AWS. Amazon EKS gestisce automaticamente la disponibilità e la scalabilità dei nodi del piano di controllo di Kubernetes responsabili della pianificazione dei container, della gestione della disponibilità applicativa, della memorizzazione dei dati del cluster e di altre attività chiave. Con Amazon EKS, puoi sfruttare tutte le performance, la scalabilità, l'affidabilità e la disponibilità dell'infrastruttura AWS, oltre alle integrazioni con i servizi di rete e sicurezza AWS.

Architettura

Questa soluzione combina le funzionalità di pianificazione del carico di lavoro di multicloud ibrido di Domino Nexus con i servizi dati NetApp per creare una piattaforma MLOps del cloud ibrido unificata. Per ulteriori informazioni, consultare la tabella seguente.

Componente	Nome	Ambiente
Piano di controllo MLOps	"Piattaforma Domino Enterprise ai con Domino Nexus"	AWS

Componente	Nome	Ambiente
Ambienti di calcolo della piattaforma MLOps	"Domino Nexus Data Planes"	AWS, data center on-premise
Piattaforma di calcolo on-premise	"Kubernetes" con "NetApp Astra Trident"	Data center on-premise
Piattaforma di cloud computing	"Amazon Elastic Kubernetes Service (EKS)" con "NetApp Astra Trident"	AWS
Piattaforma per i dati on-premise	"Appliance di storage NetApp" basato su "NetApp ONTAP"	Data center on-premise
Piattaforma per i dati nel cloud	"Amazon FSX per NetApp ONTAP"	AWS



Setup iniziale

Questa sezione descrive le attività di configurazione iniziali che devono essere eseguite per utilizzare Domino Nexus con i servizi dati NetApp in un ambiente ibrido che incorpora un data center on-premise e AWS.

Prerequisiti

Prima di eseguire le operazioni descritte in questa sezione, si presuppone che siano già state eseguite le seguenti operazioni:

- Hai già implementato e configurato la tua piattaforma di storage NetApp ONTAP on-premise. Per ulteriori informazioni, consultare la "Documentazione sui prodotti NetApp".
- Hai già eseguito il provisioning di un'istanza di Amazon FSX per NetApp ONTAP in AWS. Per ulteriori informazioni, consultare la "Pagina del prodotto Amazon FSX per NetApp ONTAP".
- Hai già effettuato il provisioning di un cluster Kubernetes nel tuo data center on-premise. Per ulteriori informazioni, consultare la "Guida amministratore di Domino".

- Hai già effettuato il provisioning di un cluster Amazon EKS in AWS. Per ulteriori informazioni, consultare la "Guida amministratore di Domino".
- Hai installato NetApp Astra Trident nel tuo cluster Kubernetes on-premise. Inoltre, hai configurato questa istanza Trident per utilizzare la tua piattaforma di storage NetApp ONTAP on-premise durante il provisioning e la gestione delle risorse di storage. Per ulteriori informazioni, consultare la "Documentazione di NetApp Astra Trident".
- Hai installato NetApp Astra Trident nel tuo cluster Amazon EKS. Inoltre, questa istanza Trident è stata configurata per utilizzare la tua istanza di Amazon FSX per NetApp ONTAP durante il provisioning e la gestione delle risorse di storage. Per ulteriori informazioni, consultare la "Documentazione di NetApp Astra Trident".
- Devi disporre di connettività di rete bidirezionale tra il data center on-premise e il cloud privato virtuale (VPC) in AWS. Per ulteriori informazioni sulle varie opzioni di implementazione, fare riferimento alla "Documentazione di Amazon Virtual Private Network (VPN)".

Installare Domino Enterprise ai Platform in AWS

Per installare Domino Enterprise MLOps Platform in AWS, seguire le istruzioni riportate in "Guida amministratore di Domino". Devi implementare Domino nello stesso cluster Amazon EKS di cui hai precedentemente eseguito il provisioning. Inoltre, NetApp Astra Trident deve essere già installato e configurato in questo cluster EKS, ed è necessario specificare una classe di storage gestita da Trident come classe di storage condiviso nel file di configurazione dell'installazione di domino.yml.

> Fare riferimento a. "Guida di riferimento per l'installazione di Domino" per informazioni dettagliate su come specificare una classe di archiviazione condivisa nel file di configurazione di installazione domino.yml.

> "Report tecnico TR-4952" Guida l'implementazione di Domino in AWS con Amazon FSX per NetApp ONTAP e può essere un utile riferimento per la risoluzione di eventuali problemi che si verificano.

Attivare Domino Nexus

Successivamente, è necessario attivare Domino Nexus. Fare riferimento a. "Guida amministratore di Domino" per ulteriori informazioni.

Implementare un piano dati Domino nel data center locale

Successivamente, è necessario implementare un piano dati Domino nel data center locale. Devi implementare questo piano dati nel cluster Kubernetes on-premise che hai precedentemente fornito il provisioning. Inoltre, NetApp Astra Trident deve essere già installato e configurato in questo cluster Kubernetes. Fare riferimento a. "Guida amministratore di Domino" per ulteriori informazioni.

Esporre i volumi NetApp esistenti a Domino

In questa sezione vengono descritte le attività da eseguire per esporre i volumi NFS NetApp ONTAP esistenti alla piattaforma Domino MLOps. Questi stessi passaggi si applicano sia on-premise che in AWS.

Perché esporre NetApp ONTAP Volumes a Domino?

L'utilizzo di NetApp Volumes insieme a Domino offre i seguenti vantaggi:

- Puoi eseguire carichi di lavoro su set di dati estremamente grandi sfruttando le funzionalità scale-out di NetApp ONTAP.
- Puoi eseguire carichi di lavoro su diversi nodi di calcolo senza dover copiare i dati nei singoli nodi.
- Puoi sfruttare le funzionalità di spostamento e sincronizzazione dei dati del multicloud ibrido di NetApp per accedere ai dati attraverso più data center e/o cloud.
- Vuoi creare in modo rapido e semplice una cache dei dati in un data center o un cloud diverso.

Esponi volumi NFS esistenti che non sono stati sottoposti a provisioning da Astra Trident

Se il tuo volume NFS NetApp ONTAP esistente non è stato sottoposto a provisioning da Astra Trident, segui i passaggi descritti in questa sottosezione.

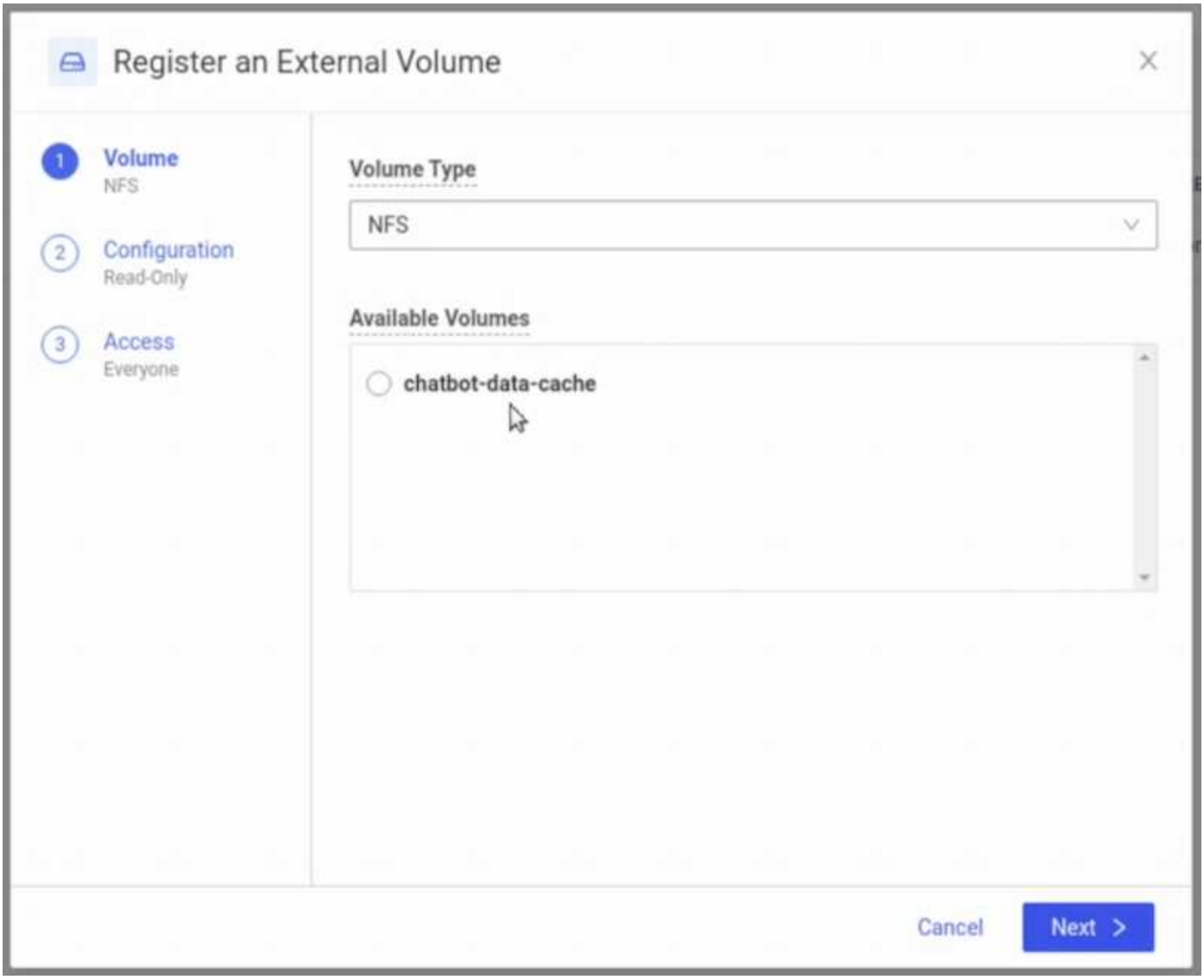
Crea PV e PVC in Kubernetes

> Per i volumi on-premise, creare PV e PVC nel cluster Kubernetes on-premise. Per Amazon FSX per i volumi NetApp ONTAP, crea il PV e il PVC in Amazon EKS.

Innanzitutto, devi creare un volume persistente (PV) e una dichiarazione di volume persistente (PVC) nel cluster Kubernetes. Per creare PV e PVC, utilizzare il "ESEMPIO DI NFS PV/PVC" Dalla guida dell'amministratore di Domino e aggiornare i valori in modo che riflettano l'ambiente in uso. Assicurarsi di specificare i valori corretti per `namespace`, `nfs.path`, e. `nfs.server` campi. Inoltre, consigliamo di assegnare nomi univoci ai PV e PVC che rappresentino la natura dei dati memorizzati nel volume NFS ONTAP corrispondente. Ad esempio, se il volume contiene immagini di difetti di fabbricazione, è possibile denominare PV, `pv-mfg-defect-images`, E il PVC, `pvc-mfg-defect-images`.

Registra volume dati esterni in Domino

Successivamente, è necessario registrare un volume di dati esterno in Domino. Per registrare un volume di dati esterno, fare riferimento alla "istruzioni" Nella guida dell'amministratore di Domino. Quando si registra il volume, assicurarsi di selezionare "NFS" dal menu a discesa "tipo volume". Dopo aver selezionato "NFS", dovresti vedere il tuo PVC nell'elenco "Available Volumes".



Esponi i volumi esistenti che erano stati sottoposti a provisioning da Astra Trident

Se il tuo volume esistente è stato sottoposto a provisioning da Astra Trident, segui i passaggi descritti in questa sottosezione.

Modifica PVC esistente

Se il provisioning del tuo volume è stato eseguito da Astra Trident, disponi già di una richiesta persistente per il volume (PVC) corrispondente al tuo volume. Per esporre questo volume a Domino, è necessario modificare il PVC e aggiungere la seguente etichetta all'elenco delle etichette nella `metadata.labels` campo:

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

Registra volume dati esterni in Domino

Successivamente, è necessario registrare un volume di dati esterno in Domino. Per registrare un volume di dati esterno, fare riferimento alla "istruzioni" Nella guida dell'amministratore di Domino. Quando si registra il volume, assicurarsi di selezionare "Generico" dal menu a discesa "tipo volume". Dopo aver selezionato "Generico", il PVC dovrebbe essere visualizzato nell'elenco "volumi disponibili".

Accedi agli stessi dati in ambienti diversi

Questa sezione descrive le attività che devono essere eseguite per accedere agli stessi dati in diversi ambienti di elaborazione. Nella piattaforma Domino MLOps, gli ambienti di calcolo sono denominati "piani dati". Segui le attività descritte in questa sezione se i tuoi dati risiedono su un volume NetApp in un piano dati ma devi accedervi in un altro piano dati. Questo tipo di scenario viene spesso chiamato "bursting" o, quando l'ambiente di destinazione è il cloud, "cloud bursting". Questa funzionalità è spesso necessaria quando si gestiscono risorse di calcolo limitate o sovrascritte. Ad esempio, se il tuo cluster di calcolo on-premise è sottoposto a un'iscrizione eccessiva, potresti voler programmare i workload nel cloud, dove possono essere avviati immediatamente.

Esistono due opzioni consigliate per l'accesso a un volume NetApp che si trova in un piano dati diverso. Queste opzioni sono illustrate nelle sottosezioni riportate di seguito. Scegliere una di queste opzioni in base alle proprie esigenze specifiche. I vantaggi e gli svantaggi delle due opzioni sono descritti nella tabella seguente.

Opzione	Benefici	Svantaggi
Opzione 1 - cache	- Flusso di lavoro più semplice - Possibilità di memorizzare nella cache un sottoinsieme di dati in base alle esigenze - Capacità di scrivere i dati di nuovo alla fonte - Nessuna copia remota da gestire	- Maggiore latenza all'accesso iniziale ai dati quando la cache viene idratata.
Opzione 2 - specchietto	- Copia completa del volume di origine - Nessuna latenza aumentata a causa dell'idratazione della cache (dopo il completamento dell'operazione mirror)	- Prima di accedere ai dati, è necessario attendere il completamento del funzionamento del mirror - Deve gestire una copia remota - Nessuna capacità di scrivere di nuovo alla fonte

Opzione 1 - creare una cache di un volume che risiede in un piano dati diverso

Con "Tecnologia NetApp FlexCache", È possibile creare una cache di un volume NetApp che risiede in un piano dati diverso. Ad esempio, se hai un volume NetApp nel tuo piano dati on-premise e hai bisogno di accedere a quel volume nel tuo piano dati AWS, puoi creare una cache del volume in AWS. In questa sezione vengono descritte le attività che è necessario eseguire per creare una cache di un volume NetApp che risiede in un piano dati diverso.

Creare un volume FlexCache nell'ambiente di destinazione

> Se l'ambiente di destinazione è il tuo data center on-premise, creerai il volume FlexCache sul tuo sistema ONTAP on-premise. Se l'ambiente di destinazione è AWS, creerai il volume FlexCache sull'istanza di Amazon FSX per NetApp ONTAP.

Innanzitutto, è necessario creare un volume FlexCache nell'ambiente di destinazione.

Ti consigliamo di utilizzare BlueXP per creare il volume FlexCache. Per creare un volume FlexCache con

BlueXP, segui le istruzioni riportate nella "Documentazione di caching dei volumi di BlueXP".

Se preferisci non utilizzare BlueXP, puoi usare ONTAP System Manager o l'interfaccia a riga di comando di ONTAP per creare il volume FlexCache. Per creare un volume FlexCache con System Manager, fare riferimento alle istruzioni riportate nella "Documentazione ONTAP". Per creare un volume FlexCache con l'interfaccia a riga di comando di ONTAP, fare riferimento alle istruzioni nella "Documentazione ONTAP".

Se si desidera automatizzare questo processo, è possibile utilizzare "API BlueXP", il "API REST di ONTAP"o il "Raccolta Ansible ONTAP".

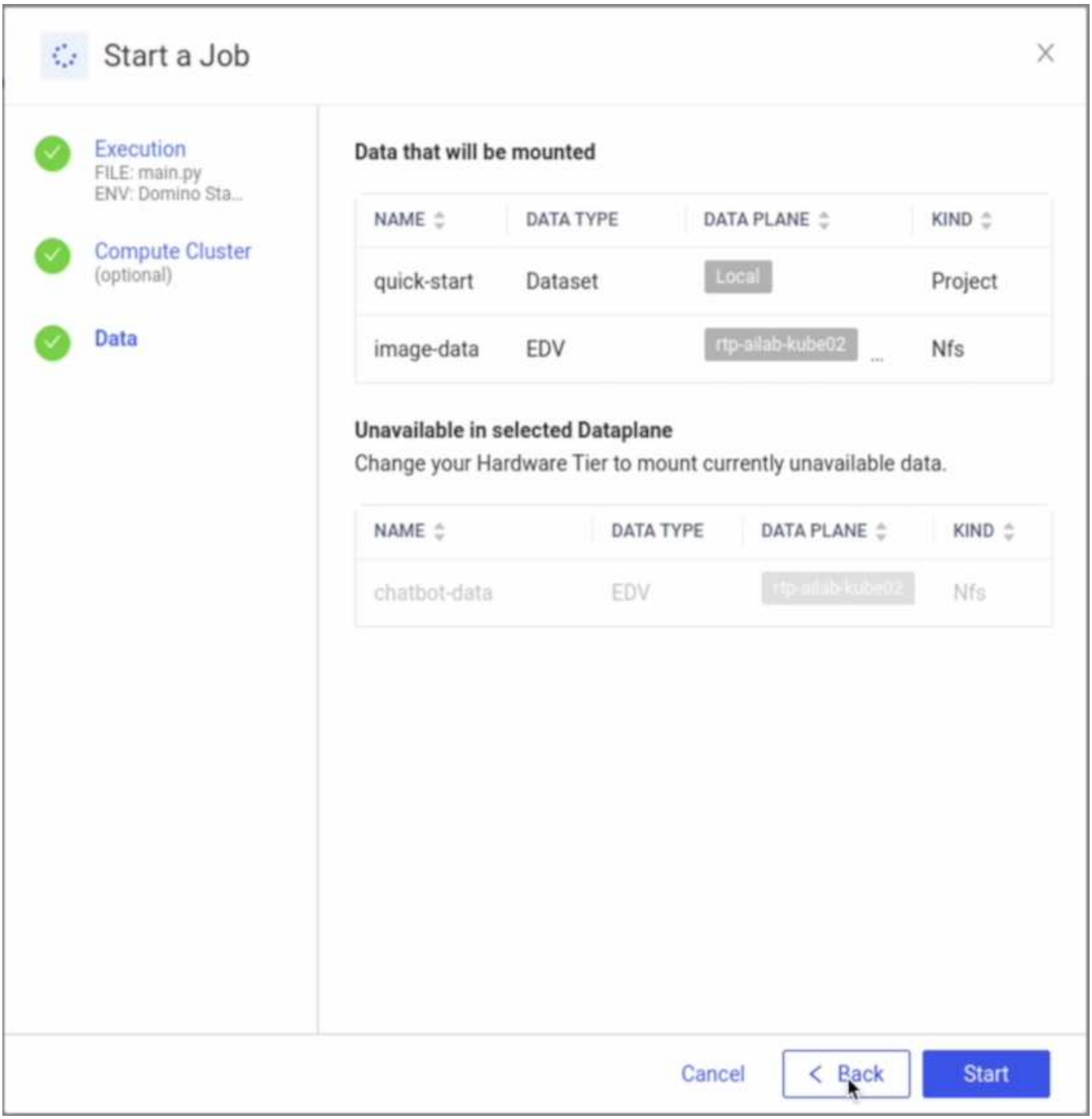
System Manager non è disponibile in Amazon FSX per NetApp ONTAP.

Esporre il volume FlexCache a Domino

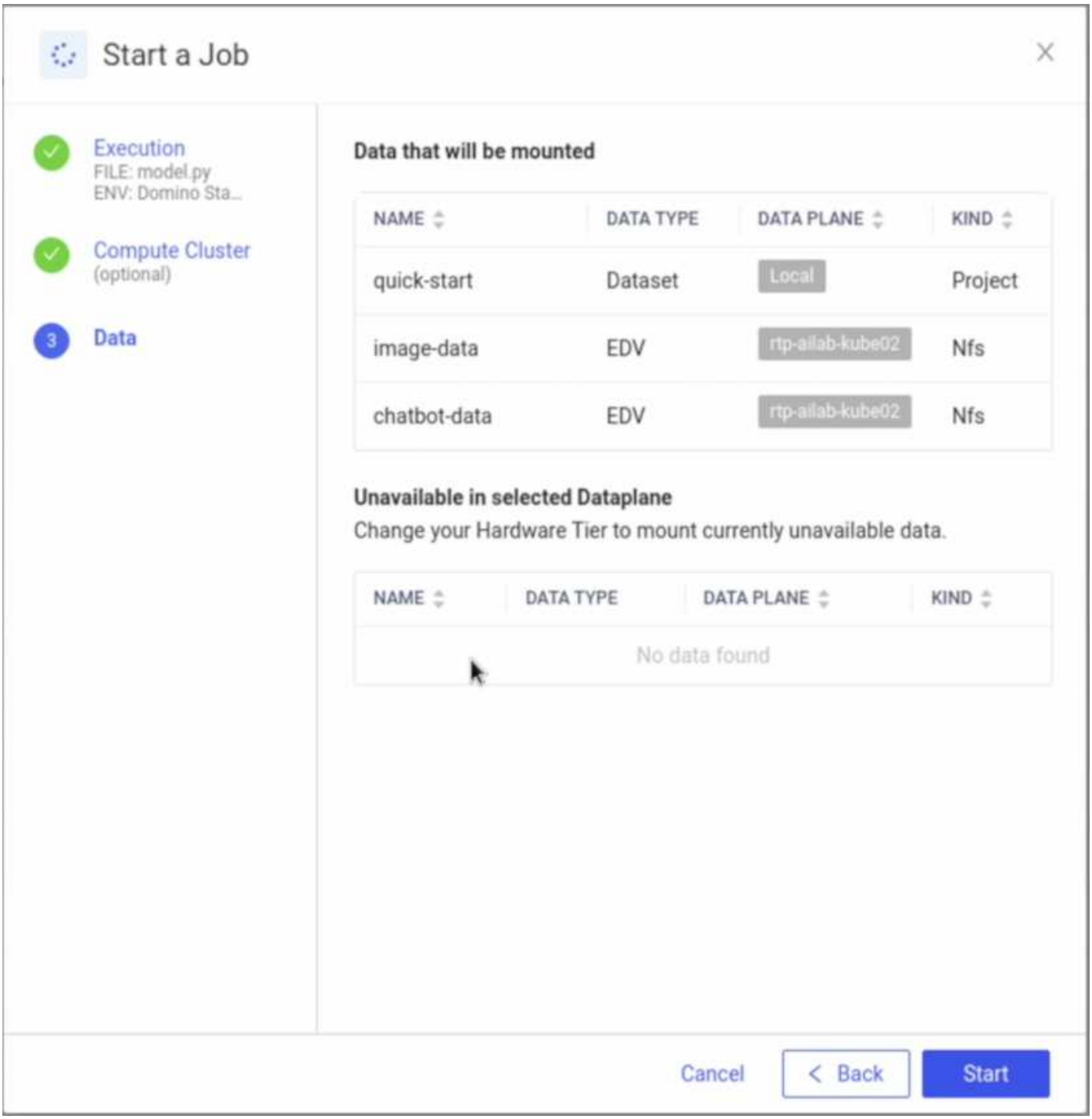
Successivamente, è necessario esporre il volume FlexCache alla piattaforma Domino MLOps. Per esporre il volume FlexCache a Domino, segui le istruzioni riportate nella sottosezione "Exposure existing NFS Volumes that non sono stati forniti da Astra Trident" di "Sezione "esponi volumi NetApp esistenti a Domino"" di questa soluzione.

A questo punto, sarà possibile montare il volume FlexCache quando si avviano processi e aree di lavoro nel piano dati di destinazione, come illustrato nelle seguenti schermate.

Prima di creare un volume FlexCache



Dopo l’esposizione del volume FlexCache a Domino



Opzione 2 - replicare un volume che risiede in un piano dati diverso

Con "Tecnologia di replica dei dati NetApp SnapMirror", È possibile creare una copia di un volume NetApp che risiede in un piano dati diverso. Ad esempio, se hai un volume NetApp nel tuo piano dati on-premise e hai bisogno di accedere a quel volume nel tuo piano dati AWS, puoi creare una copia del volume in AWS. In questa sezione vengono descritte le attività da eseguire per creare una copia di un volume NetApp che risiede in un piano dati diverso.

Creare una relazione SnapMirror

Innanzitutto, occorre creare una relazione SnapMirror tra il volume di origine e un nuovo volume di destinazione nell'ambiente di destinazione. Il volume di destinazione verrà creato come parte del processo di creazione della relazione SnapMirror.

Ti consigliamo di usare BlueXP per creare la relazione di SnapMirror. Per creare una relazione di SnapMirror con BlueXP, segui le istruzioni nella "Documentazione sulla replica BlueXP".

Se preferisci non utilizzare BlueXP, puoi usare ONTAP System Manager o l'interfaccia a riga di comando di ONTAP per creare la relazione di SnapMirror. Per creare una relazione di SnapMirror con System Manager, fare riferimento alle istruzioni nella "Documentazione ONTAP". Per creare una relazione di SnapMirror con l'interfaccia della riga di comando di ONTAP, fare riferimento alle istruzioni nella "Documentazione ONTAP".

Se si desidera automatizzare questo processo, è possibile utilizzare "API BlueXP", il "API REST di ONTAP"o il "Raccolta Ansible ONTAP".

System Manager non è disponibile in Amazon FSX per NetApp ONTAP.

Interrompere la relazione di SnapMirror

Successivamente, occorre interrompere la relazione di SnapMirror per attivare il volume di destinazione per l'accesso ai dati. Attendere il completamento della replica iniziale prima di eseguire questa operazione.

Puoi determinare se la replica è completa o meno controllando lo stato del mirror in BlueXP, ONTAP System Manager o nell'interfaccia a riga di comando di ONTAP. Al termine della replica, lo stato del mirror sarà "snapmirrored".

Consigliamo di utilizzare BlueXP per interrompere la relazione di SnapMirror. Per interrompere un rapporto di SnapMirror con BlueXP, segui le istruzioni riportate nella "Documentazione sulla replica BlueXP".

Se preferisci non utilizzare BlueXP, puoi usare ONTAP System Manager o l'interfaccia a riga di comando di ONTAP per interrompere la relazione di SnapMirror. Per interrompere una relazione di SnapMirror con System Manager, fare riferimento alle istruzioni nella "Documentazione ONTAP". Per interrompere una relazione di SnapMirror con la CLI ONTAP, fare riferimento alle istruzioni riportate nella "Documentazione ONTAP".

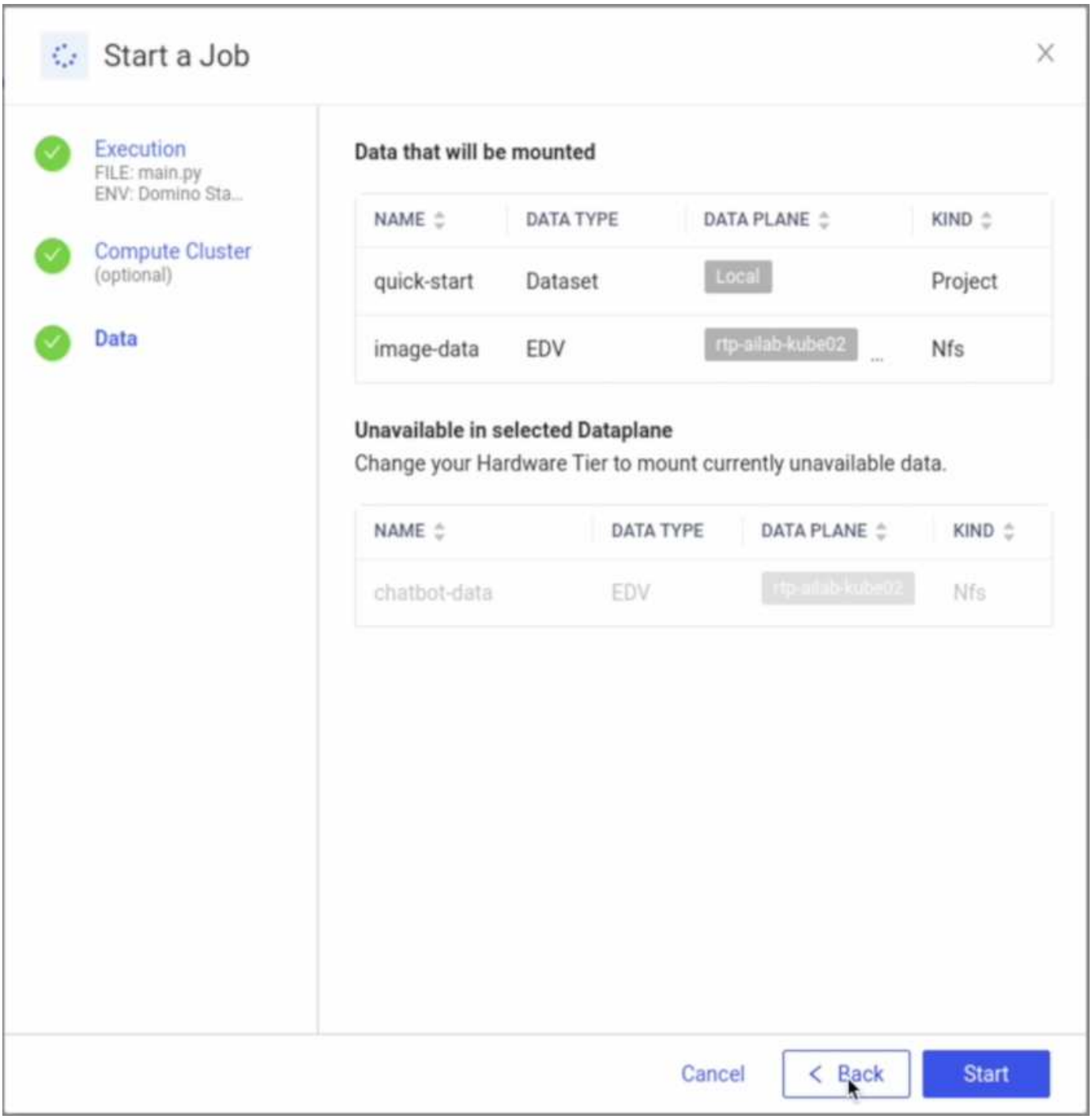
Se si desidera automatizzare questo processo, è possibile utilizzare "API BlueXP", il "API REST di ONTAP"o il "Raccolta Ansible ONTAP".

Esporre il volume di destinazione a Domino

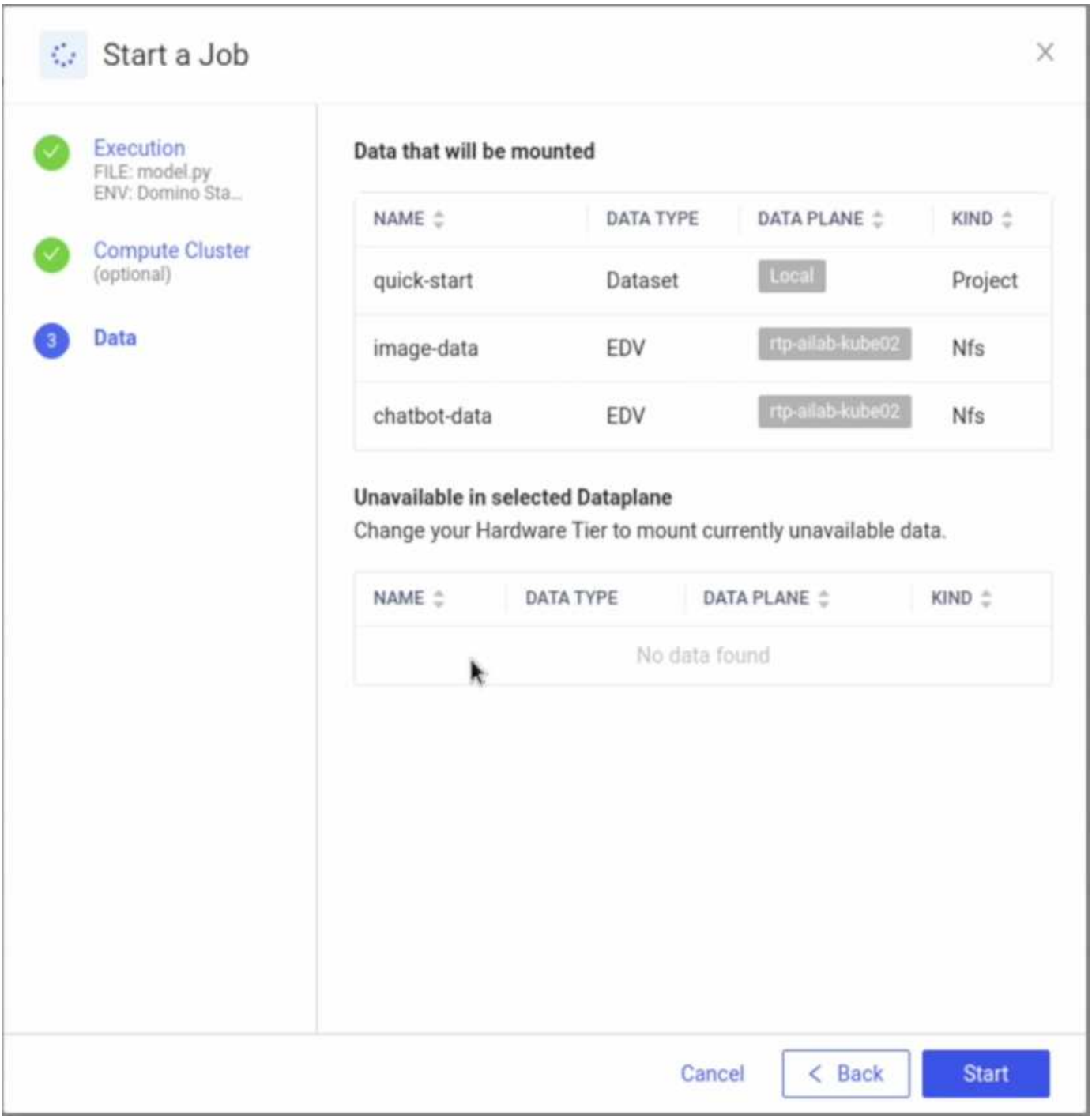
Successivamente, è necessario esporre il volume di destinazione alla piattaforma Domino MLOps. Per esporre il volume di destinazione a Domino, seguire le istruzioni riportate nella sottosezione "Expose Exposure Existing NFS Volumes that non Were Provisioning da Astra Trident" di "Sezione "esponi volumi NetApp esistenti a Domino"" di questa soluzione.

A questo punto, sarà possibile montare il volume di destinazione quando si avviano processi e aree di lavoro nel piano dati di destinazione, come illustrato nelle seguenti schermate.

Prima di creare una relazione SnapMirror



Dopo l’esposizione del volume di destinazione a Domino



Dove trovare ulteriori informazioni

Per ulteriori informazioni sulle informazioni descritte in questo documento, consultare i seguenti documenti e/o siti Web:

- Domino Data Lab

  "https://domino.ai"

- Domino Nexus

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

- NetApp BlueXP

  "https://bluexp.netapp.com"

- Software per la gestione dei dati NetApp ONTAP

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

- Soluzioni NetApp ai

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

Ringraziamenti

- Josh Mineroff, Director of SA for Tech Alliances, Domino Data Lab
- Nicholas Jablonski, Field CTO, Domino Data Lab
- Prabu Arjunan, Solution Architect presso NetApp
- Brian Young, Global Alliance Director, Technology Alliance Partners, NetApp

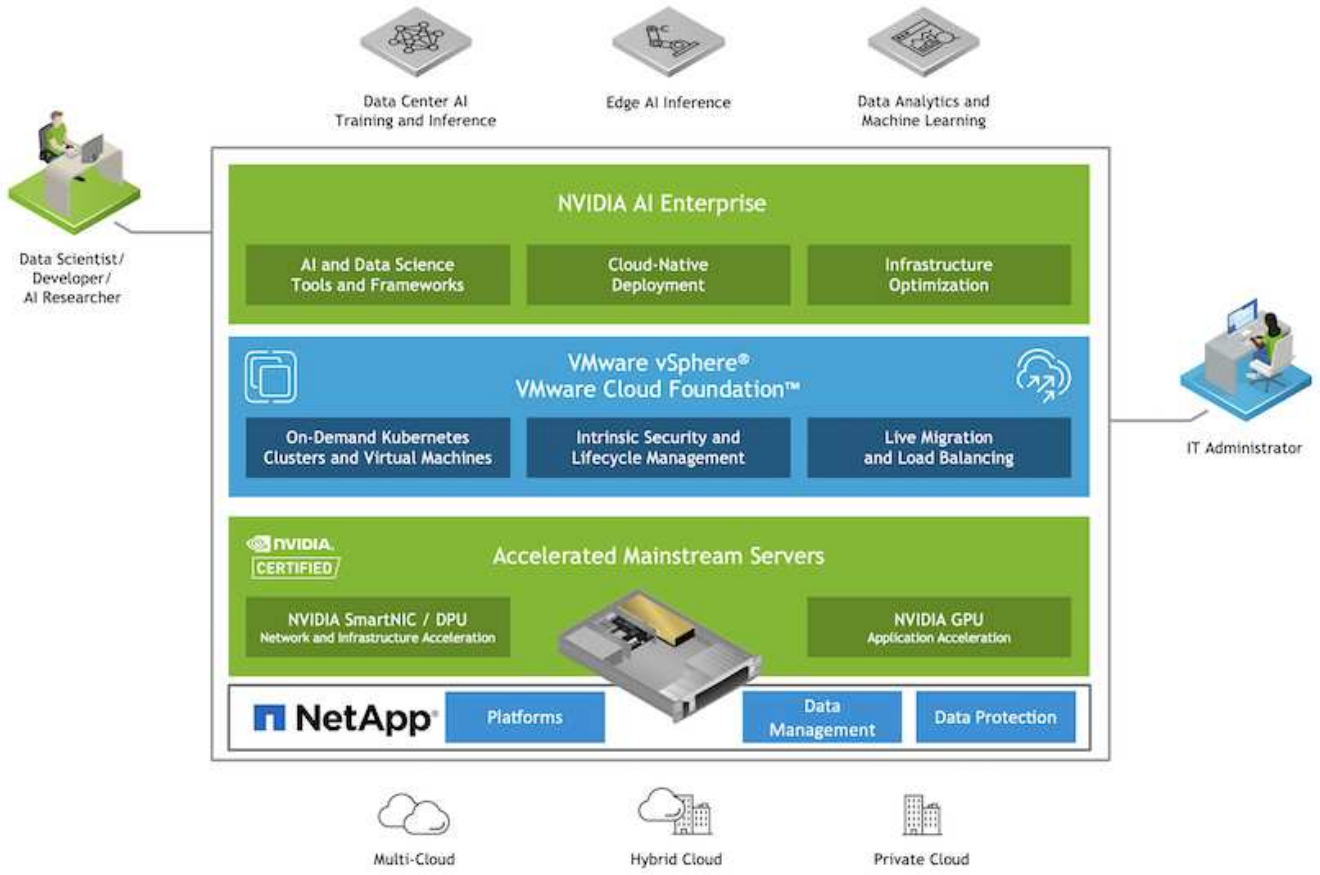
NVIDIA ai Enterprise con NetApp e VMware

NVIDIA ai Enterprise con NetApp e VMware

Mike Oglesby, NetApp

Per gli architetti E gli amministratori IT, gli strumenti di ai possono essere complicati e poco familiari. Inoltre, molte piattaforme ai non sono Enterprise-ready. NVIDIA ai Enterprise, basata su NetApp e VMware, è stata creata per offrire un'architettura ai di livello Enterprise ottimizzata.

NVIDIA ai Enterprise è una suite end-to-end nativa del cloud di software di ai e data analytics ottimizzato, certificato e supportato da NVIDIA per l'esecuzione su VMware vSphere con sistemi certificati NVIDIA. Questo software facilita l'implementazione, la gestione e la scalabilità semplici e rapide dei carichi di lavoro ai nel moderno ambiente di cloud ibrido. NVIDIA ai Enterprise, basata su NetApp e VMware, offre gestione dei dati e dei workload ai di livello Enterprise in un pacchetto semplice e familiare.



Panoramica sulla tecnologia

NVIDIA ai Enterprise

NVIDIA ai Enterprise è una suite end-to-end nativa del cloud di software di ai e data analytics ottimizzato, certificato e supportato da NVIDIA per l'esecuzione su VMware vSphere con sistemi certificati NVIDIA. Questo software facilita l'implementazione, la gestione e la scalabilità semplici e rapide dei carichi di lavoro ai nel moderno ambiente di cloud ibrido.

NVIDIA GPU CLOUD (NGC)

NVIDIA NGC ospita un catalogo di software ottimizzato per GPU per i professionisti dell'ai per sviluppare le proprie soluzioni di ai. Fornisce inoltre accesso a vari servizi di ai, tra cui NVIDIA base Command per la formazione sui modelli, NVIDIA Fleet Command per l'implementazione e il monitoraggio dei modelli e NGC Private Registry per l'accesso e la gestione sicuri del software proprietario di ai. Inoltre, i clienti di NVIDIA ai Enterprise possono richiedere il supporto tramite il portale NGC.

VMware vSphere

VMware vSphere è la piattaforma di virtualizzazione di VMware, che trasforma i data center in infrastrutture di calcolo aggregate che includono CPU, storage e risorse di rete. VSphere gestisce queste infrastrutture come un ambiente operativo unificato e fornisce agli amministratori gli strumenti per gestire i data center che partecipano a tale ambiente.

I due componenti principali di vSphere sono ESXi e vCenter Server. ESXi è la piattaforma di virtualizzazione in cui gli amministratori creano ed eseguono macchine virtuali e appliance virtuali. VCenter Server è il servizio

attraverso il quale gli amministratori gestiscono più host connessi in una rete e pool di risorse host.

NetApp ONTAP

ONTAP 9, l'ultima generazione di software per la gestione dello storage NetApp, consente alle aziende di modernizzare l'infrastruttura e passare a un data center predisposto per il cloud. Sfruttando le funzionalità di gestione dei dati leader del settore, ONTAP consente la gestione e la protezione dei dati con un singolo set di strumenti, indipendentemente dalla posizione dei dati. Puoi anche spostare liberamente i dati ovunque siano necessari: Edge, core o cloud. ONTAP 9 include numerose funzionalità che semplificano la gestione dei dati, accelerano e proteggono i dati critici e abilitano le funzionalità dell'infrastruttura di nuova generazione nelle architetture di cloud ibrido.

Semplifica la gestione dei dati

La gestione dei dati è fondamentale per le operazioni IT aziendali e per i data scientist, in modo che le risorse appropriate vengano utilizzate per le applicazioni ai e per la formazione dei set di dati ai/ML. Le seguenti informazioni aggiuntive sulle tecnologie NetApp non rientrano nell'ambito di questa convalida, ma potrebbero essere rilevanti a seconda dell'implementazione.

Il software per la gestione dei dati ONTAP include le seguenti funzionalità per ottimizzare e semplificare le operazioni e ridurre il costo totale delle operazioni:

- Compaction dei dati inline e deduplica estesa. La compattazione dei dati riduce lo spazio sprecato all'interno dei blocchi di storage e la deduplica aumenta significativamente la capacità effettiva. Ciò vale per i dati memorizzati localmente e per i dati a più livelli nel cloud.
- Qualità del servizio (AQoS) minima, massima e adattativa. I controlli granulari della qualità del servizio (QoS) aiutano a mantenere i livelli di performance per le applicazioni critiche in ambienti altamente condivisi.
- NetApp FabricPool. Offre il tiering automatico dei dati cold per le opzioni di cloud storage pubblico e privato, tra cui Amazon Web Services (AWS), Azure e la soluzione di storage NetApp StorageGRID. Per ulteriori informazioni su FabricPool, vedere "TR-4598: Best practice FabricPool".

Accelera e proteggi i dati

ONTAP offre livelli superiori di performance e protezione dei dati ed estende queste funzionalità nei seguenti modi:

- Performance e latenza ridotta. ONTAP offre il throughput più elevato possibile con la latenza più bassa possibile.
- Protezione dei dati. ONTAP offre funzionalità di protezione dei dati integrate con gestione comune su tutte le piattaforme.
- NetApp Volume Encryption (NVE). ONTAP offre crittografia nativa a livello di volume con supporto per la gestione delle chiavi sia integrata che esterna.
- Multi-tenancy e autenticazione a più fattori. ONTAP consente la condivisione delle risorse dell'infrastruttura con i massimi livelli di sicurezza.

Infrastruttura a prova di futuro

ONTAP aiuta a soddisfare le esigenze di business esigenti e in continua evoluzione con le seguenti funzionalità:

- Scalabilità perfetta e operazioni senza interruzioni. ONTAP supporta l'aggiunta senza interruzioni di capacità ai controller esistenti e ai cluster scale-out. I clienti possono eseguire l'upgrade alle tecnologie più

recenti, come NVMe e 32GB FC, senza costose migrazioni dei dati o interruzioni.

- Connessione al cloud. ONTAP è il software per la gestione dello storage più connesso al cloud, con opzioni per lo storage software-defined (ONTAP Select) e le istanze native del cloud (NetApp Cloud Volumes Service) in tutti i cloud pubblici.
- Integrazione con le applicazioni emergenti. ONTAP offre servizi dati di livello Enterprise per piattaforme e applicazioni di prossima generazione, come veicoli autonomi, città intelligenti e industria 4.0, utilizzando la stessa infrastruttura che supporta le applicazioni aziendali esistenti.

NetApp DataOps Toolkit

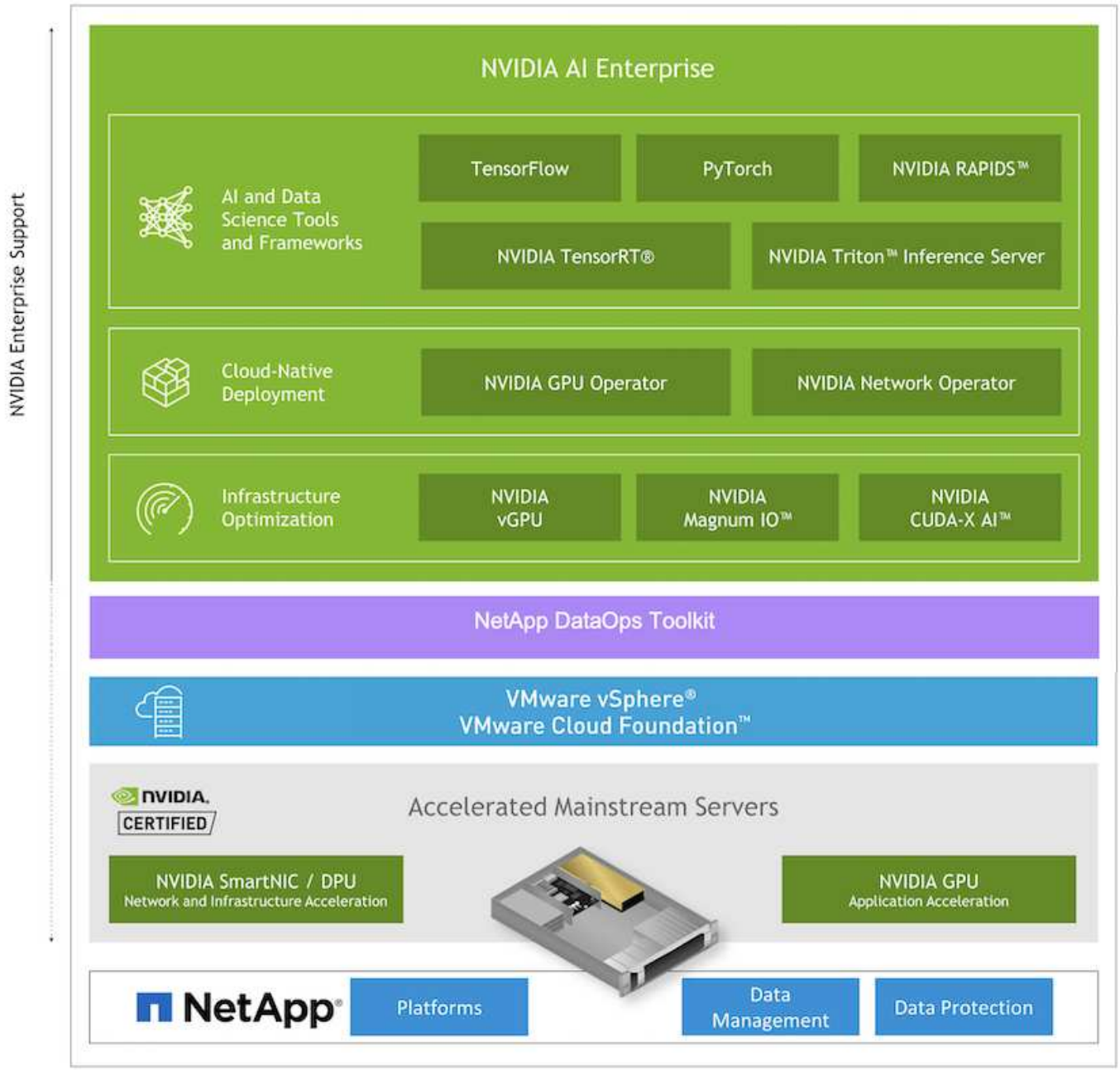
Il NetApp DataOps Toolkit è uno strumento basato su Python che semplifica la gestione degli spazi di lavoro di sviluppo/formazione e dei server di inferenza supportati dallo storage NetApp scale-out dalle performance elevate. Le funzionalità principali includono:

- Provisioning rapido di nuove aree di lavoro JupyterLab ad alta capacità supportate da storage NetApp scale-out dalle performance elevate.
- Provisioning rapido delle nuove istanze di NVIDIA Triton Inference Server supportate dallo storage NetApp di livello Enterprise.
- Clonare quasi instantaneamente le aree di lavoro JupyterLab ad alta capacità per consentire la sperimentazione o l'iterazione rapida.
- Salvataggio quasi istantaneo di snapshot di aree di lavoro JupyterLab ad alta capacità per backup e/o tracciabilità/baselining.
- Provisioning, cloning e snapshot near-instataneamente di volumi di dati ad alta capacità e performance elevate.

Architettura

Questa soluzione si basa su un'architettura collaudata e familiare con sistemi certificati NetApp, VMware e NVIDIA. Per ulteriori informazioni, consultare la tabella seguente.

Componente	Dettagli
Software ai e Data Analytics	"NVIDIA ai Enterprise per VMware"
Piattaforma di virtualizzazione	"VMware vSphere"
Piattaforma di calcolo	"Sistemi certificati NVIDIA"
Piattaforma per la gestione dei dati	"NetApp ONTAP"



Setup iniziale

In questa sezione vengono descritte le attività di configurazione iniziali che devono essere eseguite per utilizzare NVIDIA ai Enterprise con NetApp e VMware.

Prerequisiti

Prima di eseguire i passaggi descritti in questa sezione, si presuppone che siano già state implementate VMware vSphere e NetApp ONTAP. Fare riferimento a. "Matrice di supporto dei prodotti NVIDIA ai Enterprise" Per ulteriori informazioni sulle versioni di vSphere supportate. Fare riferimento a. "Documentazione sulle soluzioni NetApp e VMware" Per informazioni dettagliate sull'implementazione di VMware vSphere con NetApp ONTAP.

Installare il software host NVIDIA ai Enterprise

Per installare il software host NVIDIA ai EnEnterprise, seguire le istruzioni riportate nelle sezioni 1-4 del "NVIDIA ai Enterprise Quick Start Guide".

Utilizzare il software NVIDIA NGC

Questa sezione descrive le attività da eseguire per utilizzare il software NVIDIA NGC Enterprise in un ambiente NVIDIA ai Enterprise.

Setup (Configurazione)

In questa sezione vengono descritte le operazioni di configurazione iniziali da eseguire per utilizzare il software NVIDIA NGC Enterprise in un ambiente NVIDIA ai Enterprise.

Prerequisiti

Prima di eseguire i passaggi descritti in questa sezione, si presuppone che il software host NVIDIA ai EnEnterprise sia già stato implementato seguendo le istruzioni riportate nella "Setup iniziale" pagina.

Creare una VM ospite Ubuntu con vGPU

Innanzitutto, è necessario creare una macchina virtuale guest Ubuntu 20.04 con vGPU. Per creare una macchina virtuale guest Ubuntu 20.04 con vGPU, seguire le istruzioni riportate nella "Guida all'implementazione di NVIDIA ai Enterprise".

Scaricare e installare il software NVIDIA Guest

Quindi, è necessario installare il software guest NVIDIA richiesto nella macchina virtuale guest creata al passaggio precedente. Per scaricare e installare il software guest NVIDIA richiesto nella macchina virtuale guest, seguire le istruzioni riportate nelle sezioni 5.1-5.4 della "NVIDIA ai Enterprise Quick Start Guide".

> Quando si eseguono le attività di verifica descritte nella sezione 5.4, potrebbe essere necessario utilizzare un tag di versione dell'immagine contenitore CUDA diverso poiché l'immagine contenitore CUDA è stata aggiornata dopo la scrittura della guida. Nella nostra convalida, abbiamo utilizzato "nvidia/cuda:11.0.3-base-ubuntu20.04".

Scarica i container ai/Analytics Framework

Quindi, devi scaricare le immagini container di ai o del framework di analisi necessarie da NVIDIA NGC in modo che siano disponibili all'interno della tua macchina virtuale guest. Per scaricare i container del framework all'interno della macchina virtuale guest, seguire le istruzioni riportate nella "Guida all'implementazione di NVIDIA ai Enterprise".

Installare e configurare il NetApp DataOps Toolkit

Quindi, è necessario installare il NetApp DataOps Toolkit per ambienti tradizionali all'interno della macchina virtuale guest. Il toolkit NetApp DataOps può essere utilizzato per gestire volumi di dati scale-out sul sistema ONTAP direttamente dal terminale all'interno della macchina virtuale guest. Per installare il NetApp DataOps Toolkit nella macchina virtuale guest, eseguire le seguenti operazioni.

1. Installare il pip.

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

2. Disconnettersi dal terminale della macchina virtuale guest e quindi effettuare nuovamente l'accesso.
3. Configurare il NetApp DataOps Toolkit. Per completare questo passaggio, sono necessari i dettagli di accesso API per il sistema ONTAP. Potrebbe essere necessario ottenerli dall'amministratore dello storage.

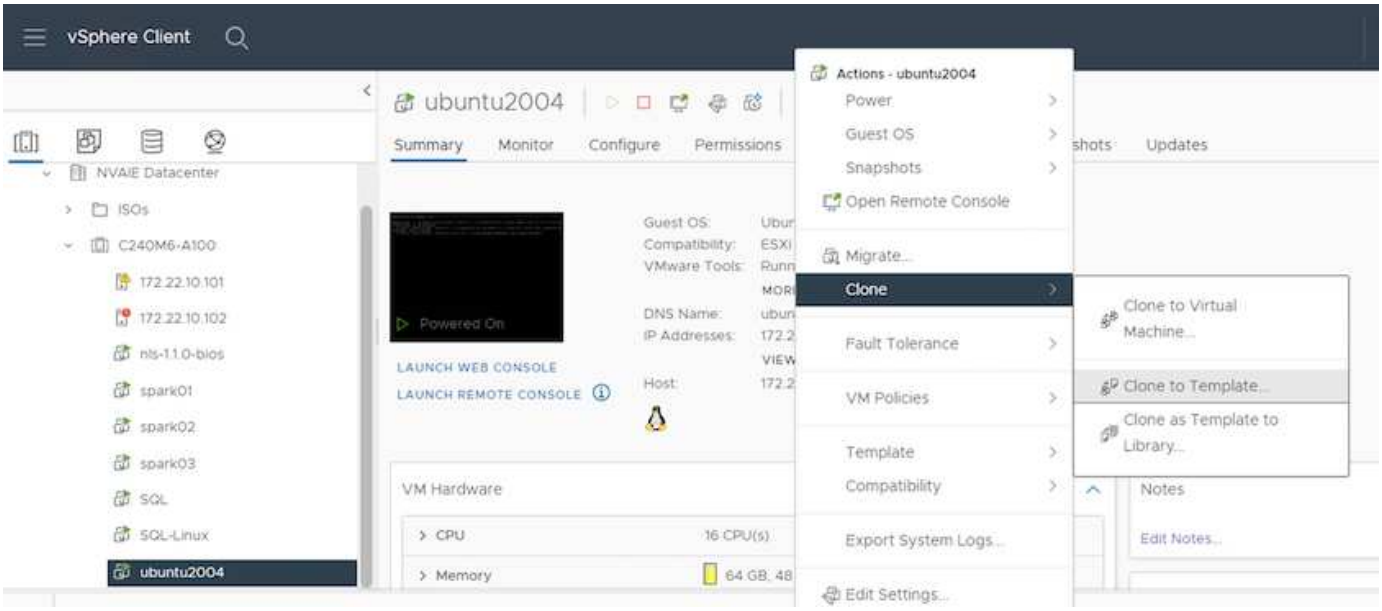
```
$ netapp_dataops_cli.py config

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'.
```

Creare un modello di macchina virtuale guest

Infine, è necessario creare un modello di macchina virtuale basato sulla macchina virtuale guest. Sarà possibile utilizzare questo modello per creare rapidamente macchine virtuali guest per l'utilizzo del software NVIDIA NGC.

Per creare un modello di macchina virtuale in base alla macchina virtuale guest, accedere a VMware vSphere, fare clic sul nome della macchina virtuale guest, scegliere "Clone", "Clone to Template…", quindi seguire la

procedura guidata.



Esempio di caso d'utilizzo - lavoro di training TensorFlow

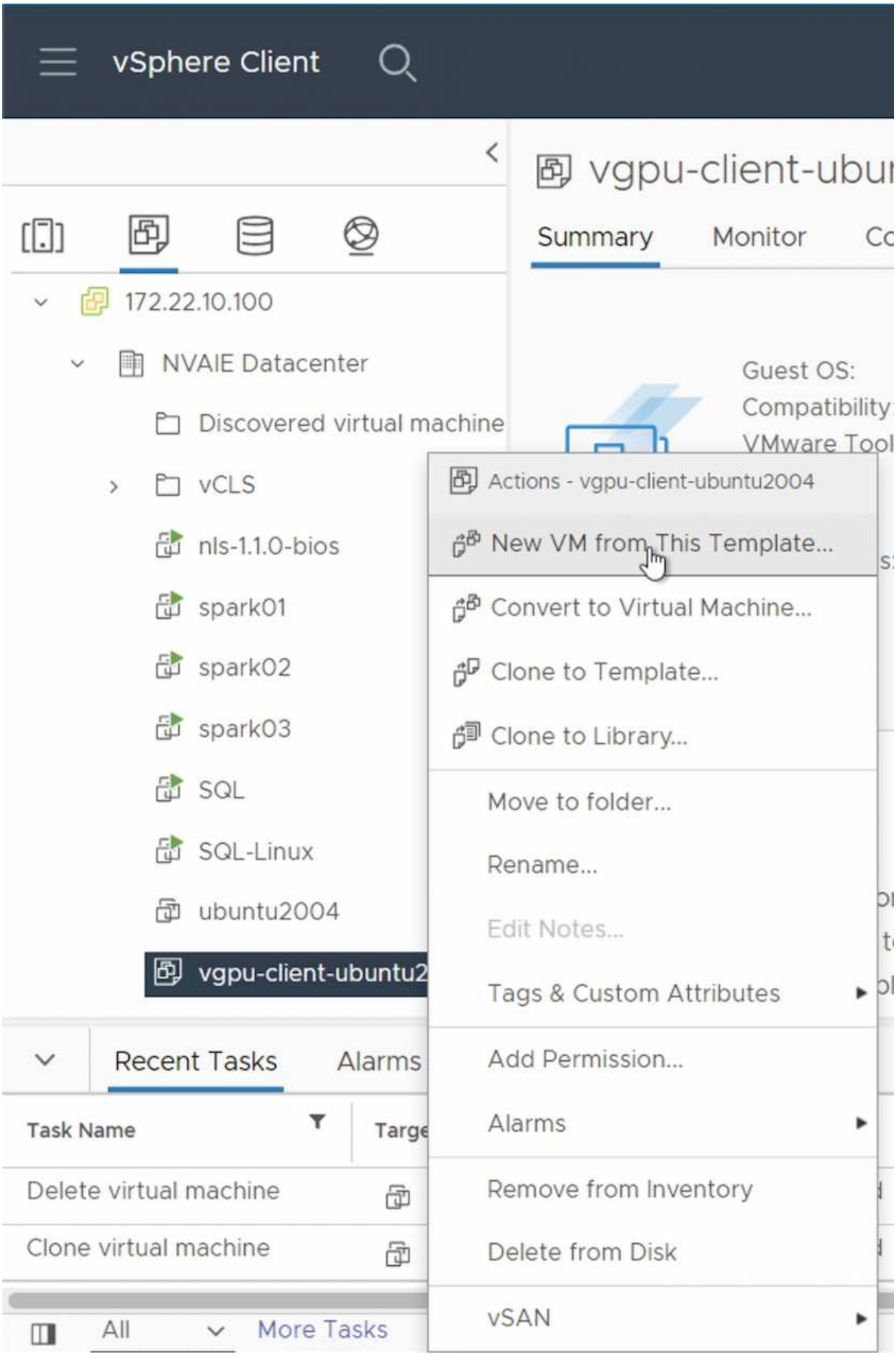
Questa sezione descrive le attività da eseguire per eseguire un lavoro di training TensorFlow in un ambiente NVIDIA ai Enterprise.

Prerequisiti

Prima di eseguire i passaggi descritti in questa sezione, si presuppone che sia già stato creato un modello di macchina virtuale guest seguendo le istruzioni riportate nella "Setup (Configurazione)" pagina.

Creare una macchina virtuale guest dal modello

Innanzitutto, è necessario creare una nuova macchina virtuale guest dal modello creato nella sezione precedente. Per creare una nuova macchina virtuale guest dal modello, accedere a VMware vSphere, fare clic sul nome del modello, scegliere 'Nuova macchina virtuale da questo modello…', quindi seguire la procedura guidata.


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

Creare e montare un volume di dati

Quindi, è necessario creare un nuovo volume di dati su cui memorizzare il set di dati di training. È possibile creare rapidamente un nuovo volume di dati utilizzando il NetApp DataOps Toolkit. Il comando di esempio che segue mostra la creazione di un volume denominato 'imagenet' con una capacità di 2 TB.

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

Prima di poter popolare i dati nel volume di dati, è necessario montarli all'interno della macchina virtuale guest. È possibile montare rapidamente un volume di dati utilizzando il NetApp DataOps Toolkit. Il comando di esempio che segue mostra il mouse del volume creato nel passaggio precedente.

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

Popolare il volume di dati

Una volta eseguito il provisioning e il montaggio del nuovo volume, è possibile recuperare il set di dati di training dalla posizione di origine e posizionarlo sul nuovo volume. In genere, ciò comporta il prelievo dei dati da un data Lake S3 o Hadoop e talvolta comporta l'aiuto di un data engineer.

Eseguire il lavoro di training TensorFlow

Ora, sei pronto per eseguire il tuo lavoro di training TensorFlow. Per eseguire il tuo lavoro di training TensorFlow, esegui le seguenti attività.

1. Estrarre l'immagine del container NVIDIA NGC Enterprise TensorFlow.

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

2. Avviare un'istanza di NVIDIA NGC Enterprise TensorFlow Container. Utilizzare l'opzione '-v' per collegare il volume di dati al container.

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

3. Esegui il tuo programma di training TensorFlow all'interno del container. Il comando di esempio che segue mostra l'esecuzione di un programma di training ResNet-50 di esempio incluso nell'immagine container.

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

Dove trovare ulteriori informazioni

Per ulteriori informazioni sulle informazioni descritte in questo documento, consultare i seguenti documenti e/o siti Web:

- Software per la gestione dei dati NetApp ONTAP: Libreria di informazioni ONTAP

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

- NetApp DataOps Toolkit

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

- NVIDIA ai Enterprise con VMware

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

Ringraziamenti

- Bobby Oommen, Sr Manager, NetApp
- Ramesh Isaac, System Administrator, NetApp
- Roney Daniel, Technical Marketing Engineer, NetApp

TR-4851: Data Lake NetApp StorageGRID per carichi di lavoro di guida autonoma - progettazione della soluzione

David Arnette, NetApp

TR-4851 dimostra l'utilizzo dello storage a oggetti NetApp StorageGRID come repository di dati e sistema di gestione per l'apprendimento automatico (ML) e lo sviluppo di software di deep learning (DL). Questo documento descrive il flusso di dati e i requisiti nello sviluppo di software per veicoli autonomi e le funzionalità di StorageGRID che ottimizzano il ciclo di vita dei dati. Questa soluzione si applica a qualsiasi workflow di pipeline di dati multistadio tipico dei processi di sviluppo ML e DL.

"TR-4851: Data Lake NetApp StorageGRID per carichi di lavoro di guida autonoma - progettazione della soluzione"

Piano di controllo ai di NetApp

TR-4798: Piano di controllo ai di NetApp

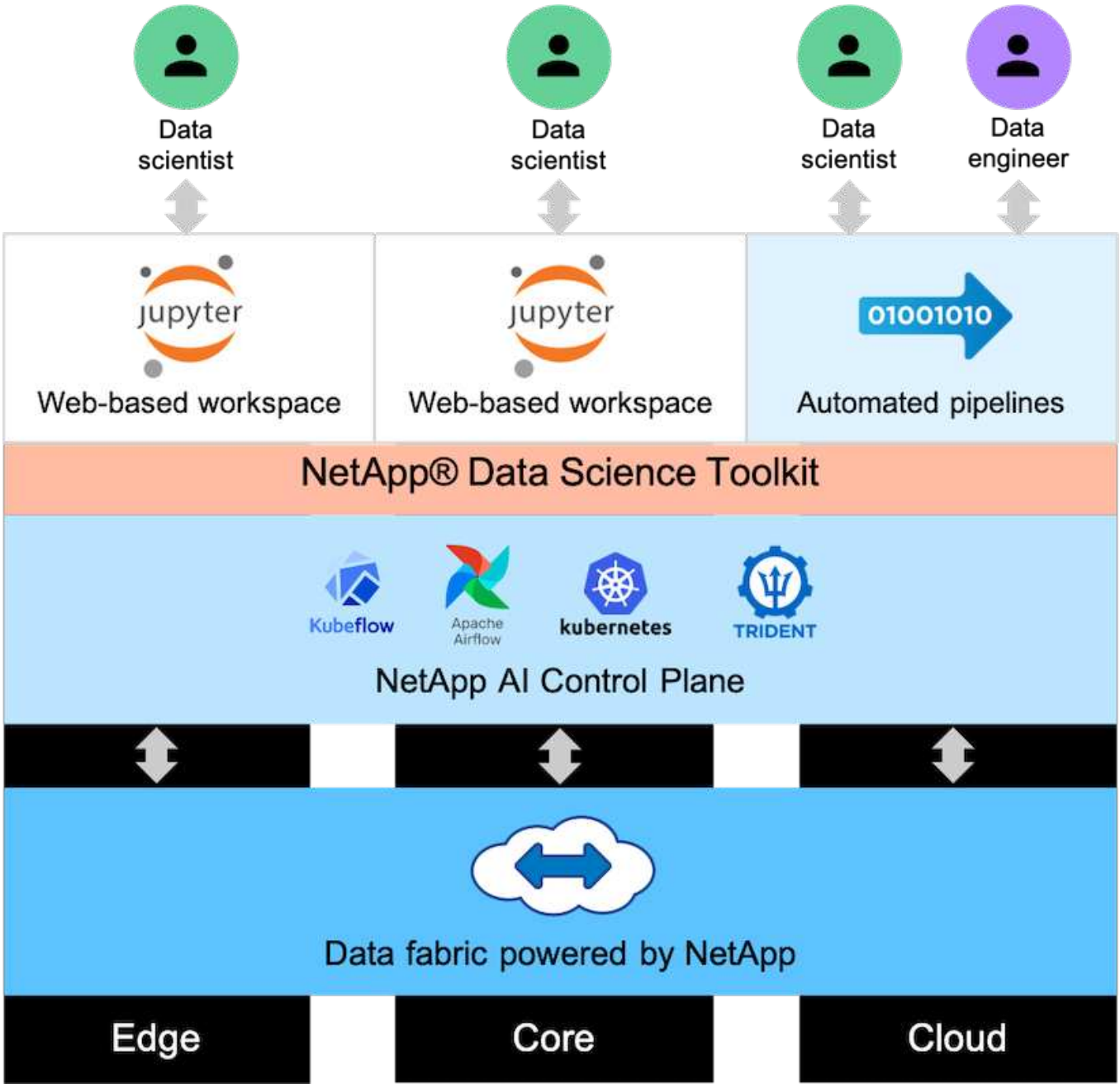
Mike Oglesby, NetApp

Aziende e organizzazioni di ogni dimensione e in molti settori stanno passando all'intelligenza artificiale (ai), all'apprendimento automatico (ML) e al deep learning (DL) per risolvere problemi reali, offrire prodotti e servizi innovativi e ottenere un vantaggio in un mercato sempre più competitivo. Man mano che le organizzazioni aumentano l'utilizzo di ai, ML e DL, devono affrontare molte sfide, tra cui la scalabilità dei workload e la disponibilità dei dati. Questo documento dimostra come affrontare queste sfide utilizzando il NetApp ai Control Plane, una soluzione che unisce le funzionalità di gestione dei dati di NetApp con i più diffusi framework e tool open-source.

Questo report mostra come clonare rapidamente uno spazio dei nomi dei dati. Mostra inoltre come replicare perfettamente i dati tra siti e regioni per creare una pipeline di dati ai/ML/DL coesa e unificata. Inoltre, ti guida attraverso la definizione e l'implementazione di workflow di training ai, ML e DL che incorporano la creazione quasi istantanea di dati e linee di base dei modelli per la tracciabilità e il controllo delle versioni. Con questa soluzione, è possibile tracciare ogni ciclo di training del modello fino all'esatto set di dati utilizzato per la formazione e/o la convalida del modello. Infine, questo documento illustra come eseguire rapidamente il provisioning delle aree di lavoro dei notebook Jupyter con accesso a set di dati di grandi dimensioni.

Nota: Per i training distribuiti in stile HPC su larga scala che coinvolgono un gran numero di server GPU che richiedono l'accesso condiviso allo stesso set di dati, o se si desidera un file system parallelo, consultare la sezione "TR-4890". Questo report tecnico descrive come includere "La soluzione di file system parallelo completamente supportata di NetApp BeeGFS" Come parte del NetApp ai Control Plane. Questa soluzione è progettata per scalare da una manciata di sistemi NVIDIA DGX A100 fino a un SuperPOD a 140 nodi completo.

Il piano di controllo ai di NetApp è rivolto a data scientist e data engineer e, di conseguenza, è necessaria una competenza minima di NetApp o NetApp ONTAP®. Con questa soluzione, le funzioni di gestione dei dati possono essere eseguite utilizzando interfacce e strumenti semplici e familiari. Se disponete già di storage NetApp nel vostro ambiente, potete testare il NetApp ai Control Plane oggi stesso. Se si desidera provare la soluzione ma non si dispone già di storage NetApp, visitare il sito "cloud.netapp.com"E potrai essere operativo con una soluzione di storage NetApp basata sul cloud in pochi minuti. La figura seguente fornisce una visualizzazione della soluzione.



Concetti e componenti

Intelligenza artificiale

L'ai è una disciplina informatica in cui i computer sono formati per imitare le funzioni cognitive della mente umana. Gli sviluppatori di ai addestrano i computer per imparare e risolvere i problemi in modo simile o addirittura superiore agli esseri umani. Il deep learning e l'apprendimento automatico sono sottocampi dell'ai. Le organizzazioni stanno adottando sempre più ai, ML e DL per supportare le loro esigenze aziendali critiche. Di seguito sono riportati alcuni esempi:

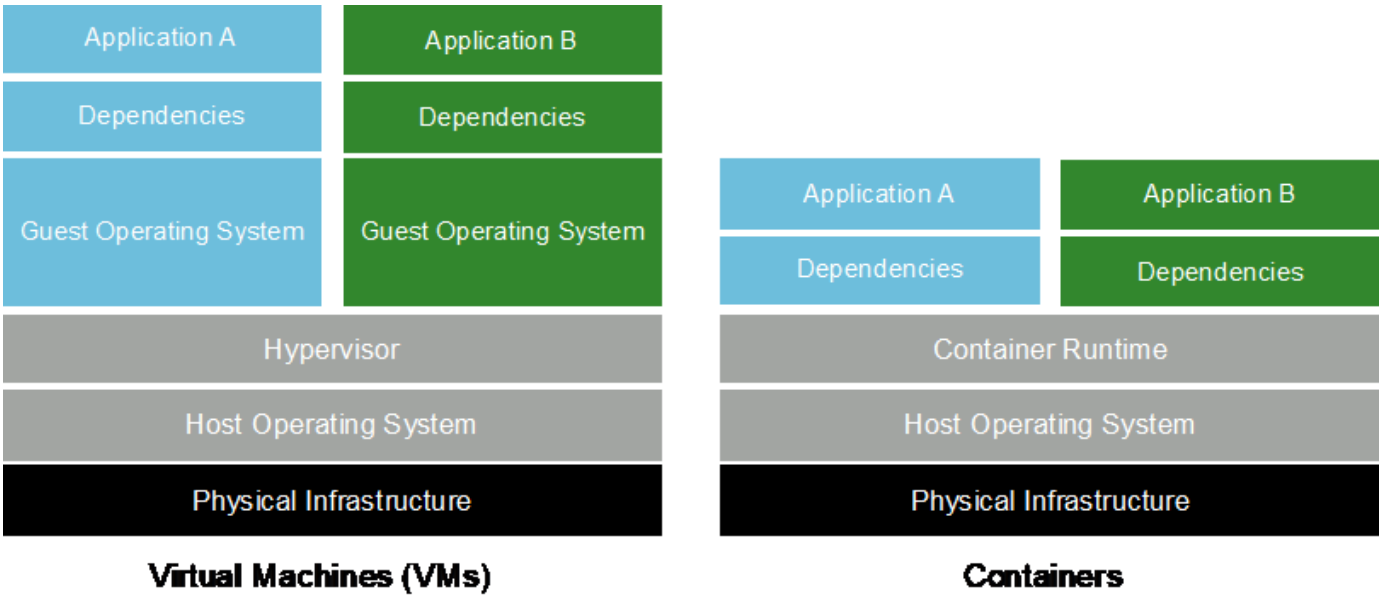
- Analisi di grandi quantità di dati per scoprire informazioni di business precedentemente sconosciute
- Interagire direttamente con i clienti utilizzando l'elaborazione del linguaggio naturale
- Automazione di vari processi e funzioni di business

I moderni carichi di lavoro di training e inferenza ai richiedono funzionalità di calcolo estremamente parallele. Pertanto, le GPU vengono sempre più utilizzate per eseguire le operazioni ai perché le funzionalità di elaborazione parallela delle GPU sono notevolmente superiori a quelle delle CPU generiche.

Container

I container sono istanze isolate dello spazio utente eseguite su un kernel del sistema operativo host condiviso. L'adozione dei container è in rapida crescita. I container offrono molti degli stessi vantaggi offerti dalle macchine virtuali (VM) per il sandboxing delle applicazioni. Tuttavia, poiché l'hypervisor e i livelli del sistema operativo guest su cui si basano le macchine virtuali sono stati eliminati, i container sono molto più leggeri. La figura seguente mostra una visualizzazione delle macchine virtuali rispetto ai container.

I container consentono inoltre un efficiente packaging delle dipendenze delle applicazioni, dei tempi di esecuzione e così via, direttamente con un'applicazione. Il formato di packaging dei container più comunemente utilizzato è Docker Container. Un'applicazione che è stata containerizzata nel formato Docker container può essere eseguita su qualsiasi computer in grado di eseguire i container Docker. Ciò è vero anche se le dipendenze dell'applicazione non sono presenti sul computer perché tutte le dipendenze sono contenute nel container stesso. Per ulteriori informazioni, visitare il "Sito web di Docker".



Kubernetes

Kubernetes è una piattaforma open source, distribuita e di orchestrazione dei container, originariamente progettata da Google e ora gestita dalla Cloud Native Computing Foundation (CNCF). Kubernetes consente l'automazione delle funzioni di implementazione, gestione e scalabilità per le applicazioni containerizzate. Negli ultimi anni, Kubernetes è emersa come piattaforma dominante per l'orchestrazione di container. Sebbene siano supportati altri formati di packaging dei container e tempi di esecuzione, Kubernetes viene spesso utilizzato come sistema di orchestrazione per i container Docker. Per ulteriori informazioni, visitare il "Sito web di Kubernetes".

Trident di NetApp

Trident è un orchestratore di storage open source sviluppato e gestito da NetApp che semplifica notevolmente la creazione, la gestione e il consumo dello storage persistente per i carichi di lavoro Kubernetes. Trident, un'applicazione nativa di Kubernetes, viene eseguita direttamente all'interno di un cluster Kubernetes. Con Trident, gli utenti di Kubernetes (sviluppatori, data scientist, amministratori di Kubernetes e così via) possono creare, gestire e interagire con volumi di storage persistenti nel formato standard di Kubernetes che già

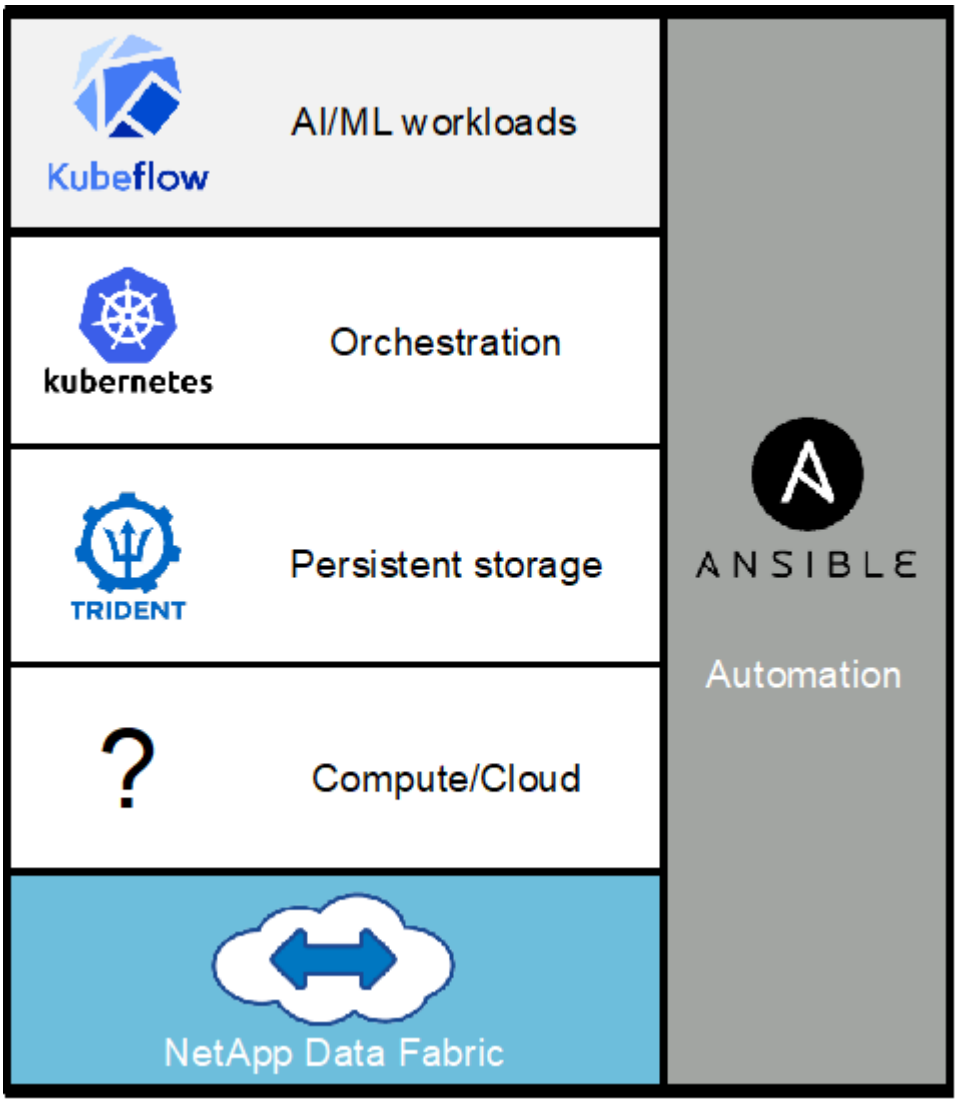
conoscono. Allo stesso tempo, possono sfruttare le funzionalità avanzate di gestione dei dati di NetApp e un data fabric basato sulla tecnologia NetApp. Trident astratta le complessità dello storage persistente e lo rende semplice da utilizzare. Per ulteriori informazioni, visitare il "Sito web di Trident".

NVIDIA DeepOps

DeepOps è un progetto open source di NVIDIA che, utilizzando Ansible, automatizza l'implementazione dei cluster di server GPU in base alle Best practice. DeepOps è modulare e può essere utilizzato per varie attività di implementazione. Per questo documento e per l'esercizio di convalida descritto, DeepOps viene utilizzato per implementare un cluster Kubernetes costituito da nodi di lavoro del server GPU. Per ulteriori informazioni, visitare il "Sito Web di DeepOps".

Kubeflow

Kubeflow è un toolkit open source ai e ML per Kubernetes sviluppato originariamente da Google. Il progetto Kubeflow rende le implementazioni dei flussi di lavoro ai e ML su Kubernetes semplici, portatili e scalabili. Kubeflow astratta le complessità di Kubernetes, consentendo agli scienziati dei dati di concentrarsi su ciò che conoscono meglio—data science. Vedere la figura seguente per una visualizzazione. Kubeflow ha ottenuto notevoli risultati con la sempre maggiore standardizzazione dei reparti IT aziendali su Kubernetes. Per ulteriori informazioni, visitare il "Sito web di Kubeflow".



Pipeline Kubeflow

Le pipeline Kubeflow sono un componente chiave di Kubeflow. Le pipeline Kubeflow sono una piattaforma e uno standard per la definizione e l'implementazione di flussi di lavoro portatili e scalabili ai e ML. Per ulteriori informazioni, consultare "Documentazione ufficiale del Kubeflow".

Jupyter notebook Server

Un Jupyter notebook Server è un'applicazione web open source che consente ai data scientist di creare documenti wiki-like denominati Jupyter Notebooks che contengono codice live e test descrittivi. I notebook Jupyter sono ampiamente utilizzati nella community ai e ML come mezzo per documentare, memorizzare e condividere progetti ai e ML. Kubeflow semplifica il provisioning e l'implementazione di Jupyter notebook Server su Kubernetes. Per ulteriori informazioni sui notebook Jupyter, visitare il "Sito web di Jupyter". Per ulteriori informazioni sui notebook Jupyter nel contesto di Kubeflow, vedere "Documentazione ufficiale del Kubeflow".

Flusso d'aria Apache

Apache Airflow è una piattaforma open-source per la gestione del workflow che consente authoring, scheduling e monitoraggio programmatici per flussi di lavoro aziendali complessi. Spesso viene utilizzato per automatizzare i flussi di lavoro ETL e della pipeline di dati, ma non è limitato a questi tipi di flussi di lavoro. Il progetto Airbnb è stato avviato da Airbnb, ma da allora è diventato molto popolare nel settore e ora è sotto gli auspici della Apache Software Foundation. Il flusso d'aria è scritto in Python, i flussi di lavoro del flusso d'aria sono creati tramite script Python e il flusso d'aria è progettato in base al principio della "configurazione come codice". Molti utenti del flusso d'aria aziendale ora eseguono il flusso d'aria su Kubernetes.

Diagrammi aciclici diretti (DAG)

Nel flusso d'aria, i flussi di lavoro sono denominati diagrammi ad aciclico diretto (DAG). I dag sono costituiti da task che vengono eseguiti in sequenza, in parallelo o in una combinazione dei due, a seconda della definizione DAG. Il programma di pianificazione del flusso d'aria esegue singole attività su un array di lavoratori, rispettando le dipendenze a livello di attività specificate nella definizione DAG. I dag vengono definiti e creati tramite script Python.

NetApp ONTAP 9

NetApp ONTAP 9 è l'ultima generazione di software per la gestione dello storage NetApp che consente a aziende come la tua di modernizzare l'infrastruttura e di passare a un data center cloud-ready. Grazie alle funzionalità di gestione dei dati leader del settore, ONTAP consente di gestire e proteggere i dati con un singolo set di strumenti, indipendentemente dalla posizione in cui risiedono. Puoi anche spostare liberamente i dati ovunque ti servano: Edge, core o cloud. ONTAP 9 include numerose funzionalità che semplificano la gestione dei dati, accelerano e proteggono i tuoi dati critici e la tua infrastruttura a prova di futuro attraverso architetture di cloud ibrido.

Semplifica la gestione dei dati

La gestione dei dati è fondamentale per le operazioni IT aziendali, in modo da poter utilizzare le risorse appropriate per le applicazioni e i set di dati. ONTAP include le seguenti funzionalità per ottimizzare e semplificare le operazioni e ridurre il costo totale delle operazioni:

- **Compattazione dei dati inline e deduplica estesa.** la compattazione dei dati riduce lo spazio sprecato all'interno dei blocchi di storage e la deduplica aumenta significativamente la capacità effettiva.
- **Qualità del servizio (QoS) minima, massima e adattiva.** i controlli QoS granulari aiutano a mantenere i livelli di performance per le applicazioni critiche in ambienti altamente condivisi.

- **ONTAP FabricPool.** questa funzione offre il tiering automatico dei dati cold per le opzioni di cloud storage pubblico e privato, tra cui Amazon Web Services (AWS), Azure e lo storage basato su oggetti NetApp StorageGRID.

Accelera e proteggi i dati

ONTAP offre livelli superiori di performance e protezione dei dati ed estende queste funzionalità con le seguenti funzionalità:

- **Prestazioni elevate e bassa latenza.** ONTAP offre il throughput più elevato possibile con la latenza più bassa possibile.
- **Tecnologia NetApp ONTAP FlexGroup.** Un volume FlexGroup è un container di dati dalle performance elevate che può scalare linearmente fino a 20 PB e 400 miliardi di file, fornendo un singolo namespace che semplifica la gestione dei dati.
- **Protezione dei dati.** ONTAP offre funzionalità di protezione dei dati integrate con gestione comune su tutte le piattaforme.
- **Crittografia dei volumi NetApp.** ONTAP offre crittografia nativa a livello di volume con supporto per la gestione delle chiavi sia integrata che esterna.

Infrastruttura a prova di futuro

ONTAP 9 aiuta a soddisfare le tue esigenze di business esigenti e in continua evoluzione:

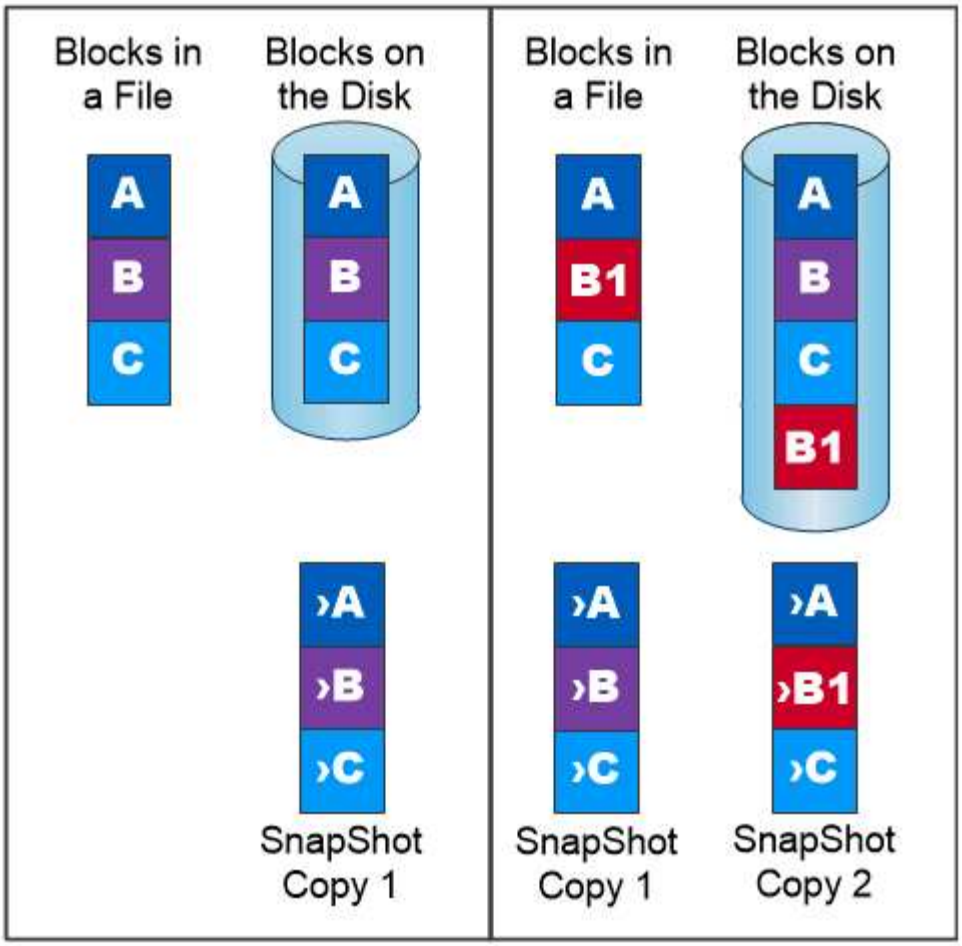
- **Scalabilità perfetta e operazioni senza interruzioni.** ONTAP supporta l'aggiunta senza interruzioni di capacità ai controller esistenti e ai cluster scale-out. Puoi eseguire l'upgrade alle tecnologie più recenti, come NVMe e 32GB FC, senza costose migrazioni dei dati o interruzioni.
- **Connessione al cloud.** ONTAP è uno dei software di gestione dello storage più connessi al cloud, con opzioni per lo storage definito tramite software (ONTAP Select) e le istanze native del cloud (NetApp Cloud Volumes Service) in tutti i cloud pubblici.
- **Integrazione con le applicazioni emergenti.** utilizzando la stessa infrastruttura che supporta le applicazioni aziendali esistenti, ONTAP offre servizi dati di livello Enterprise per piattaforme e applicazioni di prossima generazione come OpenStack, Hadoop e MongoDB.

Copie Snapshot di NetApp

Una copia Snapshot di NetApp è un'immagine point-in-time di sola lettura di un volume. L'immagine consuma uno spazio di storage minimo e comporta un overhead delle performance trascurabile, in quanto registra solo le modifiche apportate ai file creati dall'ultima copia Snapshot, come illustrato nella figura seguente.

Le copie Snapshot devono la loro efficienza alla tecnologia di virtualizzazione dello storage ONTAP principale, il layout di file Write Anywhere (WAFL). Come un database, WAFL utilizza i metadati per indicare i blocchi di dati effettivi sul disco. Tuttavia, a differenza di un database, WAFL non sovrascrive i blocchi esistenti. Scrive i dati aggiornati in un nuovo blocco e cambia i metadati. È perché ONTAP fa riferimento ai metadati quando crea una copia Snapshot, piuttosto che copiare i blocchi di dati, che le copie Snapshot sono così efficienti. In questo modo si eliminano i tempi di ricerca che altri sistemi devono affrontare per individuare i blocchi da copiare, nonché i costi di creazione della copia stessa.

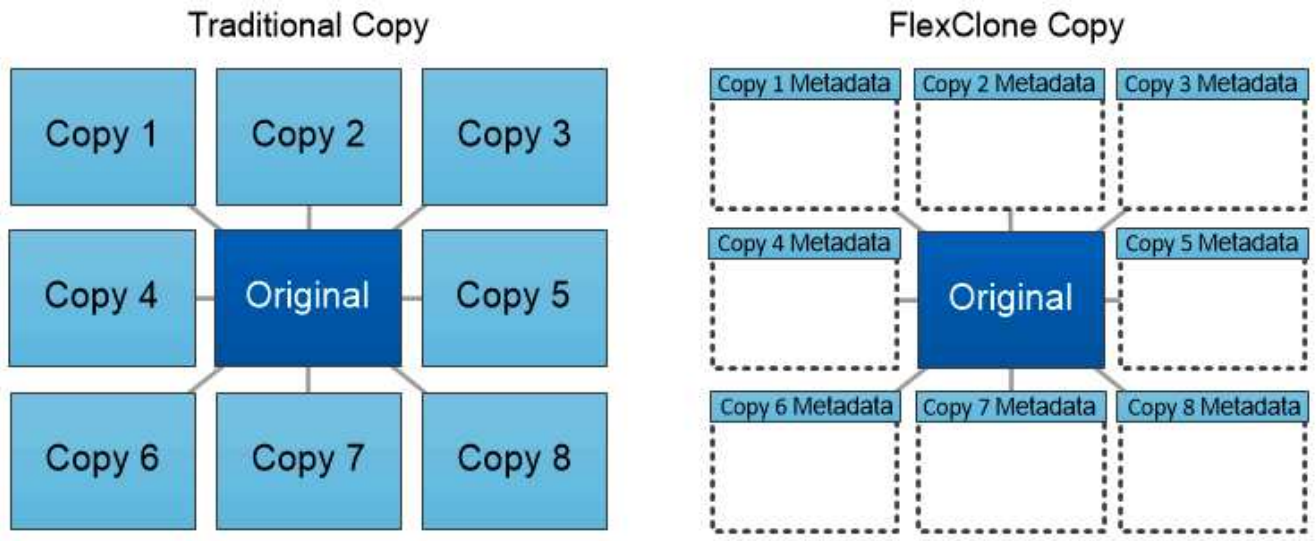
È possibile utilizzare una copia Snapshot per ripristinare singoli file o LUN o per ripristinare l'intero contenuto di un volume. ONTAP confronta le informazioni del puntatore nella copia Snapshot con i dati su disco per ricostruire l'oggetto mancante o danneggiato, senza downtime o costi di performance significativi.



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

Tecnologia NetApp FlexClone

La tecnologia NetApp FlexClone fa riferimento ai metadati Snapshot per creare copie scrivibili point-in-time di un volume. Le copie condividono i blocchi di dati con i genitori, senza consumare storage, ad eccezione di quanto richiesto per i metadati fino a quando le modifiche non vengono scritte nella copia, come illustrato nella figura seguente. Il software FlexClone consente di copiare quasi istantaneamente anche i set di dati più grandi, anche se le copie tradizionali richiedono pochi minuti o persino ore. Ciò lo rende ideale per situazioni in cui sono necessarie più copie di set di dati identici (ad esempio un'area di lavoro di sviluppo) o copie temporanee di un set di dati (test di un'applicazione rispetto a un set di dati di produzione).

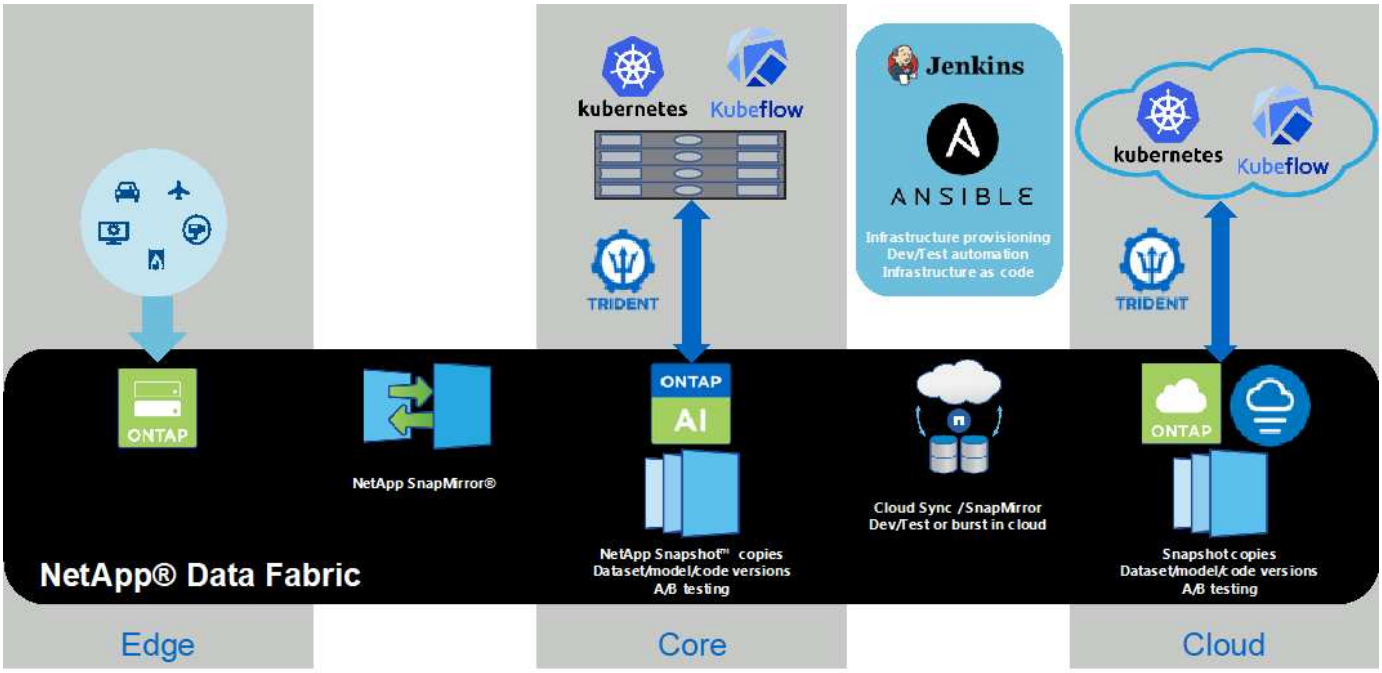


FlexClone copies share data blocks with their parents, consuming no storage except what is required for metadata.

Tecnologia NetApp SnapMirror Data Replication

Il software NetApp SnapMirror è una soluzione di replica unificata conveniente e facile da utilizzare per tutto il data fabric. Replica i dati ad alta velocità su LAN o WAN. Offre un'elevata disponibilità dei dati e una rapida replica dei dati per applicazioni di tutti i tipi, incluse le applicazioni business-critical in ambienti virtuali e tradizionali. Quando si replicano i dati su uno o più sistemi storage NetApp e si aggiornano continuamente i dati secondari, i dati vengono mantenuti aggiornati e disponibili quando necessario. Non sono richiesti server di replica esterni. Vedere la figura seguente per un esempio di architettura che sfrutta la tecnologia SnapMirror.

Il software SnapMirror sfrutta le efficienze dello storage NetApp ONTAP inviando solo i blocchi modificati sulla rete. Il software SnapMirror utilizza inoltre la compressione di rete integrata per accelerare i trasferimenti di dati e ridurre l'utilizzo della larghezza di banda di rete fino al 70%. Con la tecnologia SnapMirror, è possibile sfruttare un flusso di dati di replica con risorse limitate per creare un singolo repository che mantiene il mirror attivo e le copie point-in-time precedenti, riducendo il traffico di rete fino al 50%.



Copia e sincronizzazione di NetApp BlueXP

La copia e sincronizzazione di BlueXP è un servizio NetApp per una sincronizzazione dei dati rapida e sicura. Sia che tu debba trasferire file tra condivisioni di file NFS o SMB on-premise, NetApp StorageGRID, NetApp ONTAP S3, NetApp Cloud Volumes Service, Azure NetApp Files, AWS S3, AWS EFS, BLOB di Azure, Google Cloud Storage, o IBM Cloud Object Storage, BlueXP Copy e Sync sposta i file dove ne hai bisogno in modo rapido e sicuro.

Una volta trasferiti, i dati sono completamente disponibili per l'utilizzo sia sull'origine che sulla destinazione. La copia e sincronizzazione di BlueXP può sincronizzare i dati on-demand quando viene attivato un aggiornamento o sincronizzare costantemente i dati in base a una pianificazione predefinita. Indipendentemente, BlueXP Copy e Sync sposta solo i delta, così tempo e denaro spesi per la replica dei dati sono ridotti al minimo.

BlueXP Copy and Sync è un tool software as a service (SaaS) estremamente semplice da configurare e utilizzare. I trasferimenti dei dati attivati da BlueXP Copy e Sync sono effettuati dai broker di dati. I data broker di BlueXP Copy e Sync possono essere implementati in AWS, Azure, Google Cloud Platform o on-premise.

XCP di NetApp

NetApp XCP è un software basato su client per migrazioni di dati da qualsiasi a NetApp e da NetApp a NetApp e informazioni sui file system. XCP è progettato per scalare e ottenere le massime performance utilizzando tutte le risorse di sistema disponibili per gestire set di dati ad alto volume e migrazioni ad alte performance. XCP consente di ottenere una visibilità completa nel file system con la possibilità di generare report.

NetApp XCP è disponibile in un singolo pacchetto che supporta i protocolli NFS e SMB. XCP include un binario Linux per set di dati NFS e un eseguibile Windows per set di dati SMB.

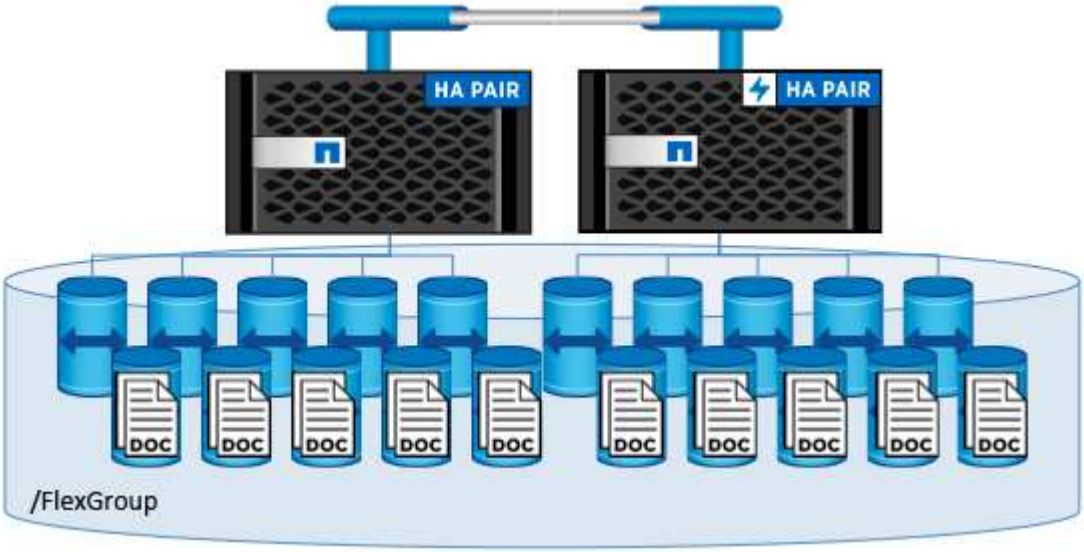
NetApp XCP file Analytics è un software basato su host che rileva le condivisioni di file, esegue scansioni sul file system e fornisce una dashboard per l'analisi dei file. XCP file Analytics è compatibile con sistemi NetApp e non NetApp ed è eseguibile su host Linux o Windows per fornire analisi per NFS e file system esportati da SMB.

NetApp ONTAP FlexGroup Volumes

Un set di dati di training può essere una raccolta di potenzialmente miliardi di file. I file possono includere testo, audio, video e altre forme di dati non strutturati che devono essere memorizzati ed elaborati per essere letti in parallelo. Il sistema di storage deve memorizzare un numero elevato di file di piccole dimensioni e leggerli in parallelo per l'i/o sequenziale e casuale

Un volume FlexGroup è un singolo namespace che comprende più volumi membri costitutivi, come illustrato nella figura seguente. Dal punto di vista dell'amministratore dello storage, un volume FlexGroup viene gestito e agisce come un volume NetApp FlexVol. I file in un volume FlexGroup vengono allocati a singoli volumi membri e non vengono sottoposti a striping tra volumi o nodi. Consentono le seguenti funzionalità:

- I volumi FlexGroup offrono diversi petabyte di capacità e bassa latenza prevedibile per carichi di lavoro con metadati elevati.
- Supportano fino a 400 miliardi di file nello stesso spazio dei nomi.
- Supportano operazioni parallelizzate nei carichi di lavoro NAS tra CPU, nodi, aggregati e volumi FlexVol costitutivi.



Requisiti hardware e software

La soluzione NetApp ai Control Plane non dipende da questo hardware specifico. La soluzione è compatibile con qualsiasi appliance di storage fisico, istanza software-defined o servizio cloud NetApp, supportato da Trident. Ad esempio, un sistema di storage NetApp AFF, Azure NetApp Files, NetApp Cloud Volumes Service, un'istanza di storage NetApp ONTAP Select definita tramite software o un'istanza di NetApp Cloud Volumes ONTAP. Inoltre, la soluzione può essere implementata su qualsiasi cluster Kubernetes purché la versione di Kubernetes utilizzata sia supportata da Kubeflow e NetApp Trident. Per un elenco delle versioni di Kubernetes supportate da Kubeflow, vedere la "Documentazione ufficiale del Kubeflow". Per un elenco delle versioni di Kubernetes supportate da Trident, vedere "Documentazione di Trident". Per informazioni dettagliate sull'ambiente utilizzato per la convalida della soluzione, consultare le tabelle seguenti.

Componente dell'infrastruttura	Quantità	Dettagli	Sistema operativo
Host di salto per l'implementazione	1	MACCHINA VIRTUALE	Ubuntu 20.04.2 LTS
Nodi master Kubernetes	1	MACCHINA VIRTUALE	Ubuntu 20.04.2 LTS
Nodi di lavoro Kubernetes	2	MACCHINA VIRTUALE	Ubuntu 20.04.2 LTS
Kubernetes nodi di lavoro GPU	2	NVIDIA DGX-1 (bare-metal)	NVIDIA DGX OS 4.0.5 (basato su Ubuntu 18.04.2 LTS)
Storage	1 coppia ha	NetApp AFF A220	NetApp ONTAP 9.7 P6

Componente software	Versione
Flusso d'aria Apache	2.0.1
Helm Chart di Apache Airflow	8.0.8
Docker	19.03.12
Kubeflow	1.2
Kubernetes	1.18.9
Trident di NetApp	21.01.2
NVIDIA DeepOps	Funzionalità di implementazione di Trident dalla filiale master al momento del commit "61898cdfda"; Tutte le altre funzionalità dalla versione 21.03

Supporto

NetApp non offre supporto Enterprise per Apache Airflow, Docker, Kubeflow, Kubernetes o NVIDIA DeepOps. Se sei interessato a una soluzione completamente supportata con funzionalità simili alla soluzione NetApp ai Control Plane, "Contatta NetApp" Informazioni sulle soluzioni ai/ML completamente supportate che NetApp offre insieme ai partner.

Implementazione di Kubernetes

Questa sezione descrive le attività da completare per implementare un cluster Kubernetes in cui implementare la soluzione NetApp ai Control Plane. Se si dispone già di un cluster Kubernetes, è possibile saltare questa sezione se si utilizza una versione di Kubernetes supportata da Kubeflow e NetApp Trident. Per un elenco delle versioni di Kubernetes supportate da Kubeflow, vedere la "Documentazione ufficiale del Kubeflow". Per un elenco delle versioni di Kubernetes supportate da Trident, vedere "Documentazione di Trident".

Per le implementazioni on-premise di Kubernetes che incorporano nodi bare-metal con GPU NVIDIA, NetApp consiglia di utilizzare il tool di implementazione DeepOps Kubernetes di NVIDIA. Questa sezione descrive l'implementazione di un cluster Kubernetes utilizzando DeepOps.

Prerequisiti

Prima di eseguire l'esercizio di implementazione descritto in questa sezione, si presuppone che siano già state eseguite le seguenti attività:

1. Sono già stati configurati nodi Kubernetes bare-metal (ad esempio, un sistema NVIDIA DGX che fa parte di un pod ai ONTAP) in base alle istruzioni di configurazione standard.
2. È stato installato un sistema operativo supportato su tutti i nodi master e worker di Kubernetes e su un host di distribuzione jump. Per un elenco dei sistemi operativi supportati da DeepOps, vedere "Sito DeepOps GitHub".

Utilizzare NVIDIA DeepOps per installare e configurare Kubernetes

Per implementare e configurare il cluster Kubernetes con NVIDIA DeepOps, eseguire le seguenti operazioni da un host di distribuzione jump:

1. Scaricare NVIDIA DeepOps seguendo le istruzioni sul "Pagina introduttiva" Sul sito NVIDIA DeepOps GitHub.
2. Implementare Kubernetes nel cluster seguendo le istruzioni sul "Pagina della Guida all'implementazione di Kubernetes" Sul sito NVIDIA DeepOps GitHub.

Implementazione e configurazione di NetApp Trident

Implementazione e configurazione di NetApp Trident

Questa sezione descrive le attività da completare per installare e configurare NetApp Trident nel cluster Kubernetes.

Prerequisiti

Prima di eseguire l'esercizio di implementazione descritto in questa sezione, si presuppone che siano già state eseguite le seguenti attività:

1. Hai già un cluster Kubernetes funzionante e stai eseguendo una versione di Kubernetes supportata da Trident. Per un elenco delle versioni supportate, vedere "Documentazione di Trident".
2. Disponete già di un'appliance di storage NetApp funzionante, di un'istanza software-defined o di un servizio di cloud storage supportato da Trident.

Installare Trident

Per installare e configurare NetApp Trident nel cluster Kubernetes, eseguire le seguenti attività dall'host di distribuzione jump:

1. Implementare Trident utilizzando uno dei seguenti metodi:
    - Se hai utilizzato NVIDIA DeepOps per implementare il cluster Kubernetes, puoi anche utilizzare NVIDIA DeepOps per implementare Trident nel cluster Kubernetes. Per implementare Trident con DeepOps, seguire "Istruzioni per l'implementazione di Trident" Sul sito NVIDIA DeepOps GitHub.
    - Se non hai utilizzato NVIDIA DeepOps per implementare il cluster Kubernetes o se preferisci semplicemente implementare Trident manualmente, puoi implementare Trident seguendo la "istruzioni per l'implementazione" Nella documentazione di Trident. Per ulteriori informazioni sulla configurazione, assicurarsi di creare almeno un backend Trident e almeno un StorageClass Kubernetes "Back-end" e. "StorageClasses" Consulta le sottosezioni collegate nei documenti di NetApp.

Se stai implementando la soluzione NetApp per il piano di controllo ai su un pod ai ONTAP, consulta "Esempi di backend Trident per implementazioni ai ONTAP" Per alcuni esempi di diversi backend Trident che si desidera creare e. "Esempi di storaglasses Kubernetes per implementazioni ai ONTAP" Per alcuni esempi di diverse Kubernetes StorageClasses che potresti voler creare.

Esempi di backend Trident per implementazioni ai ONTAP

Prima di utilizzare Trident per eseguire il provisioning dinamico delle risorse di storage all'interno del cluster Kubernetes, è necessario creare uno o più backend Trident. Gli esempi che seguono rappresentano diversi tipi di backend che è possibile creare se si sta implementando la soluzione per il piano di controllo ai di NetApp su un pod ai di ONTAP. Per ulteriori informazioni sui backend, consultare "Documentazione di Trident".

1. NetApp consiglia di creare un backend Trident abilitato per FlexGroup per ogni LIF di dati (interfaccia di rete logica che fornisce l'accesso ai dati) che si desidera utilizzare sul sistema NetApp AFF. In questo modo, potrai bilanciare i montaggi di volume tra le LIF

   I comandi di esempio che seguono mostrano la creazione di due backend Trident abilitati per FlexGroup per due diverse LIF di dati associate alla stessa SVM (Storage Virtual Machine) di ONTAP. Questi backend utilizzano `ontap-nas-flexgroup` driver di storage. ONTAP supporta due tipi principali di volumi di dati: FlexVol e FlexGroup. I volumi FlexVol sono limitati dalle dimensioni (al momento della scrittura, le dimensioni massime dipendono dalla distribuzione specifica). I volumi FlexGroup, invece, possono scalare linearmente fino a 20 PB e 400 miliardi di file, fornendo un singolo namespace che semplifica notevolmente la gestione dei dati. Pertanto, i volumi FlexGroup sono ottimali per i carichi di lavoro ai e ML che si basano su grandi quantità di dati.

   Se si lavora con una piccola quantità di dati e si desidera utilizzare volumi FlexVol invece di volumi FlexGroup, è possibile creare backend Trident che utilizzano `ontap-nas` driver di storage invece di `ontap-nas-flexgroup` driver di storage.

```
$ 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 consiglia inoltre di creare uno o più backend Trident abilitati per FlexVol. Se si utilizzano volumi

FlexGroup per lo storage dei dataset di training, è possibile utilizzare volumi FlexVol per memorizzare risultati, output, informazioni di debug e così via. Se si desidera utilizzare i volumi FlexVol, è necessario creare uno o più backend Trident abilitati per FlexVol. I comandi di esempio che seguono mostrano la creazione di un singolo backend Trident abilitato a FlexVol che utilizza una singola LIF di dati.

```
$ 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 |
+----------------------------+---------------------
+--------------------------------------+--------+---------+
```

Esempi di storage Classes Kubernetes per implementazioni ai ONTAP

Prima di utilizzare Trident per eseguire il provisioning dinamico delle risorse di storage all'interno del cluster Kubernetes, è necessario creare una o più Kubernetes StorageClasses. Gli esempi che seguono rappresentano diversi tipi di StorageClasses che potresti voler creare se stai implementando la soluzione NetApp ai Control Plane su un pod ai ONTAP. Per ulteriori informazioni su StorageClasses, vedere "Documentazione di Trident".

1. NetApp consiglia di creare una StorageClass separata per ogni backend Trident abilitato per FlexGroup creato nella sezione "Esempi di backend Trident per implementazioni ai ONTAP", fase 1. Questi StorageClasses granulari consentono di aggiungere i montaggi NFS che corrispondono a LIF specifiche (le LIF specificate al momento della creazione dei Trident Backend) come backend specifico specificato nel file delle specifiche StorageClass. I comandi di esempio che seguono mostrano la creazione di due StorageClasses che corrispondono ai due backend di esempio creati nella sezione "Esempi di backend Trident per implementazioni ai ONTAP", fase 1. Per ulteriori informazioni su StorageClasses, vedere "Documentazione di Trident".

   Per evitare che un volume persistente venga cancellato quando il PVC (PersistentVolumeClaim) corrispondente viene cancellato, nel seguente esempio viene utilizzato un `reclaimPolicy` valore di `Retain`. Per ulteriori informazioni su `reclaimPolicy` vedi il sito ufficiale "Documentazione 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 consiglia inoltre di creare un StorageClass che corrisponda al backend Trident abilitato a FlexVol creato nella sezione "Esempi di backend Trident per implementazioni ai ONTAP", punto 2. I comandi di esempio che seguono mostrano la creazione di una singola classe di storage per volumi FlexVol.

   Nell'esempio seguente, un particolare backend non viene specificato nel file di definizione StorageClass perché è stato creato un solo backend Trident abilitato a FlexVol. Quando si utilizza Kubernetes per amministrare volumi che utilizzano questo StorageClass, Trident tenta di utilizzare qualsiasi backend disponibile che utilizzi `ontap-nas` driver.

```
$ 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 consiglia inoltre di creare una classe di storage generica per i volumi FlexGroup. I seguenti comandi di esempio mostrano la creazione di una singola classe di storage generica per volumi FlexGroup.

   Si noti che un particolare backend non viene specificato nel file di definizione StorageClass. Pertanto, quando si utilizza Kubernetes per amministrare volumi che utilizzano questo StorageClass, Trident tenta di utilizzare qualsiasi backend disponibile che utilizzi `ontap-nas-flexgroup` driver.

```
$ 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
```

Implementazione di Kubeflow

In questa sezione vengono descritte le attività da completare per implementare Kubeflow nel cluster Kubernetes.

Prerequisiti

Prima di eseguire l'esercizio di implementazione descritto in questa sezione, si presuppone che siano già state eseguite le seguenti attività:

1. Hai già un cluster Kubernetes funzionante e stai eseguendo una versione di Kubernetes supportata da Kubeflow. Per un elenco delle versioni supportate, vedere "Documentazione ufficiale del Kubeflow".
2. NetApp Trident è già stato installato e configurato nel cluster Kubernetes, come descritto in "Implementazione e configurazione di Trident".

Impostare la classe di storage Kubernetes predefinita

Prima di implementare Kubeflow, è necessario specificare un StorageClass predefinito all'interno del cluster Kubernetes. Il processo di implementazione di Kubeflow tenta di eseguire il provisioning di nuovi volumi persistenti utilizzando la classe di storage predefinita. Se non viene indicato StorageClass come StorageClass predefinito, l'implementazione non riesce. Per designare una StorageClass predefinita all'interno del cluster, eseguire la seguente attività dall'host di distribuzione jump. Se è già stata designata una StorageClass predefinita all'interno del cluster, è possibile saltare questo passaggio.

1. Designare uno dei StorageClasses esistenti come StorageClass predefinito. I comandi di esempio che seguono mostrano la designazione di StorageClass denominata `ontap-ai- flexvols-retain` Come StorageClass di default.

> Il `ontap-nas-flexgroup` Il tipo di backend Trident ha una dimensione minima del PVC che è abbastanza grande. Per impostazione predefinita, Kubeflow tenta di eseguire il provisioning di PVC di dimensioni limitate a poche GB. Pertanto, non è necessario designare un StorageClass che utilizzi `ontap-nas-flexgroup` Tipo di backend come StorageClass predefinito ai fini dell'implementazione di 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
```

Utilizza NVIDIA DeepOps per implementare Kubeflow

NetApp consiglia di utilizzare il tool di implementazione Kubeflow fornito da NVIDIA DeepOps. Per implementare Kubeflow nel cluster Kubernetes utilizzando lo strumento di implementazione DeepOps, eseguire le seguenti operazioni dall'host di distribuzione jump.

> In alternativa, è possibile implementare Kubeflow manualmente seguendo la "istruzioni per l'installazione" Nella documentazione ufficiale del Kubeflow

1. Implementare Kubeflow nel cluster seguendo la "Istruzioni per l'implementazione di Kubeflow" Sul sito NVIDIA DeepOps GitHub.
2. Annotare l'URL del dashboard Kubeflow prodotto dal tool di implementazione 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. Verificare che tutti i pod implementati nello spazio dei nomi Kubeflow mostrino un `STATUS` di `Running` e verificare che nessun componente implementato all'interno dello spazio dei nomi sia in stato di errore. L'avvio di tutti i pod potrebbe richiedere alcuni minuti.

```
$ 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
```

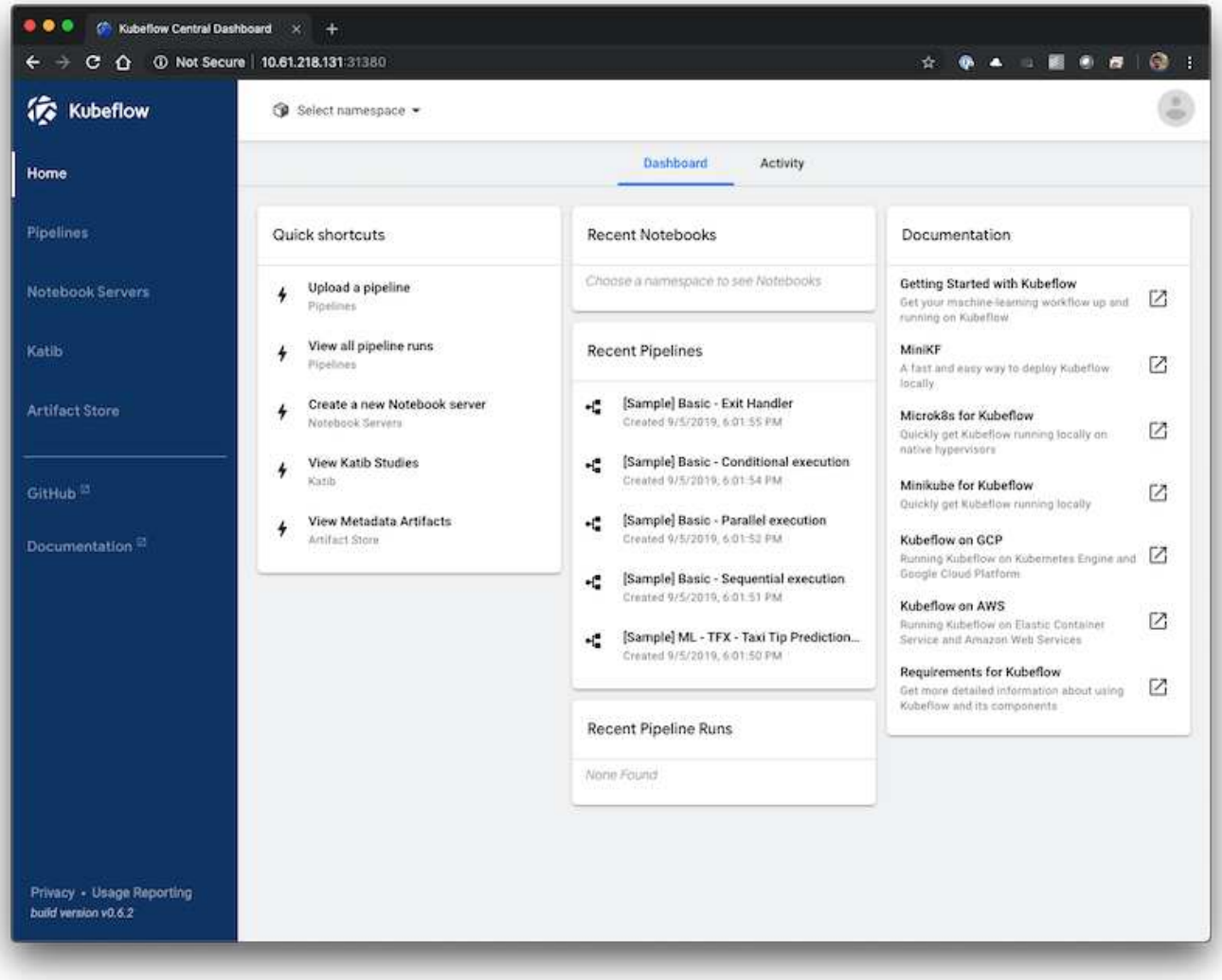
```
deployment.apps/katib-db                                    1/1     1
1           97s
deployment.apps/katib-manager                               1/1     1
1           96s
deployment.apps/katib-manager-rest                          1/1     1
1           96s
deployment.apps/katib-suggestion-bayesianoptimization       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. Nel browser Web, accedere alla dashboard centrale di Kubeflow accedendo all'URL annotato al punto 2.

Il nome utente predefinito è `admin@kubeflow.org`e la password predefinita è `12341234`. Per creare altri utenti, seguire le istruzioni in "Documentazione ufficiale del Kubeflow".



Operazioni e attività Kubeflow di esempio

Questa sezione include esempi di varie operazioni e attività che è possibile eseguire utilizzando Kubeflow.

Operazioni e attività Kubeflow di esempio

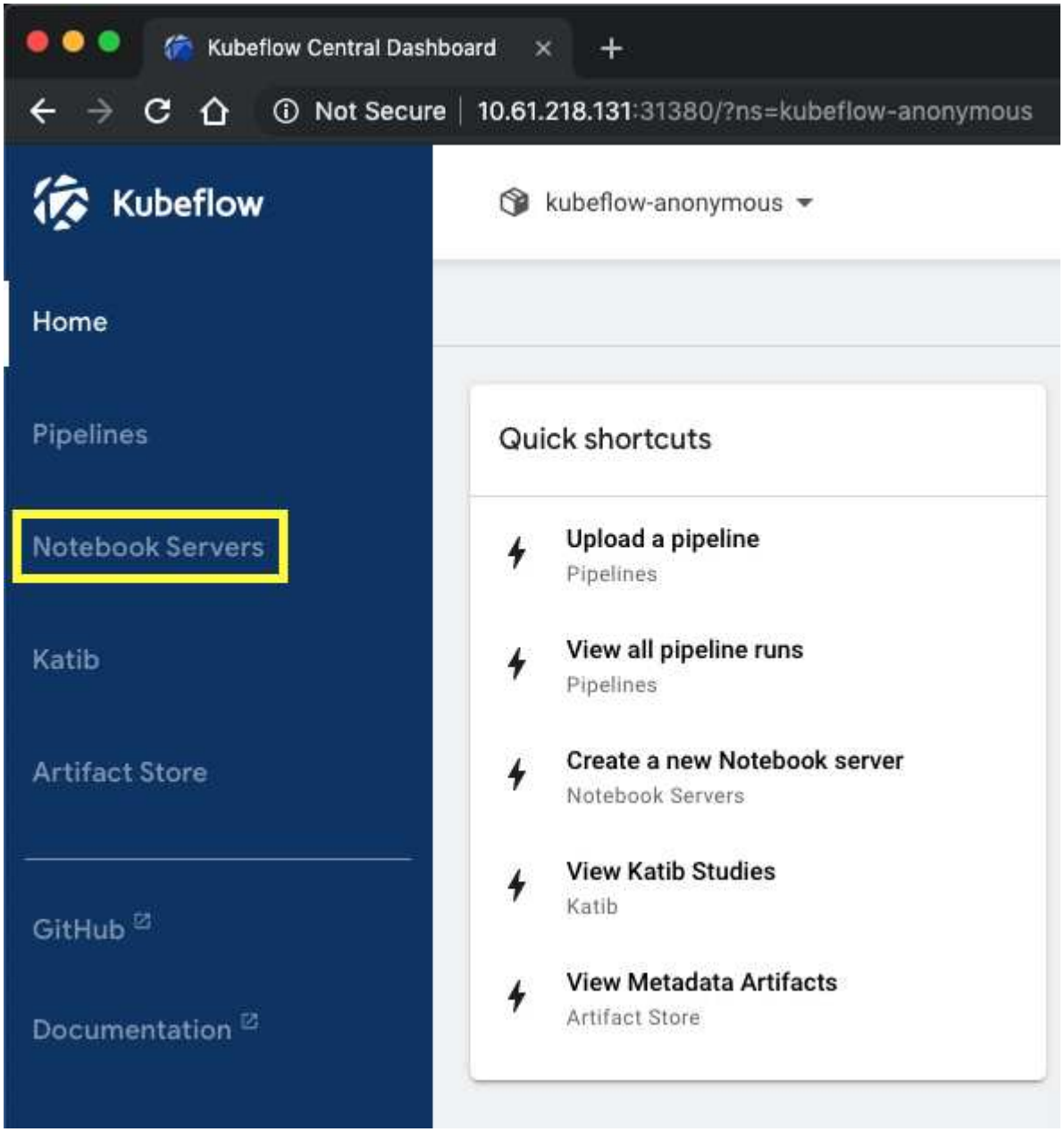
Questa sezione include esempi di varie operazioni e attività che è possibile eseguire utilizzando Kubeflow.

Provisioning di un’area di lavoro Jupyter notebook per l’utilizzo da parte di Data Scientist o Developer

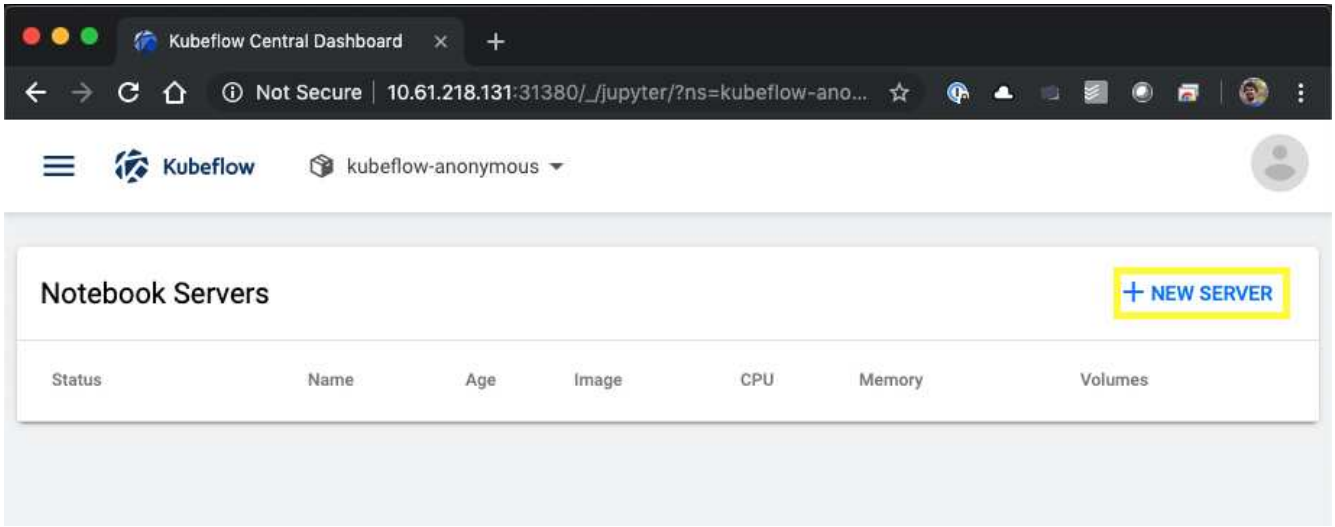
Kubeflow è in grado di eseguire rapidamente il provisioning dei nuovi server Jupyter notebook per agire come aree di lavoro per scienziati dei dati. Per eseguire il provisioning di un nuovo server Jupyter notebook con Kubeflow, eseguire le seguenti operazioni. Per ulteriori informazioni sui notebook Jupyter all’interno del contesto Kubeflow, vedere

"Documentazione ufficiale del Kubeflow".

1. Dalla dashboard centrale di Kubeflow, fare clic su notebook Servers nel menu principale per accedere alla pagina di amministrazione del server Jupyter notebook.

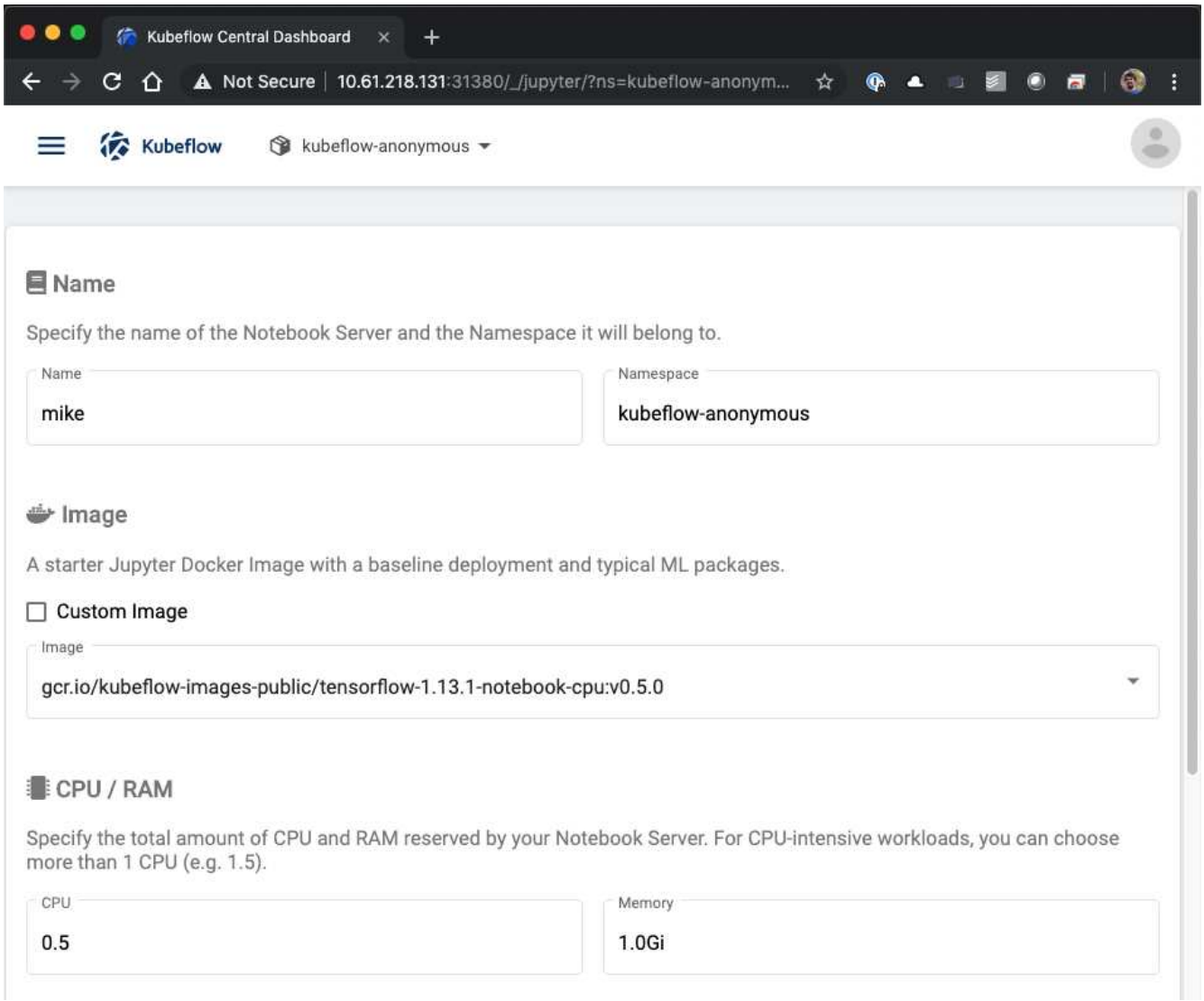


2. Fare clic su New Server (nuovo server) per eseguire il provisioning di un nuovo server Jupyter notebook.

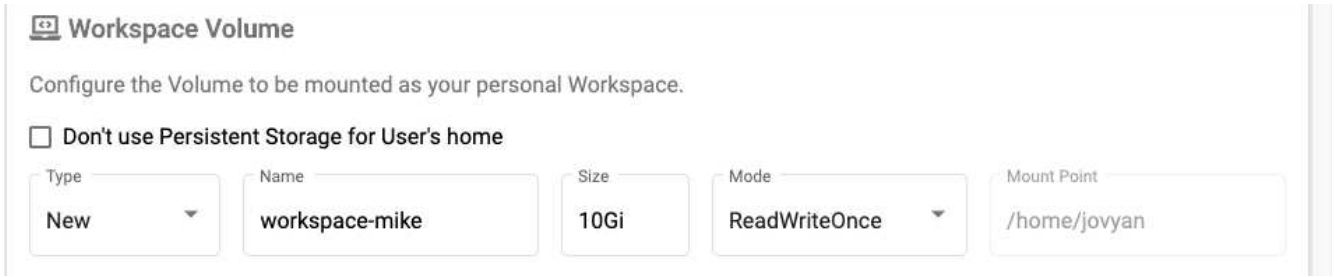


3. Assegnare un nome al nuovo server, scegliere l'immagine Docker su cui si desidera basare il server e specificare la quantità di CPU e RAM da riservare al server. Se il campo namespace è vuoto, utilizzare il menu Select namespace (Seleziona spazio dei nomi) nell'intestazione della pagina per scegliere uno spazio dei nomi. Il campo namespace viene quindi compilato automaticamente con lo spazio dei nomi scelto.

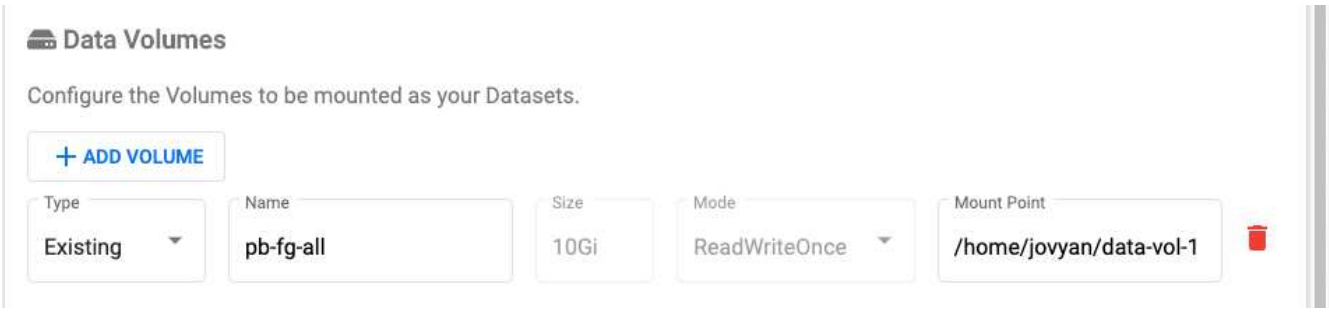
   Nell'esempio seguente, il `kubeflow-anonymous` viene scelto lo spazio dei nomi. Inoltre, vengono accettati i valori predefiniti per l'immagine Docker, la CPU e la RAM.



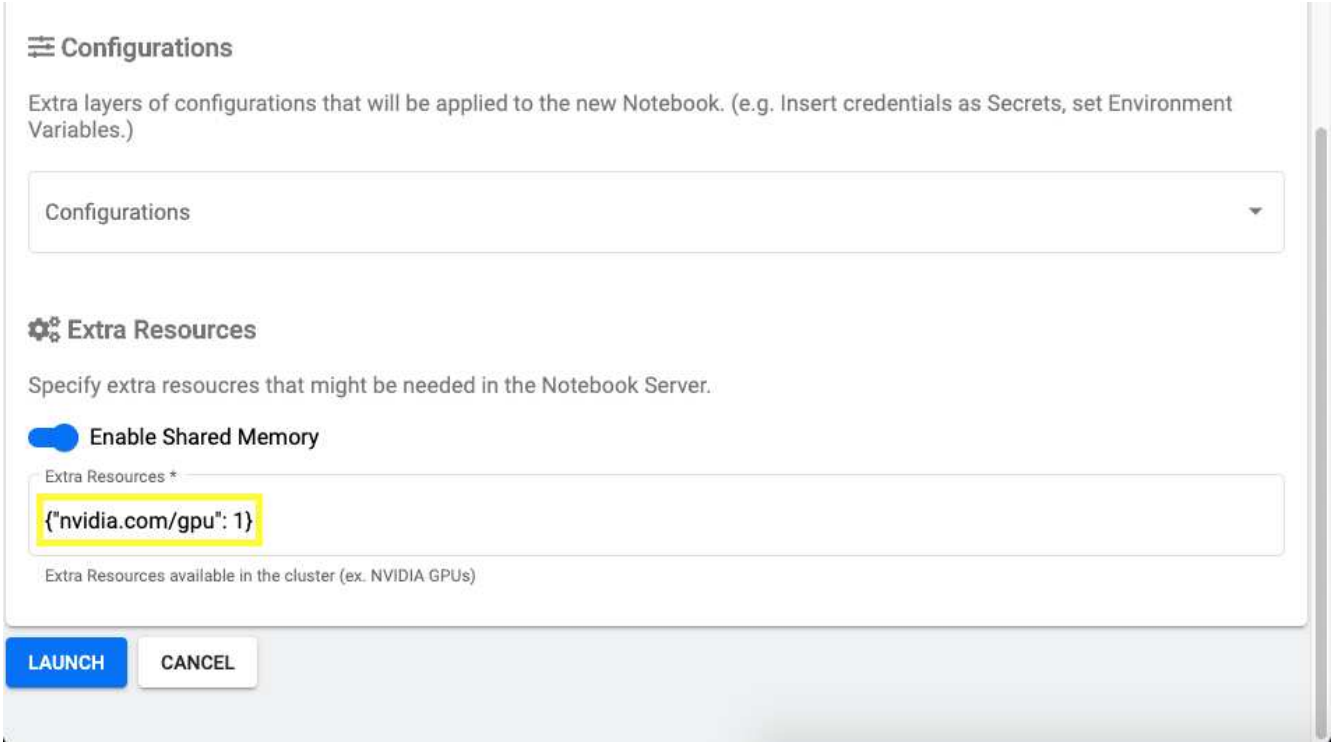
4. Specificare i dettagli del volume dello spazio di lavoro. Se si sceglie di creare un nuovo volume, il provisioning di tale volume o PVC viene eseguito utilizzando la classe di storage predefinita. Perché un StorageClass che utilizza Trident è stato designato come StorageClass predefinito nella sezione "Implementazione di Kubeflow", Il volume o PVC viene fornito con Trident. Questo volume viene montato automaticamente come area di lavoro predefinita all'interno del container Jupyter notebook Server. Tutti i notebook creati dall'utente sul server che non vengono salvati in un volume di dati separato vengono salvati automaticamente in questo volume di spazio di lavoro. Pertanto, i notebook sono persistenti durante i riavvii.



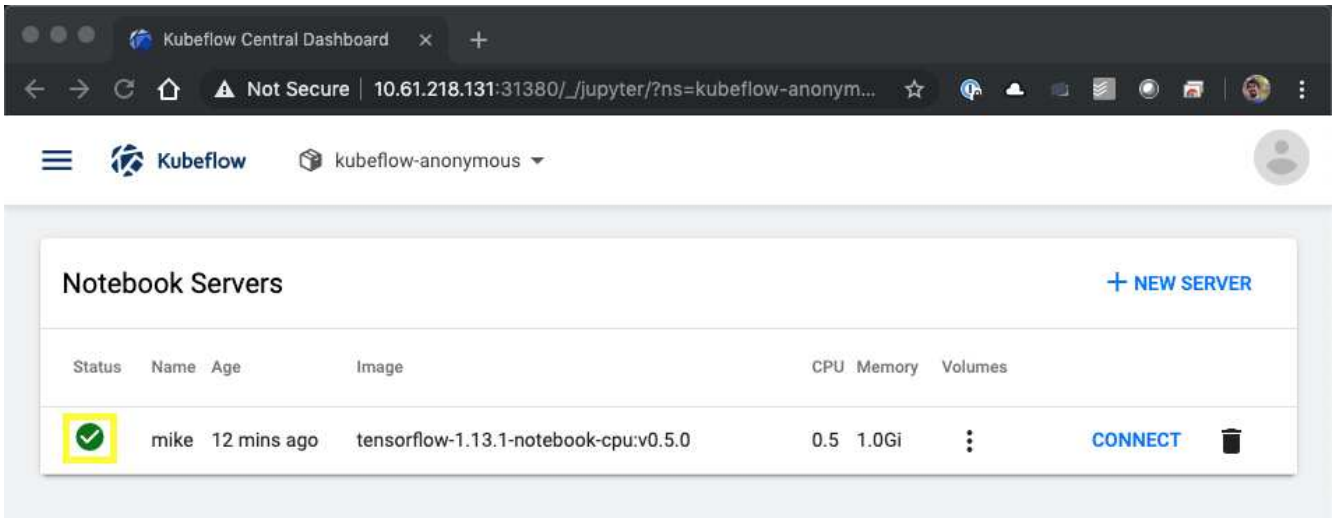
5. Aggiungere volumi di dati. Nell'esempio seguente viene specificato un PVC esistente denominato 'pb-fg-all' e viene accettato il punto di montaggio predefinito.



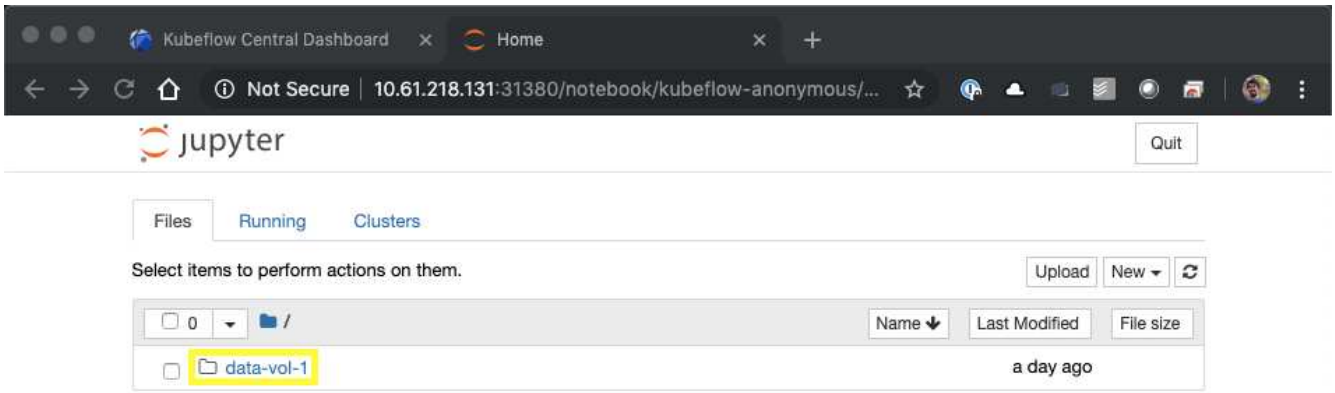
6. **Opzionale:** richiedere l'allocazione del numero desiderato di GPU al notebook server. Nell'esempio seguente, viene richiesta una GPU.

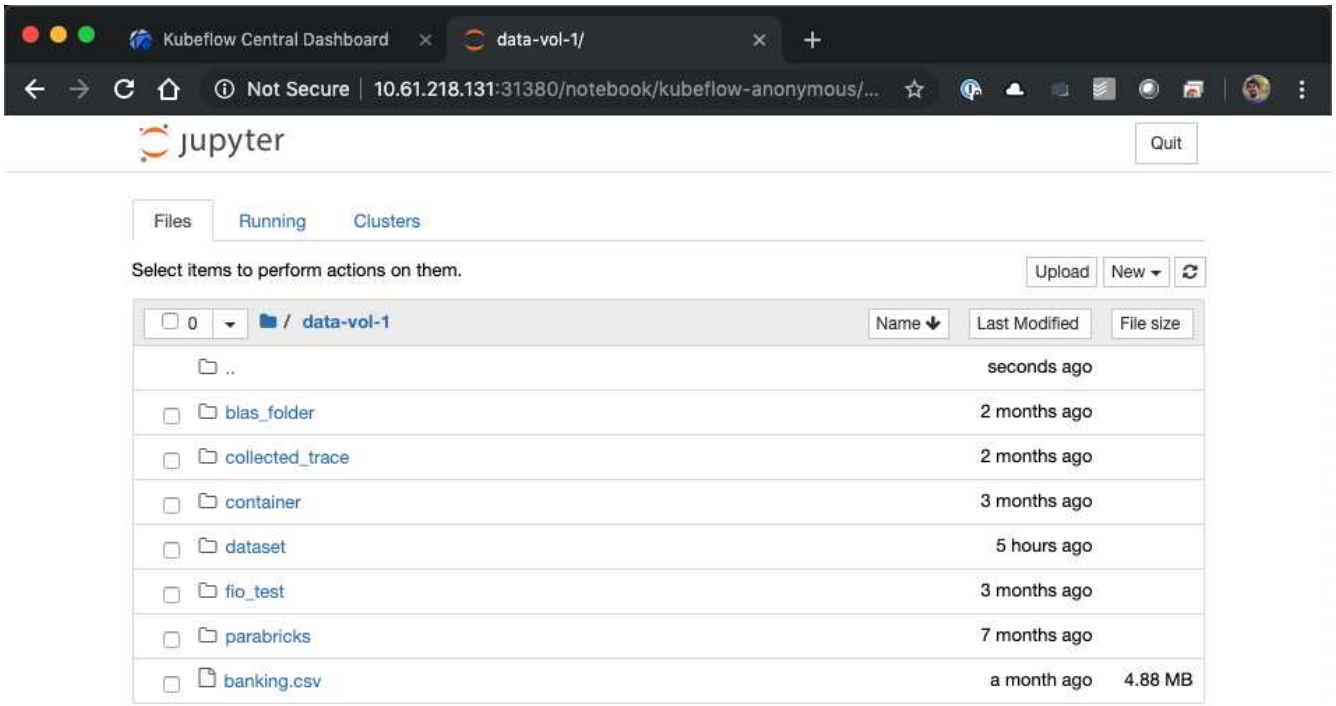


7. Fare clic su Launch (Avvia) per eseguire il provisioning del nuovo notebook server.
8. Attendere il provisioning completo del server notebook. Questa operazione può richiedere alcuni minuti se non si è mai eseguito il provisioning di un server utilizzando l'immagine Docker specificata, in quanto l'immagine deve essere scaricata. Una volta completato il provisioning del server, viene visualizzato un segno di spunta verde nella colonna Status (Stato) della pagina di amministrazione del server Jupyter notebook.



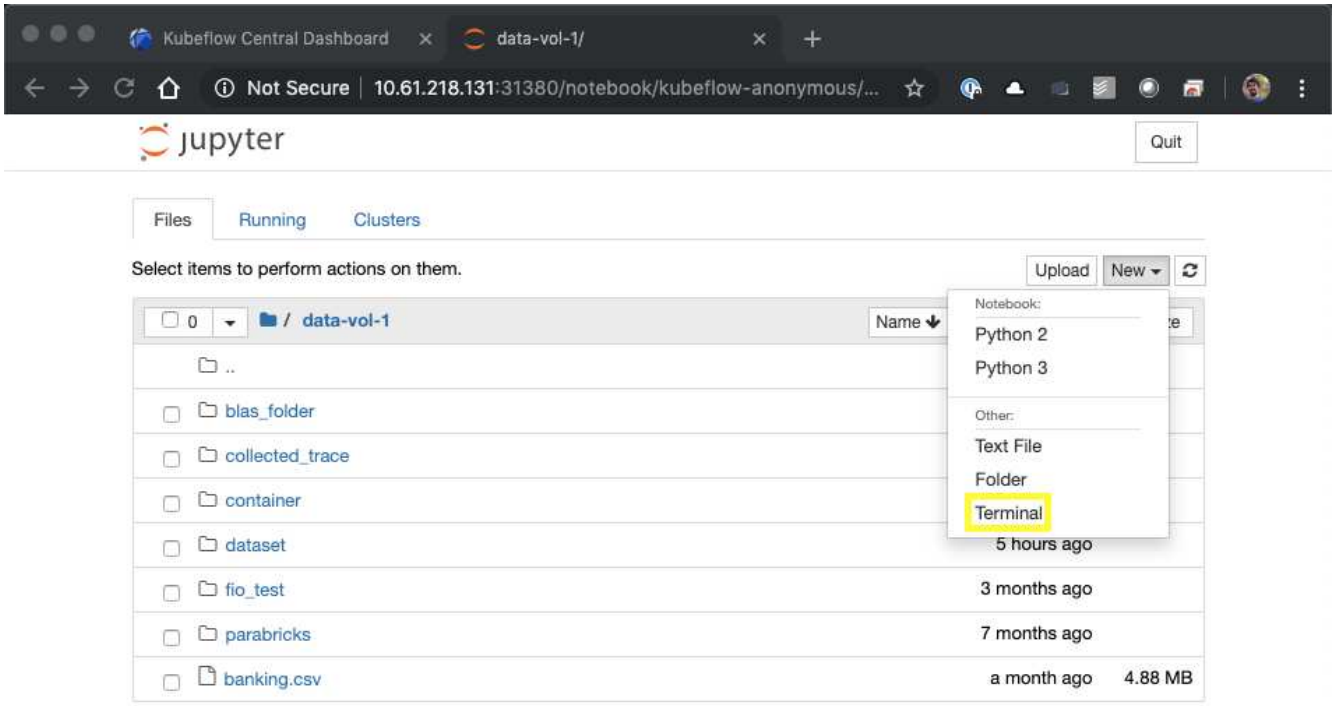
9. Fare clic su Connect (Connetti) per connettersi alla nuova interfaccia Web del server.
10. Verificare che il volume del set di dati specificato al punto 6 sia montato sul server. Si noti che questo volume viene montato nell'area di lavoro predefinita per impostazione predefinita. Dal punto di vista dell'utente, questa è solo un'altra cartella all'interno dello spazio di lavoro. L'utente, che è probabilmente un data scientist e non un esperto di infrastruttura, non deve possedere alcuna esperienza di storage per utilizzare questo volume.

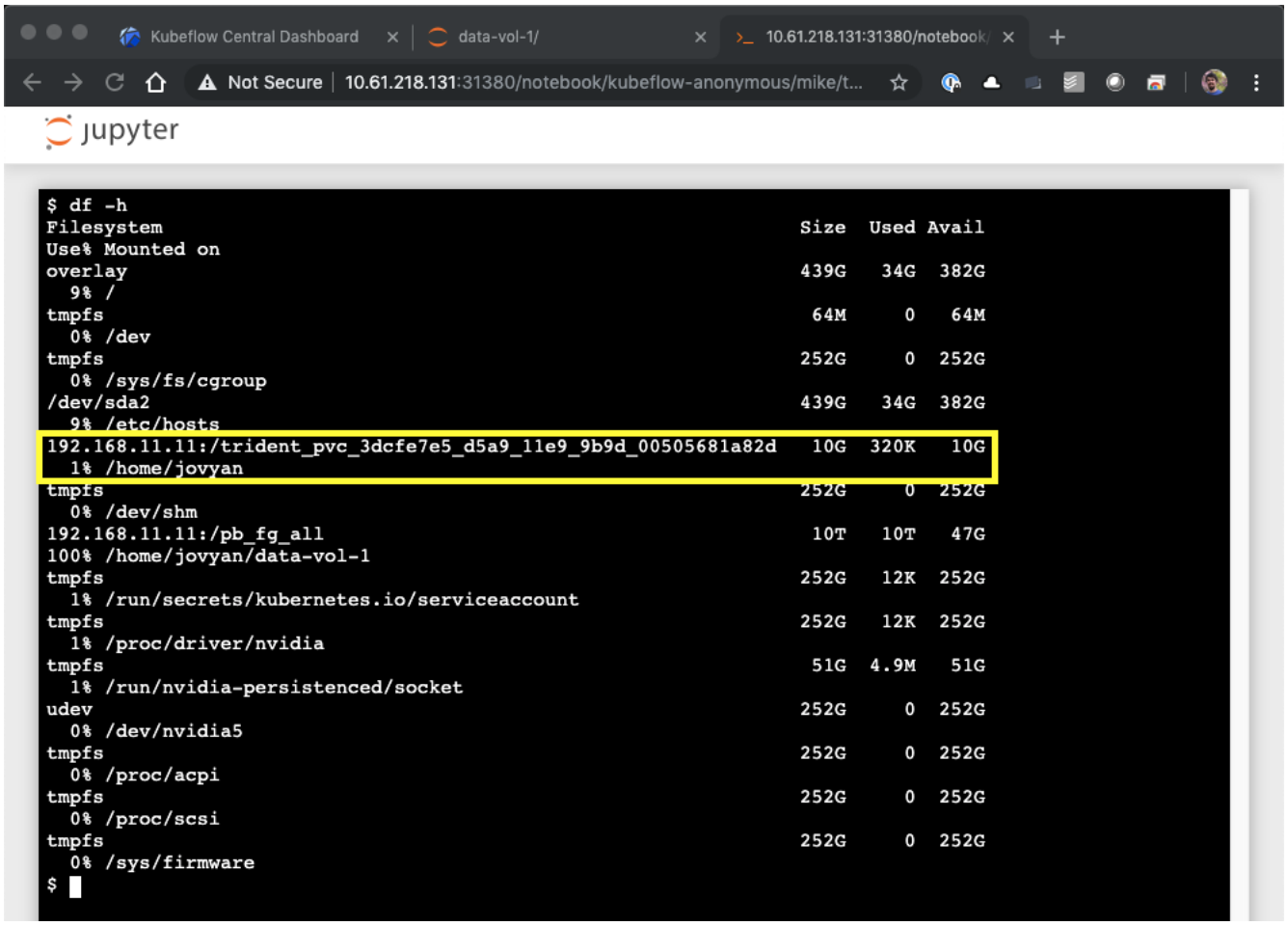




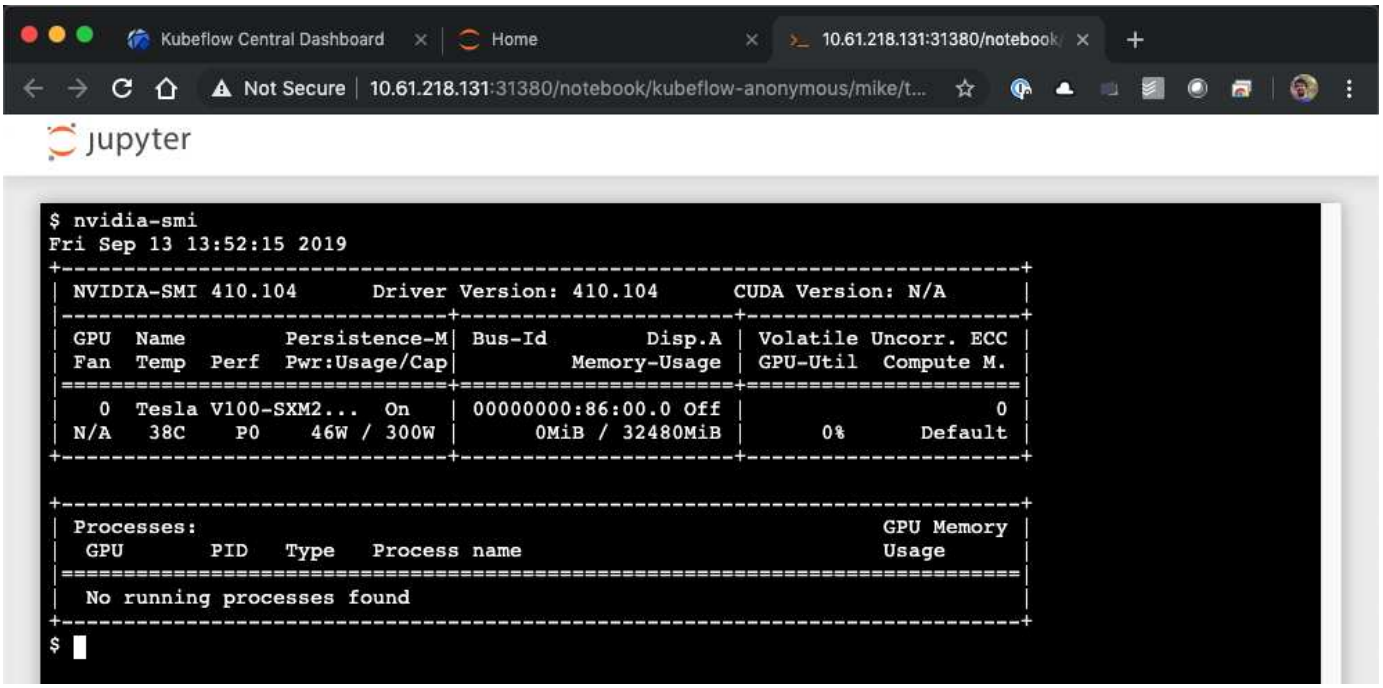
11. Aprire un terminale e, supponendo che sia stato richiesto un nuovo volume nel passaggio 5, eseguire `df -h` Per confermare che un nuovo volume persistente con provisioning Trident è montato come area di lavoro predefinita.

    La directory predefinita dello spazio di lavoro è la directory di base che viene visualizzata quando si accede per la prima volta all'interfaccia Web del server. Pertanto, tutti gli artefatti creati utilizzando l'interfaccia Web vengono memorizzati su questo volume persistente con provisioning Trident.





12. Utilizzando il terminale, eseguire `nvidia-smi` Per confermare che il numero corretto di GPU è stato allocato al notebook server. Nell'esempio seguente, una GPU è stata allocata al notebook server come richiesto nel passaggio 7.



Esempi di notebook e pipeline

Il "NetApp Data Science Toolkit per Kubernetes" Utilizzabile in combinazione con Kubeflow. L'utilizzo del NetApp Data Science Toolkit con Kubeflow offre i seguenti vantaggi:

- I data scientist possono eseguire operazioni avanzate di gestione dei dati NetApp direttamente da un Jupyter notebook.
- Le operazioni avanzate di gestione dei dati NetApp possono essere incorporate nei flussi di lavoro automatizzati utilizzando il framework Kubeflow Pipeline.

Fare riferimento a. "Esempi di Kubeflow" Sezione all'interno del repository GitHub del NetApp Data Science Toolkit per informazioni dettagliate sull'utilizzo del toolkit con Kubeflow.

Implementazione di Apache Airflow

NetApp consiglia di eseguire Apache Airflow su Kubernetes. Questa sezione descrive le attività da completare per implementare il flusso d'aria nel cluster Kubernetes.

> È possibile implementare il flusso d'aria su piattaforme diverse da Kubernetes. L'implementazione del flusso d'aria su piattaforme diverse da Kubernetes non rientra nell'ambito di questa soluzione.

Prerequisiti

Prima di eseguire l'esercizio di implementazione descritto in questa sezione, si presuppone che siano già state eseguite le seguenti attività:

1. Hai già un cluster Kubernetes funzionante.
2. NetApp Trident è già stato installato e configurato nel cluster Kubernetes, come descritto nella sezione "implementazione e configurazione di NetApp Trident".

Installare Helm

Il flusso d'aria viene implementato utilizzando Helm, un popolare gestore di pacchetti per Kubernetes. Prima di implementare il flusso d'aria, è necessario installare Helm sull'host di distribuzione jump. Per installare Helm sull'host di distribuzione jump, seguire la "istruzioni per l'installazione" Nella documentazione ufficiale di Helm.

Impostare la classe di storage Kubernetes predefinita

Prima di implementare il flusso d'aria, è necessario specificare un StorageClass predefinito all'interno del cluster Kubernetes. Il processo di implementazione del flusso d'aria tenta di eseguire il provisioning di nuovi volumi persistenti utilizzando la classe di storage predefinita. Se non viene indicato StorageClass come StorageClass predefinito, l'implementazione non riesce. Per designare una StorageClass predefinita all'interno del cluster, seguire le istruzioni riportate nella sezione "Implementazione di Kubeflow". Se è già stata designata una StorageClass predefinita all'interno del cluster, è possibile saltare questo passaggio.

USA Helm per implementare il flusso d'aria

Per implementare il flusso d'aria nel cluster Kubernetes utilizzando Helm, eseguire le seguenti operazioni dall'host di distribuzione jump:

1. Implementare il flusso d'aria utilizzando Helm seguendo il "istruzioni per l'implementazione" Per il diagramma ufficiale del flusso d'aria sull'Artifact Hub. I comandi di esempio che seguono mostrano l'implementazione del flusso d'aria con Helm. Modificare, aggiungere e/o rimuovere i valori in `custom-values.yaml` file in base alle necessità, a seconda dell'ambiente e della configurazione desiderata.

```
$ 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
```

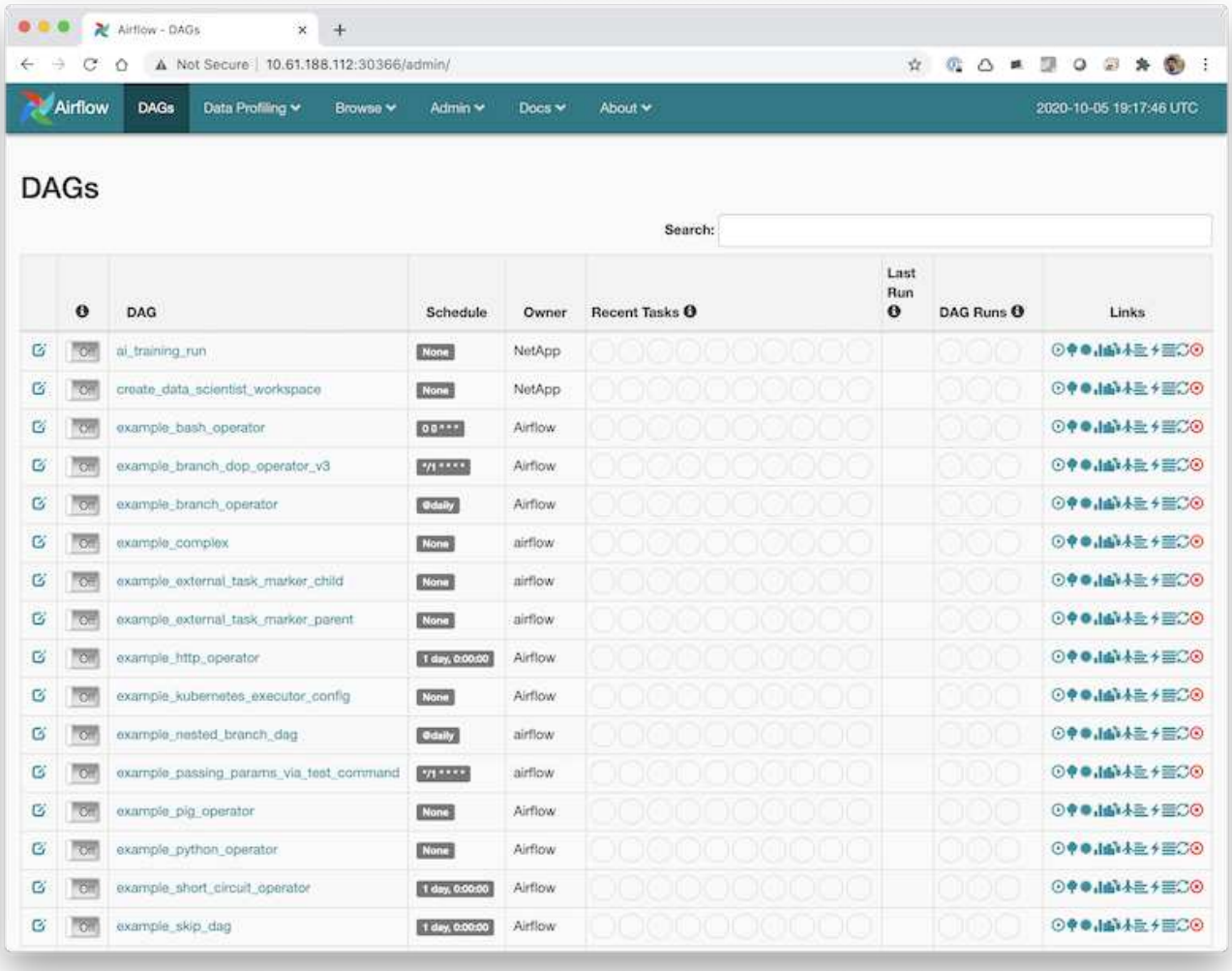
2. Verificare che tutti i pod del flusso d'aria siano in funzione. L'avvio di tutti i pod potrebbe richiedere alcuni minuti.

```
$ 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. Ottenere l'URL del servizio Web Airflow seguendo le istruzioni stampate sulla console quando si implementa Airflow utilizzando Helm nel passaggio 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. Verificare che sia possibile accedere al servizio Web Airflow.



Esempio di flussi di lavoro Apache Airflow

Il "NetApp Data Science Toolkit per Kubernetes" Utilizzabile in combinazione con il flusso d'aria. L'utilizzo del NetApp Data Science Toolkit con Airflow consente di incorporare le operazioni di gestione dei dati NetApp in flussi di lavoro automatizzati orchestrati dal flusso d'aria.

Fare riferimento a. "Esempi di flusso d'aria" Sezione all'interno del repository GitHub del NetApp Data Science Toolkit per informazioni dettagliate sull'utilizzo del toolkit con flusso d'aria.

Esempio di operazioni Trident

Questa sezione include esempi di varie operazioni che è possibile eseguire con Trident.

Importare un volume esistente

Se nel sistema/piattaforma di storage NetApp sono presenti volumi che si desidera montare su container all'interno del cluster Kubernetes, ma che non sono legati ai PVC nel cluster, è necessario importare questi volumi. È possibile utilizzare la funzionalità di importazione dei volumi Trident per importare questi volumi.

I comandi di esempio seguenti mostrano l'importazione dello stesso volume, denominato `pb_fg_all`, Due volte, una per ogni backend Trident creato nell'esempio nella sezione "Esempi di backend Trident per implementazioni ai ONTAP", fase 1. L'importazione dello stesso volume due volte in questo modo consente di montare il volume (un volume FlexGroup esistente) più volte su diverse LIF, come descritto nella sezione "Esempi di backend Trident per implementazioni ai ONTAP", fase 1. Per ulteriori informazioni sui PVC, vedere "Documentazione ufficiale di Kubernetes". Per ulteriori informazioni sulla funzionalità di importazione dei volumi, vedere "Documentazione di Trident".

An `accessModes` valore di `ReadOnlyMany` È specificato nei file delle specifiche PVC di esempio. Per ulteriori informazioni su `accessMode` vedere il campo "Documentazione ufficiale di Kubernetes".

> I nomi di backend specificati nei comandi di importazione di esempio riportati di seguito corrispondono ai backend creati nell'esempio della sezione "Esempi di backend Trident per implementazioni ai ONTAP", fase 1. I nomi StorageClass specificati nei seguenti file di definizione PVC di esempio corrispondono ai StorageClasses creati nell'esempio nella sezione "Esempi di storage Classes Kubernetes per implementazioni ai ONTAP", fase 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
```

Provisioning di un nuovo volume

È possibile utilizzare Trident per eseguire il provisioning di un nuovo volume sul sistema o sulla piattaforma di storage NetApp. I seguenti comandi di esempio mostrano il provisioning di un nuovo volume FlexVol. In questo esempio, il provisioning del volume viene eseguito utilizzando StorageClass creato nell'esempio della sezione "Esempi di storage Classes Kubernetes per implementazioni ai ONTAP", punto 2.

An `accessModes` valore di `ReadWriteMany` Viene specificato nel seguente file di definizione PVC di esempio. Per ulteriori informazioni su `accessMode` vedere il campo "Documentazione ufficiale di 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
```

Esempi di opportunità di lavoro ad alte performance per le implementazioni ai di ONTAP

Questa sezione include esempi di vari job dalle performance elevate che possono essere eseguiti quando Kubernetes viene implementato su un pod ai ONTAP.

Esempi di opportunità di lavoro ad alte performance per le implementazioni ai di ONTAP

Questa sezione include esempi di vari job dalle performance elevate che possono essere eseguiti quando Kubernetes viene implementato su un pod ai ONTAP.

Eseguire un carico di lavoro ai a nodo singolo

Per eseguire un processo ai e ML a nodo singolo nel cluster Kubernetes, eseguire le seguenti operazioni dall'host di distribuzione jump. Con Trident, è possibile rendere un volume di dati, potenzialmente contenente petabyte di dati, accessibile a un carico di lavoro Kubernetes in modo rapido e semplice. Per rendere un volume di dati accessibile dall'interno di un pod Kubernetes, è sufficiente specificare un PVC nella definizione del pod. Si tratta di un'operazione nativa di Kubernetes, senza richiedere alcuna esperienza

> In questa sezione si presuppone che sia già stato containerizzato (nel formato Docker Container) il carico di lavoro ai e ML specifico che si sta tentando di eseguire nel cluster Kubernetes.

1. I seguenti comandi di esempio mostrano la creazione di un lavoro Kubernetes per un carico di lavoro di benchmark TensorFlow che utilizza il dataset ImageNet. Per ulteriori informazioni sul set di dati ImageNet, vedere "Sito Web ImageNet".

   Questo processo di esempio richiede otto GPU e quindi può essere eseguito su un singolo nodo di lavoro GPU che dispone di otto o più GPU. Questo job di esempio potrebbe essere inviato in un cluster per il quale un nodo di lavoro con otto o più GPU non è presente o è attualmente occupato con un altro workload. In tal caso, il lavoro rimane in uno stato in sospeso fino a quando tale nodo di lavoro non diventa disponibile.

   Inoltre, per massimizzare la larghezza di banda dello storage, il volume contenente i dati di training necessari viene montato due volte all'interno del pod creato da questo lavoro. Nel pod è montato anche un altro volume. Questo secondo volume verrà utilizzato per memorizzare risultati e metriche. Questi volumi vengono referenziati nella definizione del lavoro utilizzando i nomi dei PVC. Per ulteriori informazioni sui job Kubernetes, consultare "Documentazione ufficiale di Kubernetes".

   An `emptyDir` volume con a. `medium` valore di `Memory` è montato su `/dev/shm` nel pod creato da questo lavoro di esempio. La dimensione predefinita di `/dev/shm` Il volume virtuale creato automaticamente dal runtime del container Docker può talvolta essere insufficiente per le esigenze di TensorFlow. Montaggio di un `emptyDir` il volume come nell'esempio seguente fornisce un volume sufficientemente grande `/dev/shm` volume virtuale. Per ulteriori informazioni su `emptyDir` volumes (volumi), vedere "Documentazione ufficiale di Kubernetes".

   Al singolo contenitore specificato in questa definizione di lavoro di esempio viene assegnato un `securityContext > privileged` valore di `true`. Questo valore significa che il container dispone effettivamente dell'accesso root sull'host. Questa annotazione viene utilizzata in questo caso perché il carico di lavoro specifico che viene eseguito richiede l'accesso root. In particolare, un'operazione di cancellazione della cache eseguita dal carico di lavoro richiede l'accesso root. Che sia o meno così `privileged: true` l'annotazione è necessaria a seconda dei requisiti del carico di lavoro specifico che si sta eseguendo.

```
$ 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. Verificare che il lavoro creato al punto 1 sia in esecuzione correttamente. Il seguente comando di esempio conferma che è stato creato un singolo pod per il lavoro, come specificato nella definizione del lavoro, e che questo pod è attualmente in esecuzione su uno dei nodi di lavoro 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. Verificare che il lavoro creato al passo 1 sia stato completato correttamente. I seguenti comandi di esempio confermano che il lavoro è stato completato correttamente.

```
$ 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. **Opzionale:** eliminare gli artefatti del lavoro. I seguenti comandi di esempio mostrano l'eliminazione dell'oggetto di lavoro creato al passo 1.

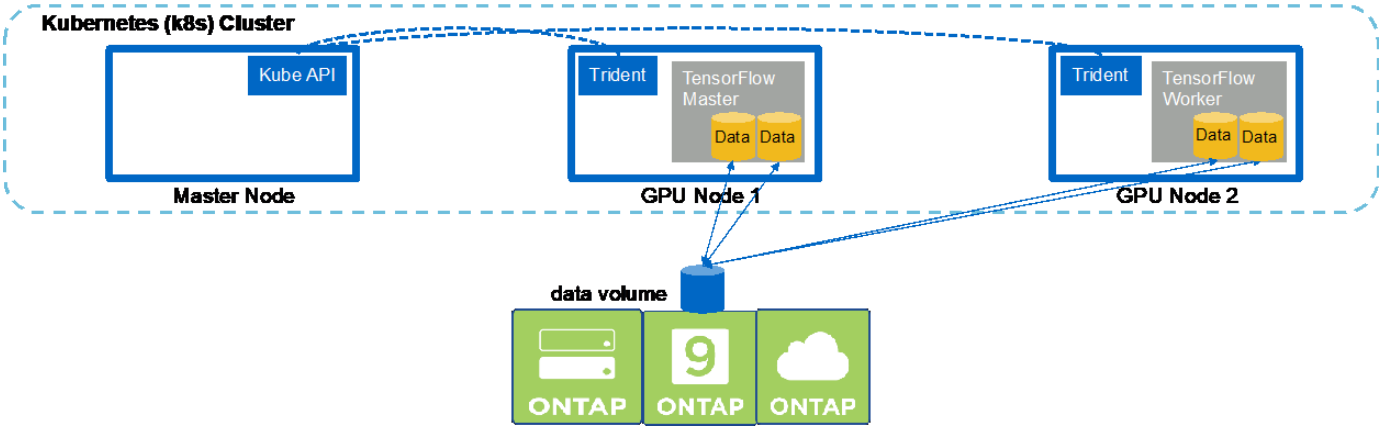
   Quando si elimina l'oggetto di lavoro, Kubernetes elimina automaticamente tutti i pod associati.

```
$ 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.
```

Eseguire un carico di lavoro ai distribuito sincrono

Per eseguire un processo ai e ML multinodo sincrono nel cluster Kubernetes, eseguire le seguenti operazioni sull'host di distribuzione jump. Questo processo consente di sfruttare i dati memorizzati su un volume NetApp e di utilizzare più GPU di quelle che un singolo nodo di lavoro può fornire. Vedere la figura seguente per un'illustrazione di un lavoro di ai distribuito sincrono.

> I lavori distribuiti sincroni possono contribuire ad aumentare la precisione delle performance e della formazione rispetto ai lavori distribuiti asincroni. Una discussione sui pro e contro dei lavori sincroni rispetto ai lavori asincroni non rientra nell'ambito di questo documento.



1. I seguenti comandi di esempio mostrano la creazione di un worker che partecipa all'esecuzione distribuita sincrona dello stesso job di benchmark TensorFlow eseguito su un singolo nodo nell'esempio della sezione "Eseguire un carico di lavoro ai a nodo singolo". In questo esempio specifico, viene implementato solo un singolo worker perché il lavoro viene eseguito su due nodi di lavoro.

In questo esempio, l'implementazione di lavoro richiede otto GPU e può quindi essere eseguita su un singolo nodo di lavoro GPU che dispone di otto o più GPU. Se i nodi di lavoro GPU dispongono di più di otto GPU, per massimizzare le performance, è possibile aumentare questo numero in modo da essere uguale al numero di GPU presenti nei nodi di lavoro. Per ulteriori informazioni sulle implementazioni di Kubernetes, vedere "Documentazione ufficiale di Kubernetes".

In questo esempio viene creata un'implementazione di Kubernetes perché questo specifico lavoratore containerizzato non viene mai completato da solo. Pertanto, non ha senso implementarlo utilizzando il costrutto di lavoro Kubernetes. Se il tuo lavoratore è stato progettato o scritto per essere completato da solo, potrebbe essere opportuno utilizzare il costrutto di lavoro per implementare il tuo lavoratore.

Al pod specificato in questa specifica di implementazione di esempio viene assegnato un `hostNetwork` valore di `true`. Questo valore significa che il pod utilizza lo stack di rete del nodo di lavoro host invece dello stack di rete virtuale creato da Kubernetes per ciascun pod. Questa annotazione viene utilizzata in questo caso perché il carico di lavoro specifico si basa su Open MPI, NCCL e Horovod per eseguire il carico di lavoro in maniera sincrona e distribuita. Pertanto, richiede l'accesso allo stack di rete host. Una discussione su Open MPI, NCCL e Horovod non rientra nell'ambito di questo documento. Che sia o meno così `hostNetwork: true` l'annotazione è necessaria a seconda dei requisiti del carico di lavoro specifico che si sta eseguendo. Per ulteriori informazioni su `hostNetwork` vedere il campo "Documentazione ufficiale di 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. Verificare che l'implementazione worker creata al punto 1 sia stata avviata correttamente. I seguenti comandi di esempio confermano che è stato creato un singolo pod di lavoro per l'implementazione, come indicato nella definizione di implementazione, e che questo pod è attualmente in esecuzione su uno dei nodi di lavoro 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. Creare un lavoro Kubernetes per un master che inizia, partecipa e tiene traccia dell'esecuzione del lavoro sincrono a più nodi. I seguenti comandi di esempio creano un master che inizia, partecipa e tiene traccia dell'esecuzione distribuita sincrona dello stesso job di benchmark TensorFlow eseguito su un singolo nodo

nell'esempio nella sezione "Eseguire un carico di lavoro ai a nodo singolo".

Questo processo master di esempio richiede otto GPU e può quindi essere eseguito su un singolo nodo di lavoro GPU che dispone di otto o più GPU. Se i nodi di lavoro GPU dispongono di più di otto GPU, per massimizzare le performance, è possibile aumentare questo numero in modo da essere uguale al numero di GPU presenti nei nodi di lavoro.

Al pod master specificato in questa definizione di lavoro di esempio viene assegnato un `hostNetwork` valore di `true`, proprio come al pod di lavoro è stato assegnato un `hostNetwork` valore di `true` nella fase 1. Per ulteriori informazioni sul motivo per cui questo valore è necessario, vedere il passaggio 1.

```
$ 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. Verificare che il lavoro principale creato al punto 3 sia in esecuzione correttamente. Il seguente comando di esempio conferma che è stato creato un singolo pod master per il lavoro, come indicato nella definizione del lavoro, e che questo pod è attualmente in esecuzione su uno dei nodi di lavoro GPU. Inoltre, il pod di lavoro inizialmente visto al punto 1 è ancora in esecuzione e i pod master e di lavoro sono in esecuzione su nodi diversi.

```
$ 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. Verificare che il lavoro principale creato al punto 3 sia stato completato correttamente. I seguenti comandi di esempio confermano che il lavoro è stato completato correttamente.

```
$ 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. Eliminare l'implementazione dei lavoratori quando non è più necessaria. I seguenti comandi di esempio mostrano l'eliminazione dell'oggetto di implementazione worker creato nel passaggio 1.

   Quando si elimina l'oggetto di implementazione worker, Kubernetes elimina automaticamente tutti i worker pod associati.

```
$ 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. **Opzionale:** eliminare gli artefatti del job master. I seguenti comandi di esempio mostrano l'eliminazione dell'oggetto di lavoro master creato nel passaggio 3.

   Quando si elimina l'oggetto di lavoro master, Kubernetes elimina automaticamente tutti i pod master associati.

```
$ 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.
```

Test delle performance

Come parte della creazione di questa soluzione, abbiamo eseguito un semplice confronto delle performance. Utilizzando Kubernetes, abbiamo eseguito diversi processi di benchmarking ai standard di NetApp e abbiamo confrontato i risultati del benchmark con le esecuzioni eseguite utilizzando un semplice comando di esecuzione di Docker. Non sono state riscontrate differenze significative in termini di performance. Pertanto, abbiamo concluso che l'utilizzo di Kubernetes per orchestrare i lavori di training ai containerizzati non influisce negativamente sulle performance. Consulta la tabella seguente per i risultati del nostro confronto delle performance.

Benchmark	Dataset	Esecuzione Docker (immagini/sec)	Kubernetes (immagini/sec)
TensorFlow a nodo singolo	Dati sintetici	6,667.2475	6,661.93125
TensorFlow a nodo singolo	ImageNet	6,570.2025	6,530.59125
Synchronous Distributed Two-Node TensorFlow	Dati sintetici	13,213.70625	13,218.288125
Synchronous Distributed Two-Node TensorFlow	ImageNet	12,941.69125	12,881.33875

Conclusione

Aziende e organizzazioni di tutte le dimensioni e in tutti i settori stanno passando all'intelligenza artificiale (ai), all'apprendimento automatico (ML) e al deep learning (DL) per risolvere problemi reali, offrire prodotti e servizi innovativi e ottenere un vantaggio in un mercato sempre più competitivo. Man mano che le organizzazioni aumentano l'utilizzo di ai, ML e DL, devono affrontare molte sfide, tra cui la scalabilità dei workload e la disponibilità dei dati. Queste sfide possono essere affrontate utilizzando la soluzione NetApp ai Control Plane.

Questa soluzione consente di clonare rapidamente uno spazio dei nomi dei dati. Inoltre, consente di definire e implementare flussi di lavoro di training ai, ML e DL che incorporano la creazione quasi istantanea di dati e linee di base dei modelli per la tracciabilità e il controllo delle versioni. Con questa soluzione, è possibile tracciare ogni singolo modello di training fino ai set di dati esatti con cui il modello è stato addestrato e/o validato. Infine, questa soluzione consente di eseguire rapidamente il provisioning degli spazi di lavoro dei notebook Jupyter con accesso a set di dati di grandi dimensioni.

Poiché questa soluzione è rivolta a data scientist e data engineer, è necessaria una competenza minima di NetApp o NetApp ONTAP. Con questa soluzione, le funzioni di gestione dei dati possono essere eseguite utilizzando interfacce e strumenti semplici e familiari. Inoltre, questa soluzione utilizza componenti completamente open-source e liberi. Pertanto, se disponete già di storage NetApp nel vostro ambiente, potete implementare questa soluzione oggi stesso. Se si desidera provare questa soluzione ma non si dispone già di storage NetApp, visitare il sito "cloud.netapp.com"E potrai essere operativo con una soluzione di storage NetApp basata sul cloud in pochissimo tempo.

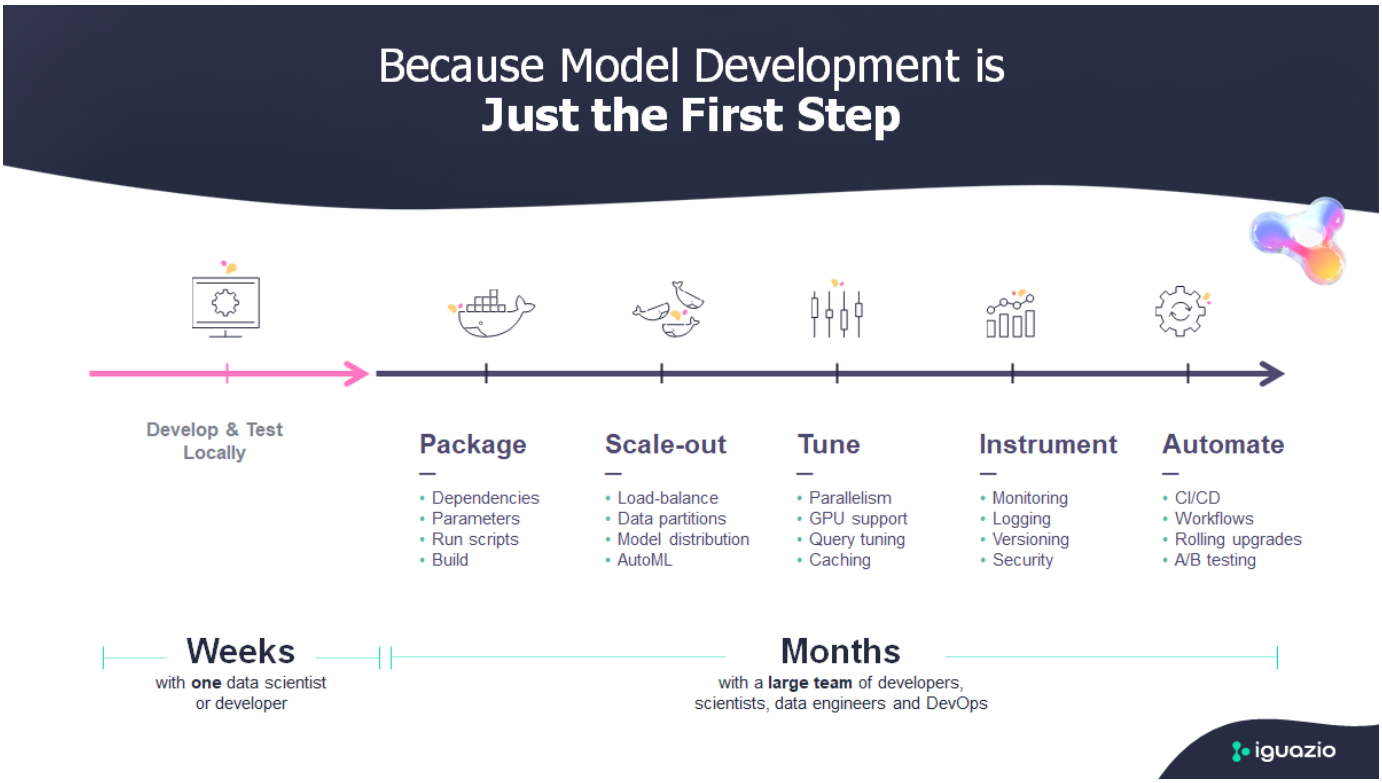
Pipeline MLRun con Iguazio

TR-4834: Pipeline NetApp e Iguazio per MLRun

Rick Huang, David Arnette, NetApp Marcelo Litovsky, Iguazio

Questo documento illustra i dettagli della pipeline MLRun che utilizza NetApp ONTAP ai, NetApp ai Control Plane, il software NetApp Cloud Volumes e la piattaforma per la scienza dei dati Iguazio. Abbiamo utilizzato la funzione senza server di Nuclio, i volumi persistenti di Kubernetes, i volumi cloud di NetApp, le copie Snapshot di NetApp, la dashboard di Grafana, E altri servizi sulla piattaforma Iguazio per creare una pipeline di dati end-to-end per la simulazione del rilevamento dei guasti di rete. Abbiamo integrato le tecnologie Iguazio e NetApp per consentire un'implementazione rapida dei modelli, la replica dei dati e le funzionalità di monitoraggio della produzione on-premise e nel cloud.

Il lavoro di un data scientist dovrebbe essere incentrato sulla formazione e sulla messa a punto di modelli di apprendimento automatico (ML) e intelligenza artificiale (ai). Tuttavia, secondo una ricerca condotta da Google, i data scientist trascorrono ~il 80% del loro tempo a capire come far funzionare i propri modelli con le applicazioni aziendali ed eseguirlo su larga scala, come mostrato nella seguente immagine che illustra lo sviluppo di modelli nel workflow ai/ML.



Per gestire i progetti ai/ML end-to-end, è necessaria una maggiore comprensione dei componenti aziendali. Sebbene DevOps abbia assunto il controllo della definizione, dell'integrazione e dell'implementazione di questi tipi di componenti, le operazioni di apprendimento automatico hanno come obiettivo un flusso simile che include progetti ai/ML. Per avere un'idea di ciò che una pipeline ai/ML end-to-end tocca nell'azienda, consulta il seguente elenco di componenti richiesti:

- Storage

- Networking
- Database
- File system
- Container
- Pipeline ci/CD (Continuous Integration and Continuous Deployment)
- Ambiente di sviluppo integrato (IDE)
- Sicurezza
- Policy di accesso ai dati
- Hardware
- Cloud
- Virtualizzazione
- Set di strumenti e librerie per le scienze dei dati

In questo documento, dimostreremo come la partnership tra NetApp e Iguazio semplifichi drasticamente lo sviluppo di una pipeline ai/ML end-to-end. Questa semplificazione accelera il time-to-market per tutte le applicazioni ai/ML.

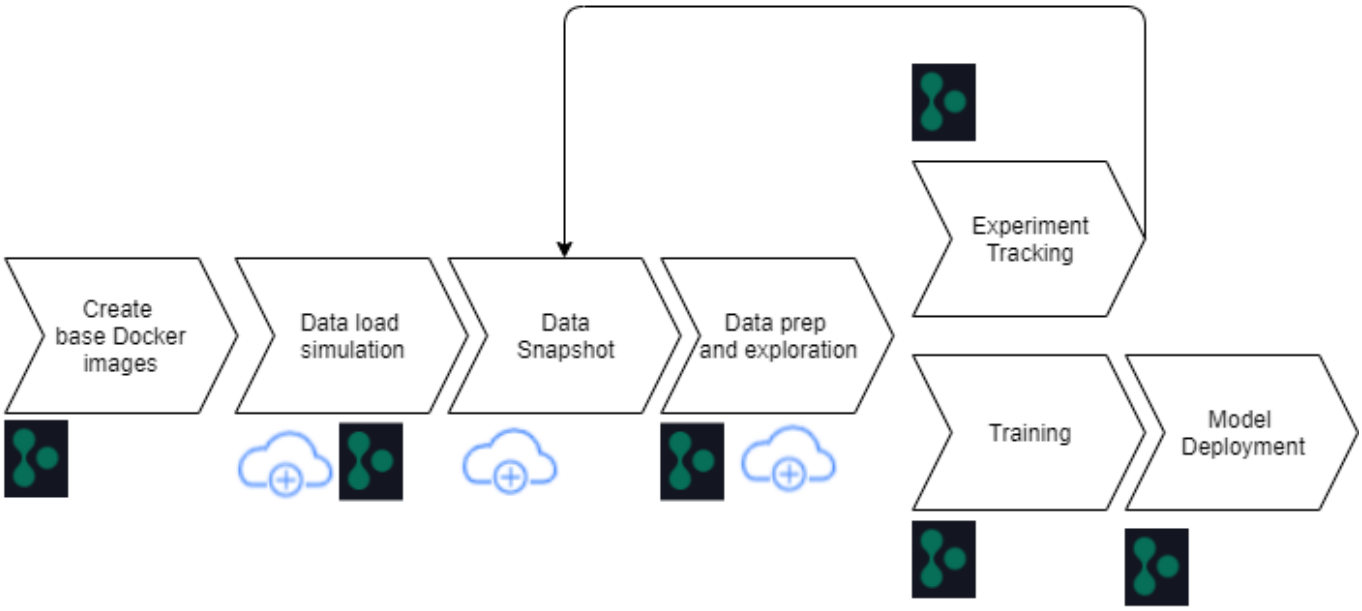
Pubblico di destinazione

Il mondo della scienza dei dati tocca diverse discipline nel settore dell'informatica e del business.

- Il data scientist ha bisogno della flessibilità necessaria per utilizzare i propri strumenti e le librerie preferite.
- Il data engineer deve sapere come i dati scorrono e dove risiedono.
- Un tecnico DevOps ha bisogno dei tool per integrare le nuove applicazioni ai/ML nelle pipeline ci/CD.
- Gli utenti aziendali desiderano avere accesso alle applicazioni ai/ML. Descriviamo in che modo NetApp e Iguazio aiutano ciascuno di questi ruoli a portare valore al business con le nostre piattaforme.

Panoramica della soluzione

Questa soluzione segue il ciclo di vita di un'applicazione ai/ML. Iniziamo con il lavoro dei data scientist per definire le diverse fasi necessarie per preparare i dati e formare e implementare i modelli. Seguiamo il lavoro necessario per creare una pipeline completa con la capacità di tenere traccia degli artefatti, sperimentare con l'esecuzione e implementare in Kubeflow. Per completare il ciclo completo, integriamo la pipeline con i volumi cloud di NetApp per abilitare il controllo delle versioni dei dati, come mostrato nell'immagine seguente.



iguazio

Cloud Volumes ONTAP

Panoramica sulla tecnologia

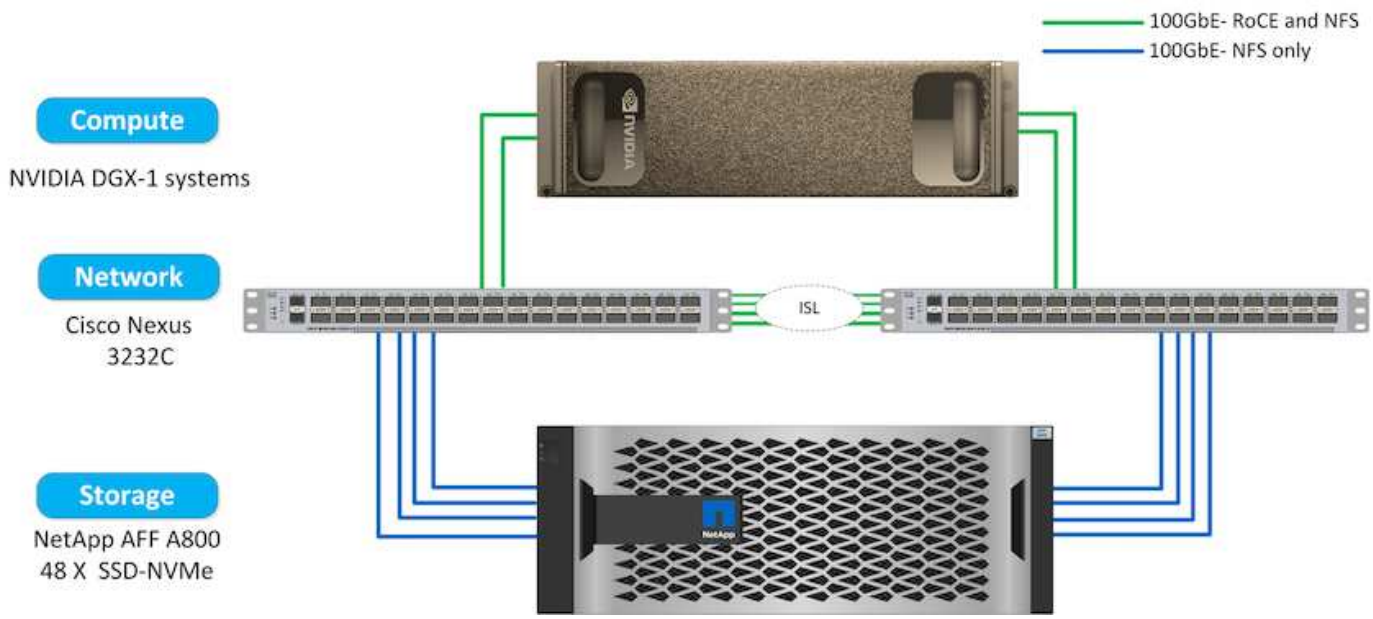
Panoramica di NetApp

NetApp è l'autorità dei dati per il cloud ibrido. NetApp offre una gamma completa di servizi dati del cloud ibrido che semplificano la gestione di applicazioni e dati in ambienti cloud e on-premise per accelerare la trasformazione digitale. Insieme ai nostri partner, NetApp consente alle organizzazioni globali di liberare il pieno potenziale dei propri dati per espandere i punti di contatto dei clienti, promuovere una maggiore innovazione e ottimizzare le operazioni.

NetApp ONTAP ai

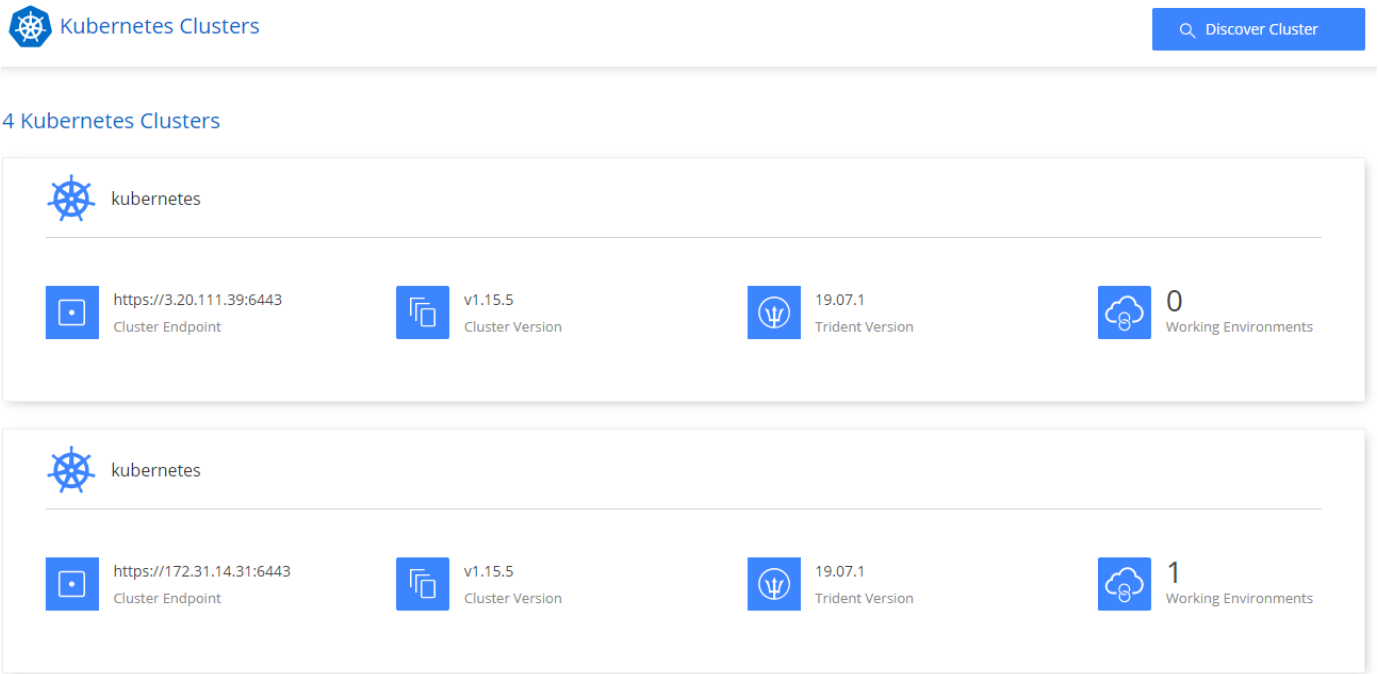
NetApp ONTAP ai, basato su sistemi NVIDIA DGX e storage all-flash connesso al cloud, ottimizza il flusso dei dati in modo affidabile e accelera l'analisi, la formazione e l'inferenza con il data fabric che si estende dall'edge al core al cloud. Offre alle organizzazioni IT un'architettura che offre i seguenti vantaggi:

- Elimina le complessità di progettazione
- Consente una scalabilità indipendente di calcolo e storage
- Consente ai clienti di partire da piccoli e scalare perfettamente
- Offre una vasta gamma di opzioni di storage per diverse performance e costi. NetApp ONTAP ai offre stack di infrastruttura convergente che incorporano NVIDIA DGX-1, un sistema ai petaflop-scale e switch Ethernet NVIDIA Mellanox dalle performance elevate per unificare i carichi di lavoro ai, semplificare l'implementazione e accelerare il ROI. Abbiamo utilizzato ONTAP ai con un sistema storage DGX-1 e NetApp AFF A800 per questo report tecnico. La seguente immagine mostra la topologia di ONTAP ai con il sistema DGX-1 utilizzato per questa convalida.



Piano di controllo ai di NetApp

Il NetApp ai Control Plane ti consente di liberare ai e ML con una soluzione che offre scalabilità estrema, implementazione semplificata e disponibilità dei dati continua. La soluzione ai Control Plane integra Kubernetes e Kubeflow con un data fabric abilitato da NetApp. Kubernetes, la piattaforma di orchestrazione dei container standard di settore per le implementazioni native del cloud, offre scalabilità e portabilità dei workload. Kubeflow è una piattaforma open-source per l'apprendimento automatico che semplifica la gestione e l'implementazione, consentendo agli sviluppatori di fare più scienza dei dati in meno tempo. Un data fabric abilitato da NetApp offre disponibilità e portabilità dei dati senza compromessi per garantire che i dati siano accessibili attraverso la pipeline, dall'edge al core al cloud. Questo report tecnico utilizza il NetApp ai Control Plane in una pipeline MLRun. L'immagine seguente mostra la pagina di gestione del cluster Kubernetes, in cui è possibile avere endpoint diversi per ciascun cluster. Abbiamo collegato i volumi persistenti NFS al cluster Kubernetes e le immagini seguenti mostrano un volume persistente connesso al cluster, dove "Trident di NetApp" offre supporto dello storage persistente e funzionalità di gestione dei dati.



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)*

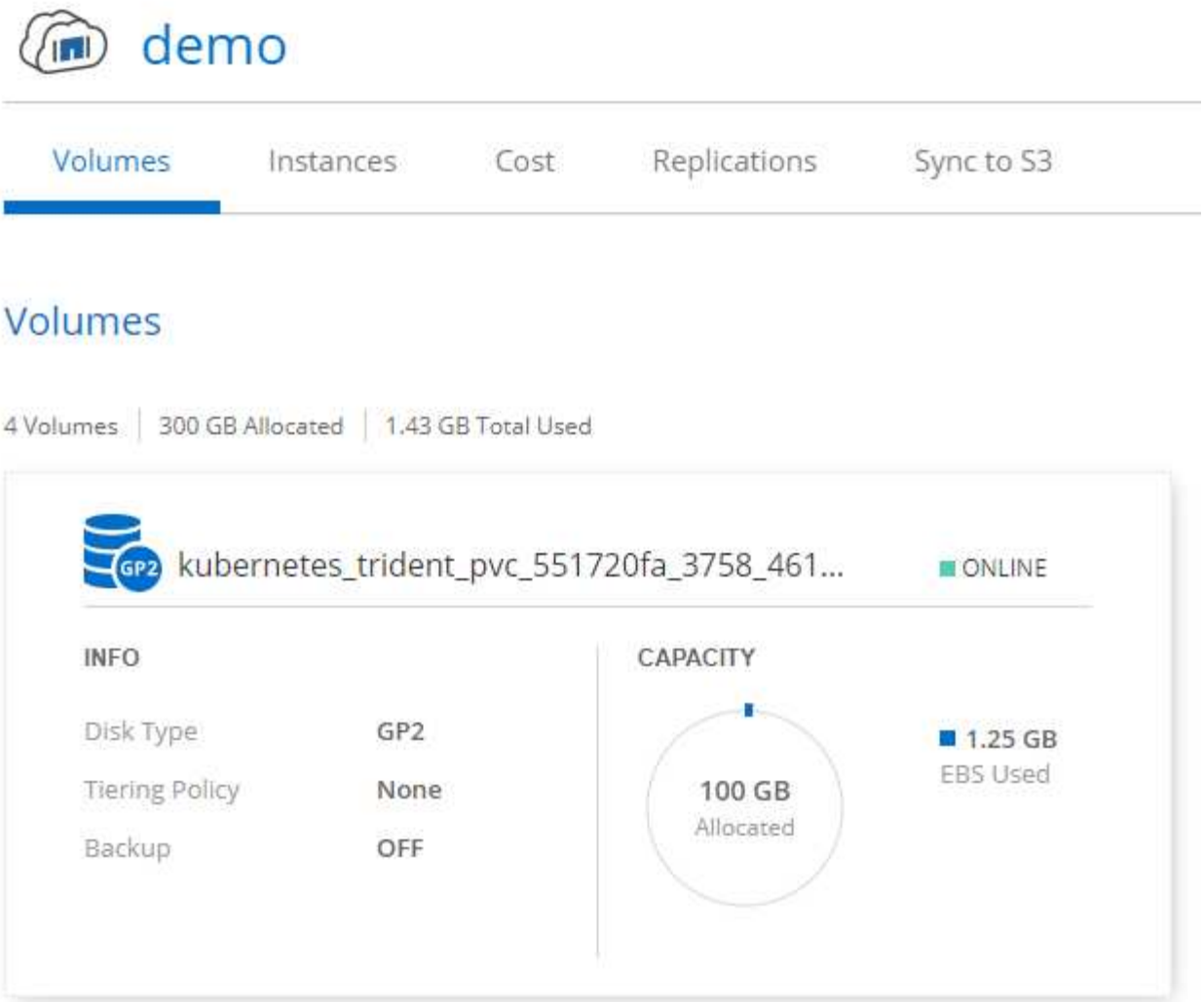
Custom Export Policy

172.31.0.0/16

☑ Set as default storage class

◉ NFS ○ iSCSI

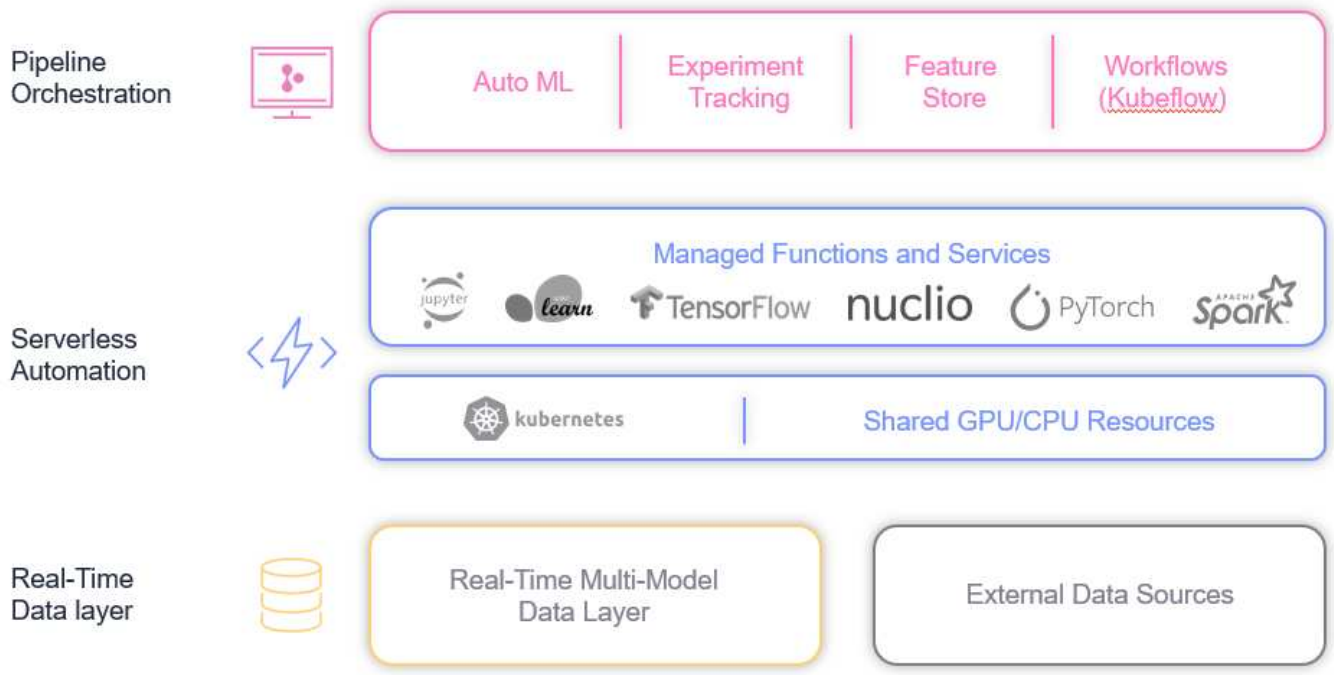
Connect | Cancel



Panoramica di Iguazio

Iguazio Data Science Platform è una piattaforma PaaS (Data Science Platform as a Service) completamente integrata e sicura che semplifica lo sviluppo, accelera le performance, facilita la collaborazione e affronta le sfide operative. Questa piattaforma incorpora i seguenti componenti e la piattaforma Iguazio Data Science è presentata nella seguente immagine:

- Un workbench per la scienza dei dati che include notebook Jupyter, motori di analisi integrati e pacchetti Python
- Gestione dei modelli con monitoraggio degli esperimenti e funzionalità di pipeline automatizzate
- Dati gestiti e SERVIZI ML su un cluster Kubernetes scalabile
- Nuclio, un framework di funzioni senza server in tempo reale
- Un livello di dati estremamente veloce e sicuro che supporta SQL, NoSQL, database Time-Series, file (oggetti semplici) e streaming
- Integrazione con origini dati di terze parti come NetApp, Amazon S3, HDFS, database SQL e protocolli di streaming o messaggistica
- Dashboard in tempo reale basati su Grafana



Requisiti software e hardware

Configurazione di rete

Di seguito è riportato il requisito di configurazione di rete per l'impostazione nel cloud:

- Il cluster Iguazio e i volumi NetApp Cloud devono trovarsi nello stesso cloud privato virtuale.
- Il cloud manager deve avere accesso alla porta 6443 sui nodi dell'applicazione Iguazio.
- Abbiamo utilizzato Amazon Web Services in questo report tecnico. Tuttavia, gli utenti hanno la possibilità di implementare la soluzione in qualsiasi provider cloud.per i test on-premise in ONTAP ai con NVIDIA DGX-1, abbiamo utilizzato il servizio DNS in hosting Iguazio per comodità.

I client devono essere in grado di accedere ai domini DNS creati dinamicamente. Se lo si desidera, i clienti possono utilizzare il proprio DNS.

Requisiti hardware

È possibile installare Iguazio on-premise nel proprio cluster. Abbiamo verificato la soluzione in NetApp ONTAP ai con un sistema NVIDIA DGX-1. La seguente tabella elenca l'hardware utilizzato per testare questa soluzione.

Hardware	Quantità
Sistemi DGX-1	1
Sistema NetApp AFF A800	1 coppia ad alta disponibilità (ha), include 2 controller e 48 SSD NVMe (3,8 TB o superiore)
Switch di rete Cisco Nexus 3232C	2

La seguente tabella elenca i componenti software necessari per il test on-premise:

Software	Versione o altre informazioni
Software per la gestione dei dati NetApp ONTAP	9.7
Firmware dello switch Cisco NX-OS	7.0(3)I6(1)
SISTEMA OPERATIVO NVIDIA DGX	4.4 - Ubuntu 18.04 LTS
Piattaforma container Docker	19.03.5
Versione container	20.01-tf1-py2
Framework per l'apprendimento automatico	TensorFlow 1.15.0
Iguazio	Versione 2.8+
Server ESX	6.5

Questa soluzione è stata completamente testata con Iguazio versione 2.5 e NetApp Cloud Volumes ONTAP per AWS. Il cluster Iguazio e il software NetApp sono entrambi in esecuzione su AWS.

Software	Versione o tipo
Iguazio	Versione 2.8+
Nodo app	M5.4xGrande
Nodo dati	I3,4 x grande

Riepilogo dei casi d'utilizzo della previsione di guasto dei dispositivi di rete

Questo caso di utilizzo si basa su un cliente Iguazio nello spazio delle telecomunicazioni in Asia. Con 100.000 clienti Enterprise e 125.000 eventi di interruzione della rete all'anno, era fondamentale prevedere e intraprendere azioni proattive per evitare che i guasti della rete influenzino i clienti. Questa soluzione ha fornito loro i seguenti vantaggi:

- Analisi predittiva dei guasti di rete
- Integrazione con un sistema di ticketing
- Intraprendere azioni proattive per prevenire i guasti di rete come risultato di questa implementazione di Iguazio, il 60% dei guasti è stato prevenuto in modo proattivo.

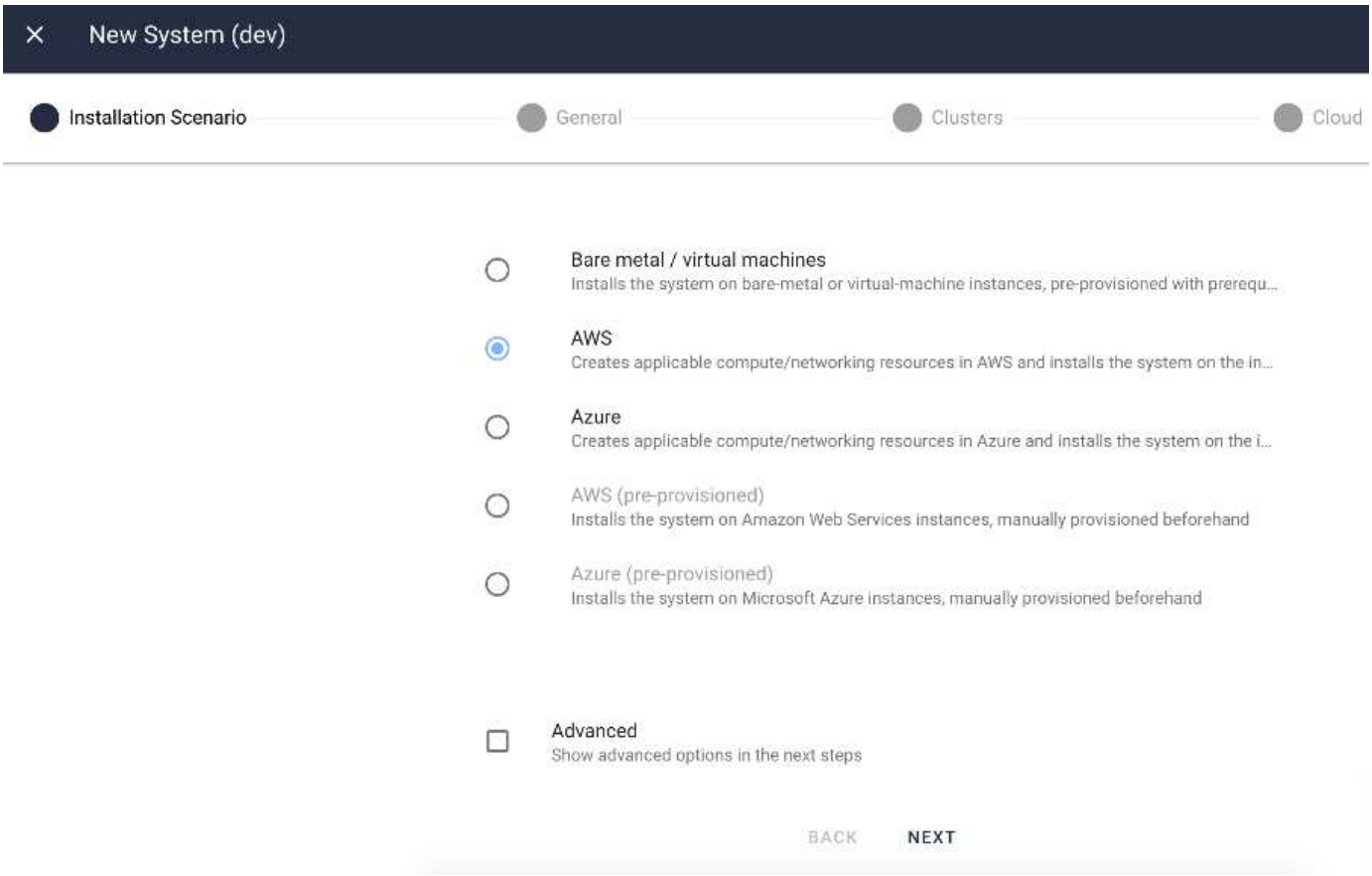
Panoramica dell'installazione

Iguazio può essere installato on-premise o su un cloud provider.

Installazione di Iguazio

Il provisioning può essere eseguito come servizio e gestito da Iguazio o dal cliente. In entrambi i casi, Iguazio fornisce un'applicazione di implementazione (Provazio) per implementare e gestire i cluster.

Per l'installazione on-premise, fare riferimento a. "NVA-1121" per la configurazione di calcolo, rete e storage. L'implementazione on-premise di Iguazio è fornita da Iguazio senza costi aggiuntivi per il cliente. Vedere "questa pagina" Per le configurazioni dei server DNS e SMTP. La pagina di installazione di Provazio viene visualizzata come segue.



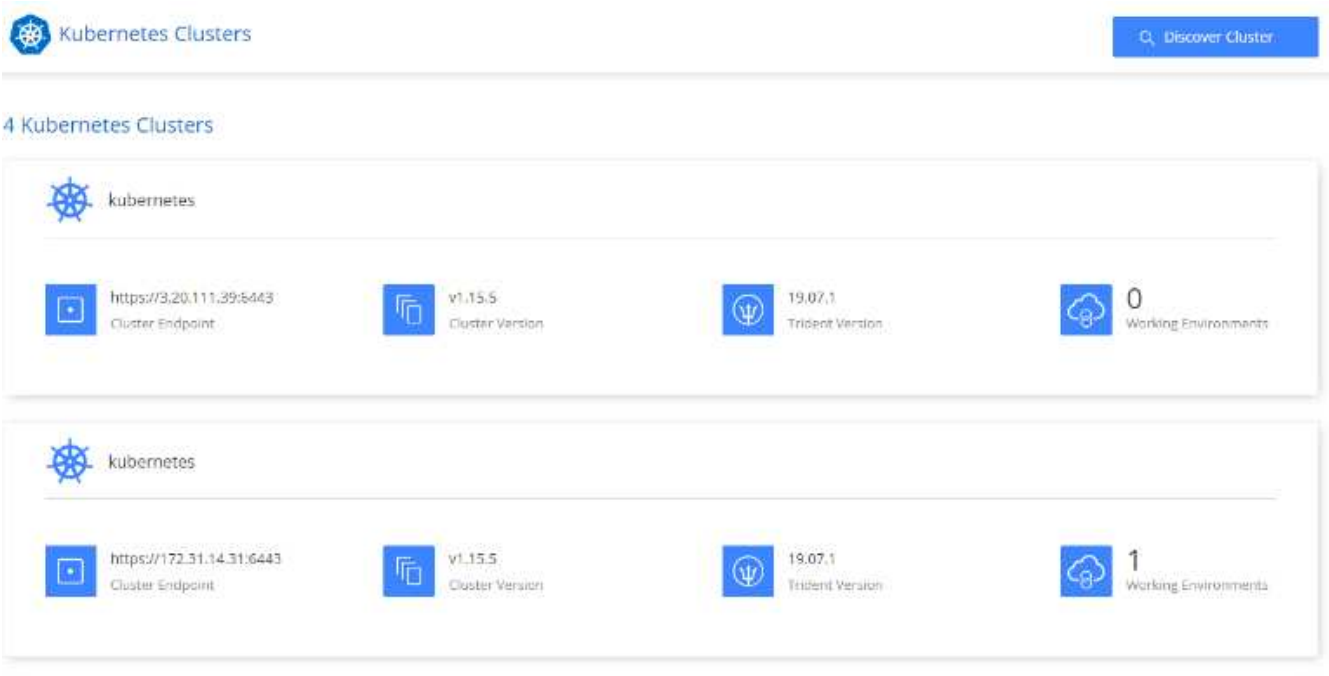
Configurazione del cluster Kubernetes

Questa sezione è suddivisa in due parti, rispettivamente per l'implementazione in cloud e on-premise.

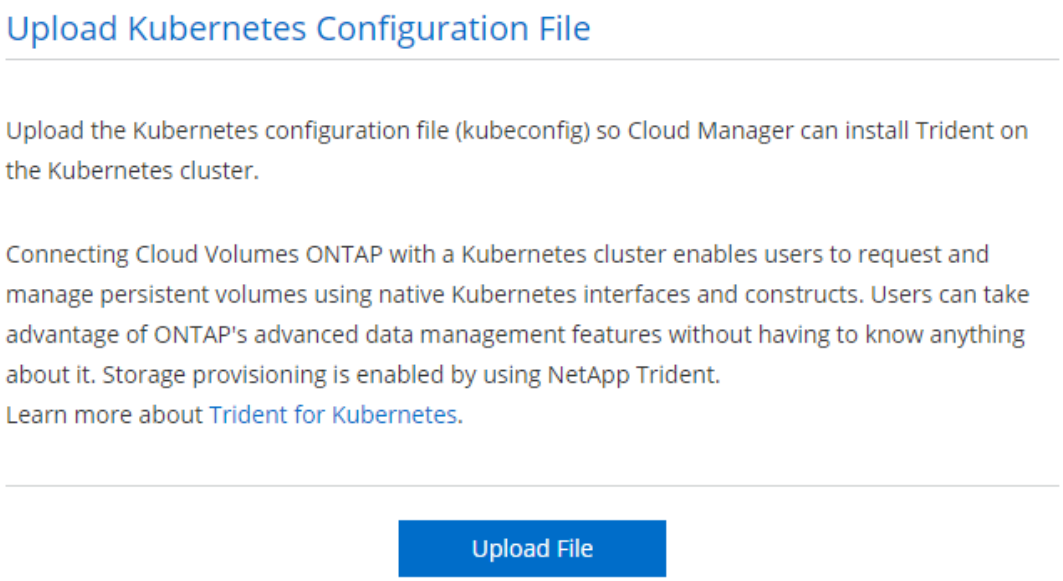
Configurazione di Kubernetes per l'implementazione del cloud

Tramite NetApp Cloud Manager, è possibile definire la connessione al cluster Iguazio Kubernetes. Trident richiede l'accesso a più risorse nel cluster per rendere disponibile il volume.

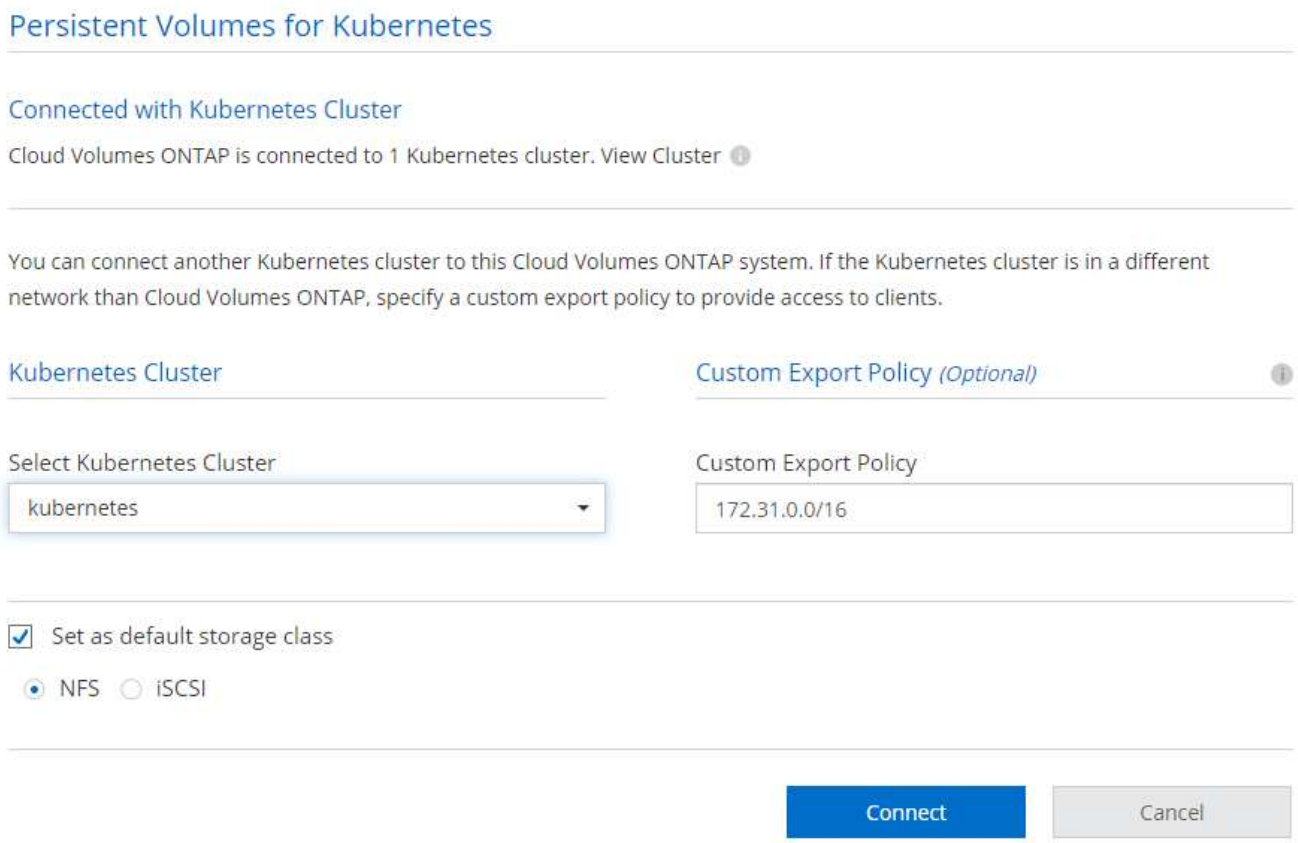
1. Per abilitare l'accesso, ottenere il file di configurazione di Kubernetes da uno dei nodi Iguazio. Il file si trova in `/home/Iguazio/.kube/config`. Scaricare questo file sul desktop.
2. Accedere a Discover Cluster (rilevamento cluster) per eseguire la configurazione.



3. Caricare il file di configurazione di Kubernetes. Vedere la seguente immagine.



4. Implementare Trident e associare un volume al cluster. Vedere la seguente immagine per definire e assegnare un volume persistente al cluster Iguazio. Questo processo crea un volume persistente (PV) nel cluster Kubernetes di Iguazio. Prima di poterlo utilizzare, è necessario definire un PVC (Persistent Volume Claim).



Configurazione di Kubernetes per l'implementazione on-premise

Per l'installazione on-premise di NetApp Trident, vedere "TR-4798" per ulteriori informazioni. Dopo aver configurato il cluster Kubernetes e aver installato NetApp Trident, è possibile collegare Trident al cluster Iguazio per abilitare le funzionalità di gestione dei dati NetApp, come l'acquisizione di copie Snapshot dei dati e del modello.

Definisci richiesta di rimborso per volumi persistenti

1. Salvare il seguente YAML in un file per creare un PVC di tipo Basic.

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

2. Applicare il file YAML al cluster Iguazio Kubernetes.

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

Collega il volume NetApp al notebook Jupyter

Iguazio offre diversi servizi gestiti per fornire ai data scientist uno stack end-to-end completo per lo sviluppo e l'implementazione di applicazioni ai/ML. Per ulteriori informazioni su questi componenti, consultare la sezione "Iguazio Panoramica dei servizi e degli strumenti applicativi".

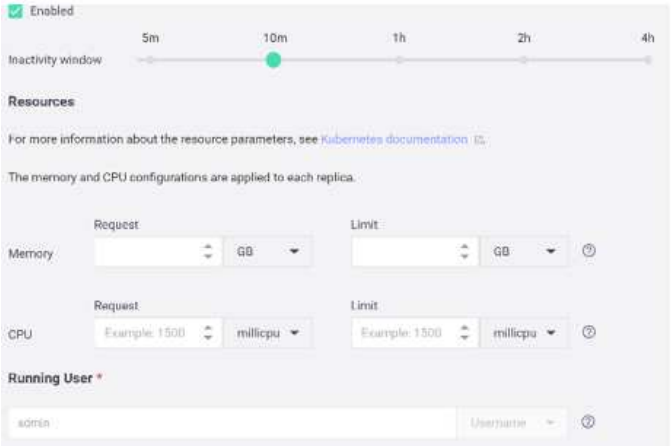
Uno dei servizi gestiti è Jupyter notebook. Ogni sviluppatore ottiene la propria implementazione di un container di notebook con le risorse necessarie per lo sviluppo. Per consentire loro l'accesso al NetApp Cloud Volume, è possibile assegnare il volume al relativo container e allocazione delle risorse, utente in esecuzione e impostazioni delle variabili di ambiente per le richieste di rimborso dei volumi persistenti sono presentate nella seguente immagine.

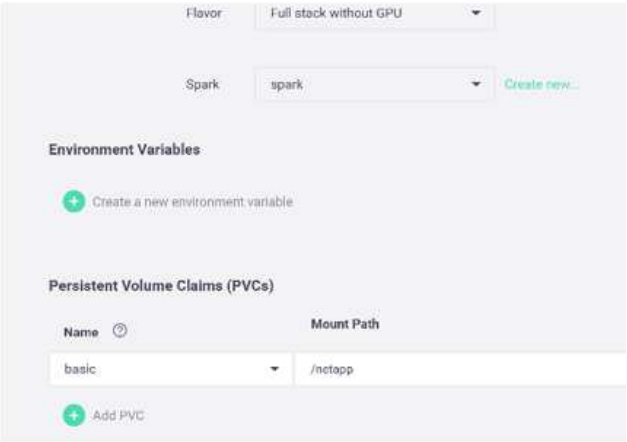
Per una configurazione on-premise, fare riferimento a. "TR-4798" Nella configurazione di Trident per abilitare le funzionalità di gestione dei dati di NetApp ONTAP, come l'acquisizione di copie Snapshot dei dati o del modello per il controllo delle versioni. Aggiungere la seguente riga nel file di configurazione back-end di Trident per rendere visibili le directory Snapshot:

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

È necessario creare un file di configurazione back-end Trident in formato JSON, quindi eseguire quanto segue "Comando Trident" come riferimento:

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





Implementazione dell'applicazione

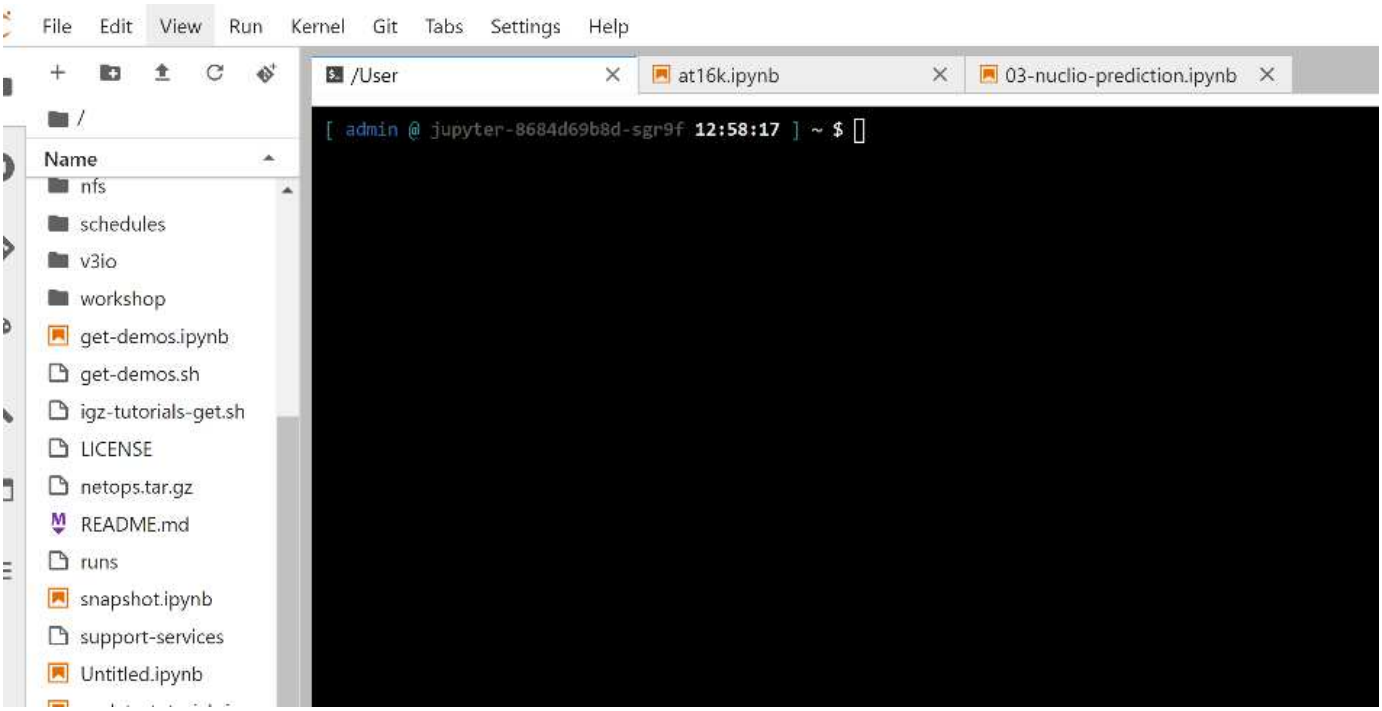
Le sezioni seguenti descrivono come installare e implementare l'applicazione.

Ottieni codice da GitHub

Ora che il volume NetApp Cloud o NetApp Trident è disponibile per il cluster Iguazio e l'ambiente di sviluppo, è possibile iniziare a rivedere l'applicazione.

Gli utenti dispongono di un proprio spazio di lavoro (directory). Su ogni notebook, il percorso alla directory dell'utente è `/User`. La piattaforma Iguazio gestisce la directory. Se si seguono le istruzioni riportate sopra, il volume NetApp Cloud è disponibile in `/netapp` directory.

Ottieni il codice da GitHub utilizzando un terminale Jupyter.



Al prompt del terminale Jupyter, clonare il progetto.

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

A questo punto, viene visualizzata la `netops- netapp` Nella struttura dei file nell'area di lavoro Jupyter.

Configurare l'ambiente di lavoro

Copiare il `Notebook` `set_env-Example.ipynb` come `set_env.ipynb`. Aprire e modificare `set_env.ipynb`. Questo notebook imposta le variabili per le credenziali, le posizioni dei file e i driver di esecuzione.

Se si seguono le istruzioni riportate in precedenza, le uniche modifiche da apportare sono le seguenti:

1. Ottieni questo valore dalla dashboard dei servizi Iguazio: `docker_registry`

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

2. Cambiare `admin` Al tuo nome utente Iguazio:

   `IGZ_CONTAINER_PATH = '/users/admin'`

   Di seguito sono riportati i dettagli di connessione del sistema ONTAP. Includere il nome del volume generato al momento dell'installazione di Trident. La seguente impostazione si intende per un cluster ONTAP on-premise:

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

   La seguente impostazione è per Cloud Volumes ONTAP:

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

Creare immagini base Docker

Tutto ciò di cui hai bisogno per creare una pipeline ML è incluso nella piattaforma Iguazio. Lo sviluppatore può definire le specifiche delle immagini Docker richieste per eseguire la pipeline ed eseguire la creazione dell'immagine da Jupyter notebook. Aprire il notebook `create- images.ipynb` Ed eseguire tutte le celle.

Questo notebook crea due immagini che utilizziamo in pipeline.

- `iguazio/netapp`. Utilizzato per gestire le attività 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 utility per gestire le copie Snapshot di 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()
```

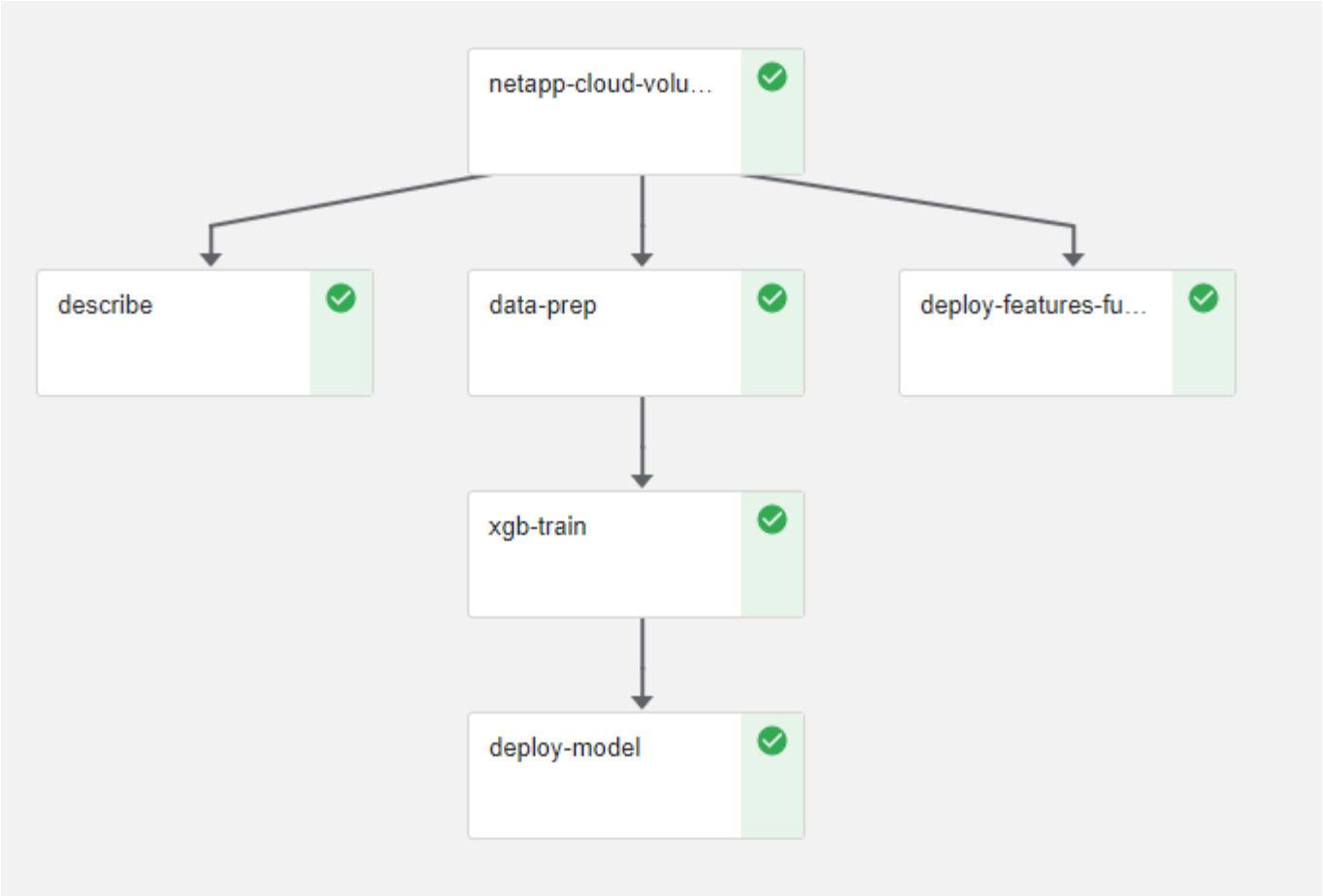
Esamina i singoli notebook Jupyter

La tabella seguente elenca le librerie e i framework utilizzati per creare questo task. Tutti questi componenti sono stati completamente integrati con i controlli di sicurezza e accesso basati sui ruoli di Iguazio.

Librerie/Framework	Descrizione
MLRun	Gestito da Iguazio per consentire l'assemblaggio, l'esecuzione e il monitoraggio di una pipeline ML/ai.
Nuclio	Un framework di funzioni senza server integrato con Iguazio. Disponibile anche come progetto open-source gestito da Iguazio.
Kubeflow	Un framework basato su Kubernetes per implementare la pipeline. Si tratta anche di un progetto open-source al quale Iguazio contribuisce. È integrato con Iguazio per una maggiore sicurezza e integrazione con il resto dell'infrastruttura.
Docker	Un registro Docker viene eseguito come servizio nella piattaforma Iguazio. È inoltre possibile modificare questa impostazione per connettersi al registro.
NetApp Cloud Volumes	I volumi cloud eseguiti su AWS ci offrono l'accesso a grandi quantità di dati e la possibilità di eseguire copie Snapshot per la versione dei set di dati utilizzati per il training.
Trident	Trident è un progetto open-source gestito da NetApp. Facilita l'integrazione con risorse di storage e calcolo in Kubernetes.

Abbiamo utilizzato diversi notebook per costruire LA pipeline ML. Ogni notebook può essere testato singolarmente prima di essere messo insieme nella pipeline. Ciascun notebook viene descritto singolarmente seguendo il flusso di implementazione di questa applicazione dimostrativa.

Il risultato desiderato è una pipeline che forma un modello basato su una copia Snapshot dei dati e implementa il modello per l'inferenza. Un diagramma a blocchi di una pipeline MLRun completa viene mostrato nell'immagine seguente.



Implementare la funzione di generazione dei dati

In questa sezione viene descritto come abbiamo utilizzato le funzioni senza server di Nuclio per generare i dati dei dispositivi di rete. Il caso di utilizzo viene adattato da un client Iguazio che ha implementato la pipeline e utilizzato i servizi Iguazio per monitorare e prevedere i guasti dei dispositivi di rete.

Abbiamo simulato i dati provenienti dai dispositivi di rete. Esecuzione del notebook Jupyter `data-generator.ipynb` Crea una funzione senza server che viene eseguita ogni 10 minuti e genera un file di parquet con nuovi dati. Per implementare la funzione, eseguire tutte le celle di questo notebook. Vedere "Sito web di Nuclio" per esaminare eventuali componenti non familiari presenti in questo notebook.

Una cella con il seguente commento viene ignorata durante la generazione della funzione. Si presume che ogni cella del notebook faccia parte della funzione. Importare il modulo Nuclio per attivarlo `%nuclio magic`.

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

Nella specifica della funzione, abbiamo definito l'ambiente in cui viene eseguita la funzione, il modo in cui viene attivata e le risorse che utilizza.

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

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

Il `init_context` La funzione viene richiamata dal framework Nuclio all'inizializzazione della funzione.

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

Quando la funzione viene inizializzata, viene richiamato qualsiasi codice non presente in una funzione. Quando lo si richiama, viene eseguita una funzione di handler. È possibile modificare il nome del gestore e specificarlo nella specifica della funzione.

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

È possibile verificare la funzione dal notebook prima dell'implementazione.

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

La funzione può essere implementata dal notebook o da una pipeline ci/CD (adattando questo codice).

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

Notebook Pipeline

Questi notebook non devono essere eseguiti singolarmente per questa configurazione. Questa è solo una recensione di ogni notebook. Li abbiamo invocati come parte della pipeline. Per eseguirli singolarmente, consultare la documentazione di MLRun per eseguirli come lavori Kubernetes.

snap_cv.ipynb

Questo notebook gestisce le copie Cloud Volume Snapshot all'inizio della pipeline. Passa il nome del volume al contesto della pipeline. Questo notebook richiama uno script shell per gestire la copia Snapshot. Durante l'esecuzione nella pipeline, il contesto di esecuzione contiene variabili che consentono di individuare tutti i file

necessari per l'esecuzione. Durante la scrittura di questo codice, lo sviluppatore non deve preoccuparsi della posizione del file nel contenitore che lo esegue. Come descritto in seguito, questa applicazione viene implementata con tutte le dipendenze, ed è la definizione dei parametri della pipeline che fornisce il contesto di esecuzione.

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

La posizione della copia Snapshot creata viene inserita nel contesto MLRun per essere utilizzata dalle fasi della pipeline.

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

I tre notebook successivi vengono eseguiti in parallelo.

data-prep.ipynb

Le metriche raw devono essere trasformate in funzionalità per consentire la formazione su modelli. Questo notebook legge le metriche raw dalla directory Snapshot e scrive le funzionalità per il training sui modelli nel volume NetApp.

Quando viene eseguito nel contesto della pipeline, l'input `DATA_DIR` Contiene la posizione della 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')))
```

descripse.ipynb

Per visualizzare le metriche in entrata, implementiamo una fase di pipeline che fornisce grafici e grafici disponibili attraverso le interfacce utente Kubeflow e MLRun. Ogni esecuzione dispone di una propria versione di questo tool di visualizzazione.

```
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

Monitoriamo continuamente le metriche alla ricerca di anomalie. Questo notebook crea una funzione senza server che genera le funzionalità necessarie per eseguire la previsione sulle metriche in entrata. Questo notebook richiama la creazione della funzione. Il codice funzione si trova nel notebook `data- prep.ipynb`. A questo scopo, utilizziamo lo stesso notebook come passaggio della pipeline.

training.ipynb

Dopo aver creato le funzionalità, avviamo il training sul modello. L'output di questa fase è il modello da utilizzare per l'deduzione. Raccogliamo inoltre statistiche per tenere traccia di ogni esecuzione (esperimento).

Ad esempio, il comando seguente inserisce il punteggio di precisione nel contesto dell'esperimento. Questo valore è visibile in Kubeflow e MLRun.

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

deploy-inference-function.ipynb

L'ultima fase della pipeline consiste nell'implementare il modello come funzione senza server per deduzione continua. Questo notebook richiama la creazione della funzione senza server definita in `nuclio-inference- function.ipynb`.

Esaminare e costruire la pipeline

La combinazione di eseguire tutti i notebook in una pipeline consente l'esecuzione continua di esperimenti per rivalutare l'accuratezza del modello rispetto alle nuove metriche. Aprire innanzitutto `pipeline.ipynb` notebook. Ti illustreremo i dettagli che mostrano come NetApp e Iguazio semplificano l'implementazione di questa pipeline ML.

Utilizziamo MLRun per fornire contesto e gestire l'allocazione delle risorse in ogni fase della pipeline. Il servizio API MLRun viene eseguito nella piattaforma Iguazio ed è il punto di interazione con le risorse Kubernetes. Ogni sviluppatore non può richiedere direttamente le risorse; l'API gestisce le richieste e abilita i controlli di accesso.

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

La pipeline può funzionare con volumi cloud NetApp e volumi on-premise. Questa dimostrazione è stata realizzata per utilizzare i volumi cloud, ma è possibile vedere nel codice l'opzione di esecuzione on-premise.

```
# 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 prima azione necessaria per trasformare un notebook Jupyter in un passo Kubeflow è trasformare il codice in una funzione. Una funzione ha tutte le specifiche richieste per eseguire il notebook. Quando scorri il notebook, puoi vedere che definiamo una funzione per ogni fase della pipeline.

Parte del notebook	Descrizione
<code_to_function> (parte del modulo MLRun)	Nome della funzione: Nome del progetto. utilizzato per organizzare tutti gli artefatti del progetto. Questo è visibile nell'interfaccia utente di MLRun. Gentile. In questo caso, un lavoro Kubernetes. Questo potrebbe essere Dask, mpi, sparkk8s e molto altro ancora. Per ulteriori informazioni, consulta la documentazione di MLRun. File. Il nome del notebook. Questa può anche essere una posizione in Git (HTTP).
immagine	Il nome dell'immagine Docker che stiamo utilizzando per questo passaggio. Abbiamo creato questo documento in precedenza con il notebook create-image.ipynb.
montaggi_volumi e volumi	Dettagli per montare il NetApp Cloud Volume in fase di esecuzione.

Definiamo anche i parametri per le fasi.

```
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
            }
```

Una volta definita la funzione per tutti i passaggi, è possibile costruire la pipeline. Utilizziamo il `kfp` per definire questa definizione. La differenza tra l'utilizzo di MLRun e la creazione di codice da soli è la semplificazione e la riduzione del codice.

Le funzioni che abbiamo definito vengono trasformate in componenti passo-passo utilizzando `as_step` Funzione di MLRun.

Definizione della fase Snapshot

Avviare una funzione Snapshot, eseguire l'output e montare v3io come origine:

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

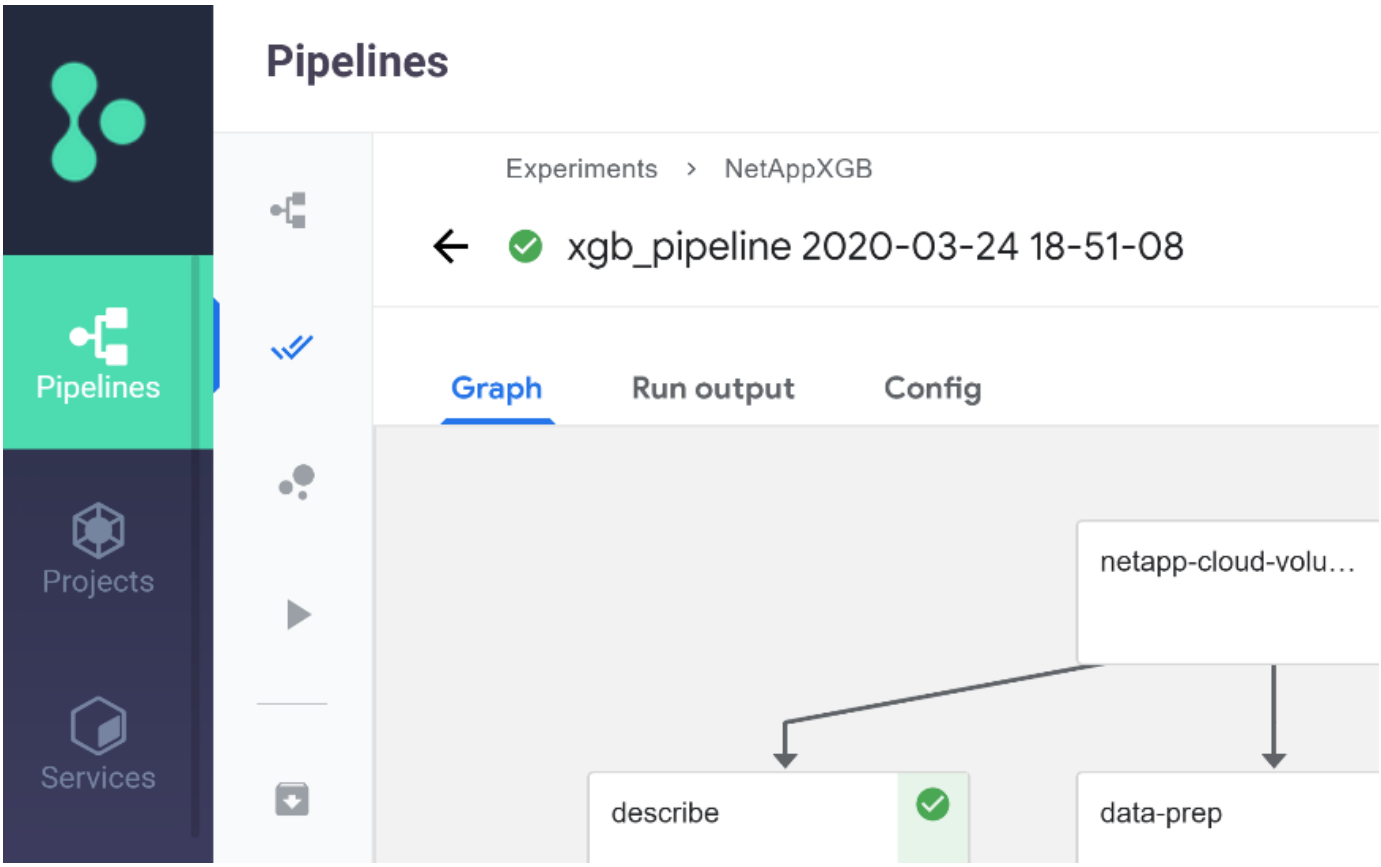
Parametri	Dettagli
NewTask	NewTask è la definizione dell'esecuzione della funzione.
(Modulo MLRun)	Gestore. Nome della funzione Python da richiamare. Abbiamo utilizzato il gestore dei nomi nel notebook, ma non è necessario. parametri. I parametri passati all'esecuzione. All'interno del codice, utilizziamo Context.get_param ('PARAMETRO') per ottenere i valori.

Parametri	Dettagli
as_step	Nome. Nome della fase della pipeline Kubeflow. output. Questi sono i valori che la procedura aggiunge al dizionario al completamento. Dai un'occhiata al notebook SNAP_cv.ipynb. mount_v3io(). In questo modo viene configurato il passo per montare /User per l'utente che esegue la pipeline.

```
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)
```

Parametri	Dettagli
input	È possibile passare a un passo gli output di un passo precedente. In questo caso, snap.outputs['snapVolumeDetails'] è il nome della copia Snapshot creata nel passo SNAP.
out_path	Una posizione in cui posizionare gli artefatti che generano utilizzando il modulo MLRun log_Artifacts.

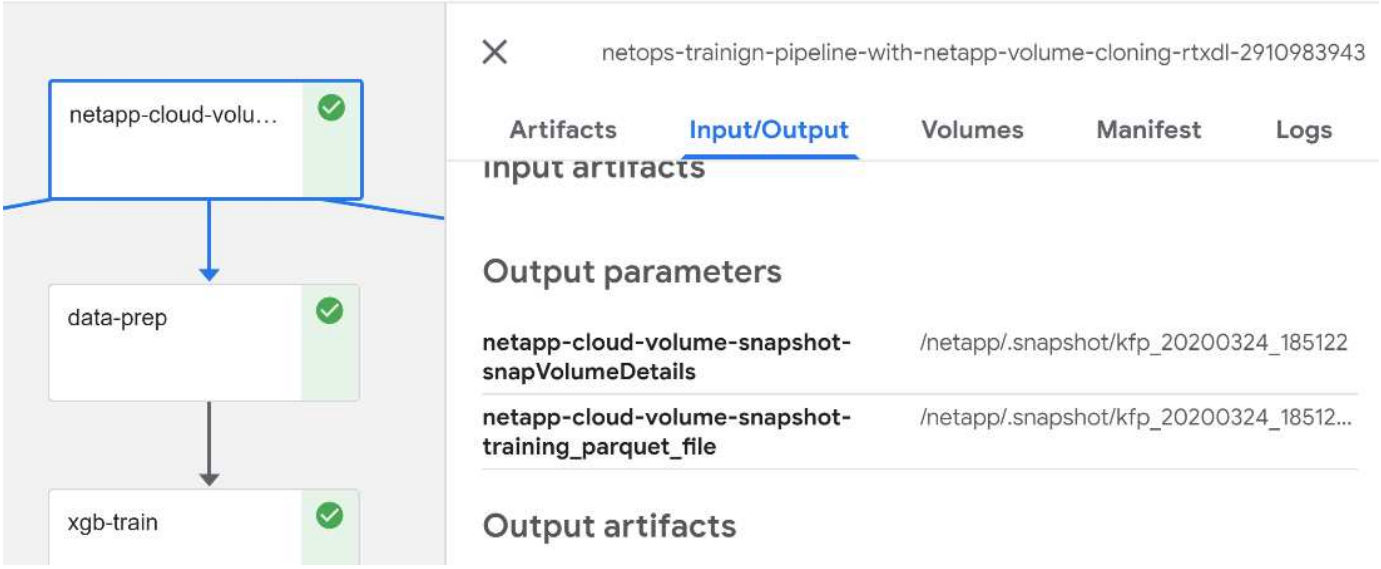
Puoi correre `pipeline.ipynb` dall'alto verso il basso. È quindi possibile accedere alla scheda Pipeline dalla dashboard di Iguazio per monitorare l'avanzamento, come mostrato nella scheda Pipeline della dashboard di Iguazio.



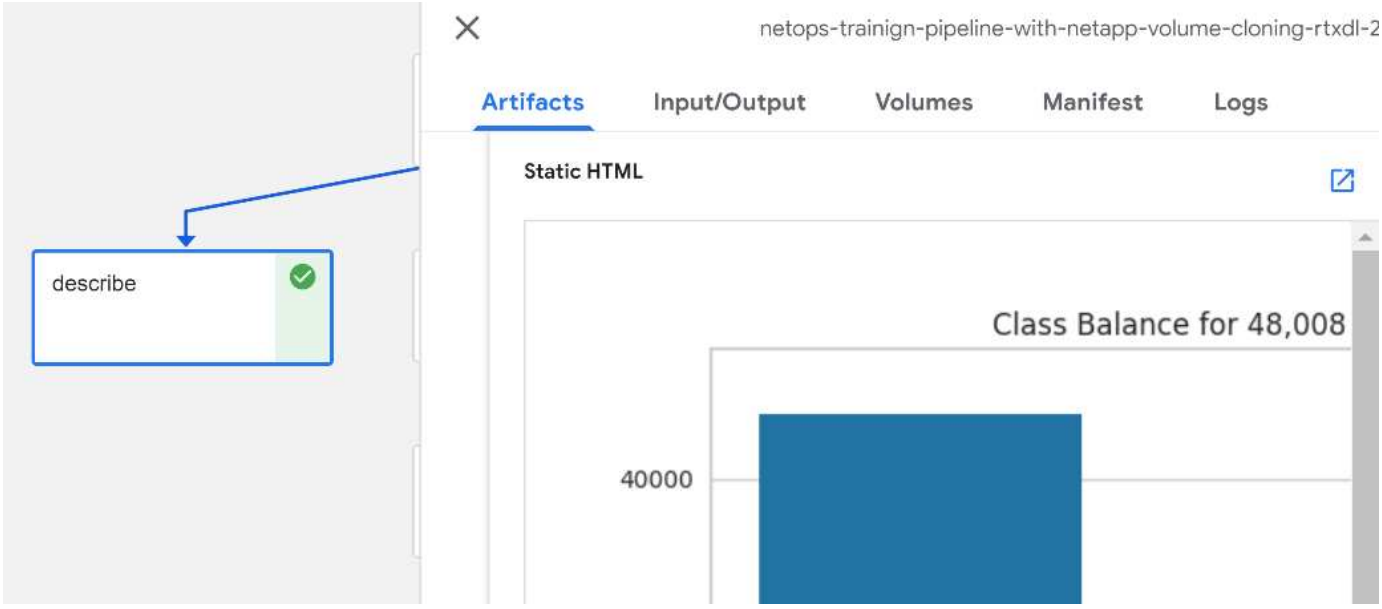
Poiché abbiamo registrato la precisione delle fasi di training in ogni sessione, abbiamo una registrazione di accuratezza per ogni esperimento, come mostrato nella documentazione relativa alla precisione del training.

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

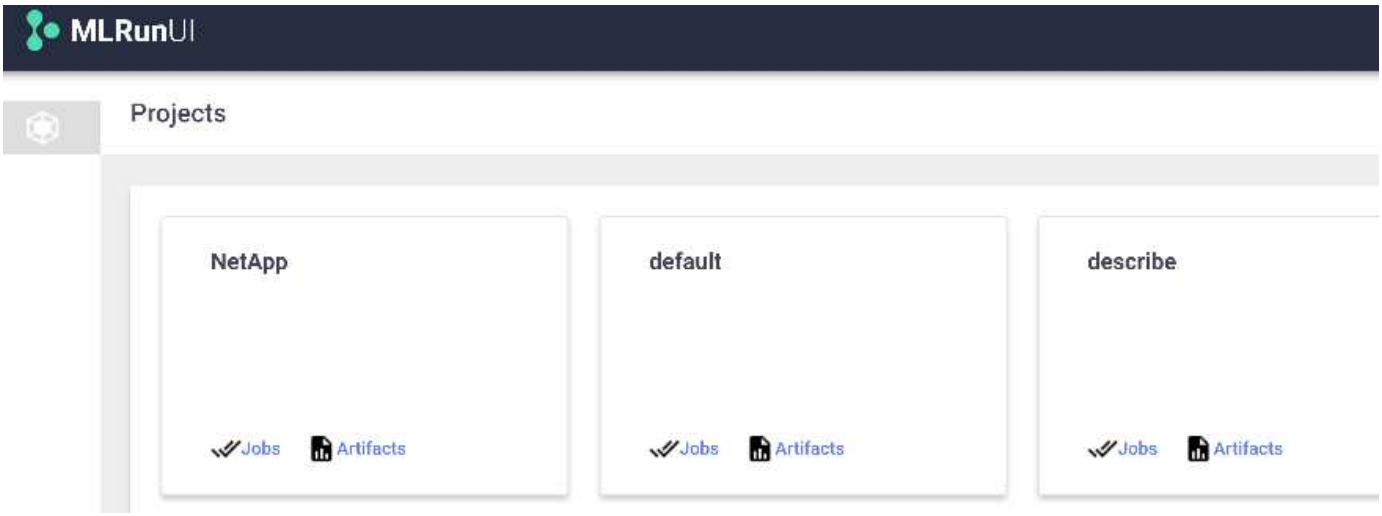
Se si seleziona la fase Snapshot, è possibile visualizzare il nome della copia Snapshot utilizzata per eseguire questo esperimento.



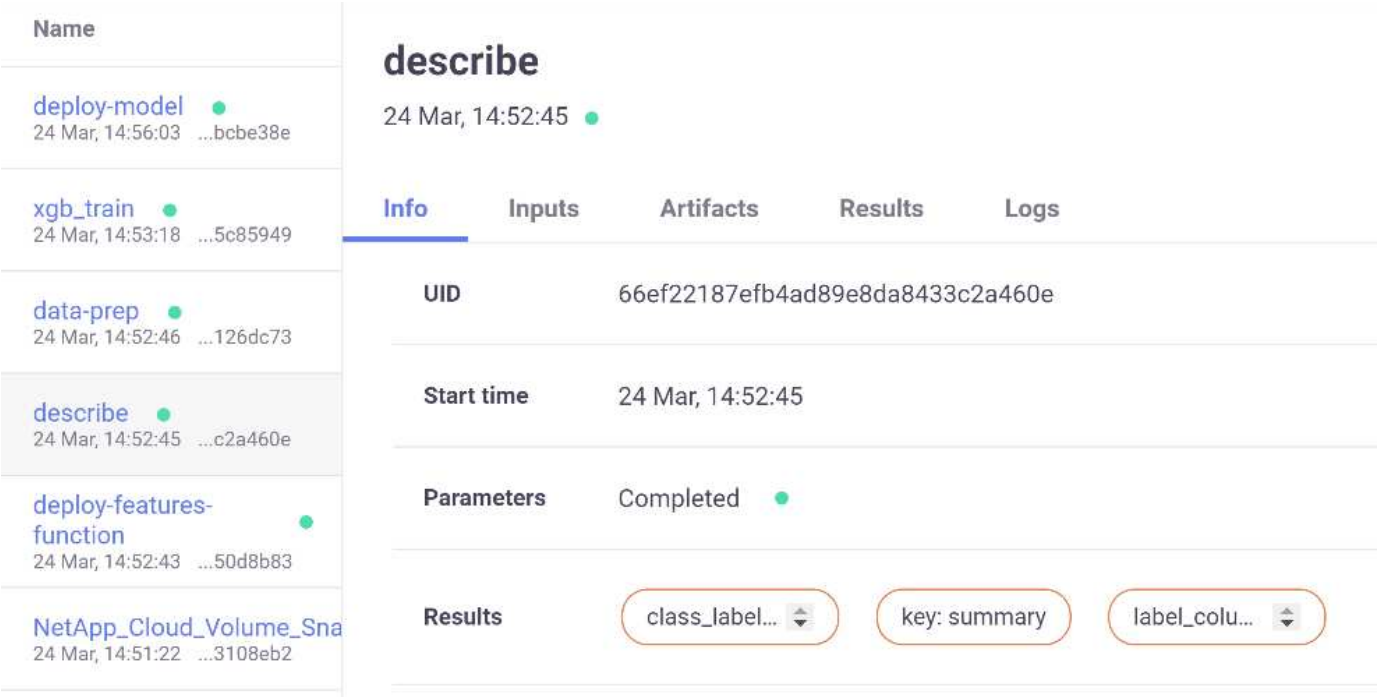
La fase descritta presenta artefatti visivi per esplorare le metriche utilizzate. È possibile espandere per visualizzare il grafico completo come mostrato nell'immagine seguente.



Il database API di MLRun tiene traccia anche di input, output e artefatti per ogni esecuzione organizzata per progetto. Un esempio di input, output e artefatti per ciascuna seriografia può essere visualizzato nell'immagine seguente.



Per ogni lavoro, memorizziamo ulteriori dettagli.



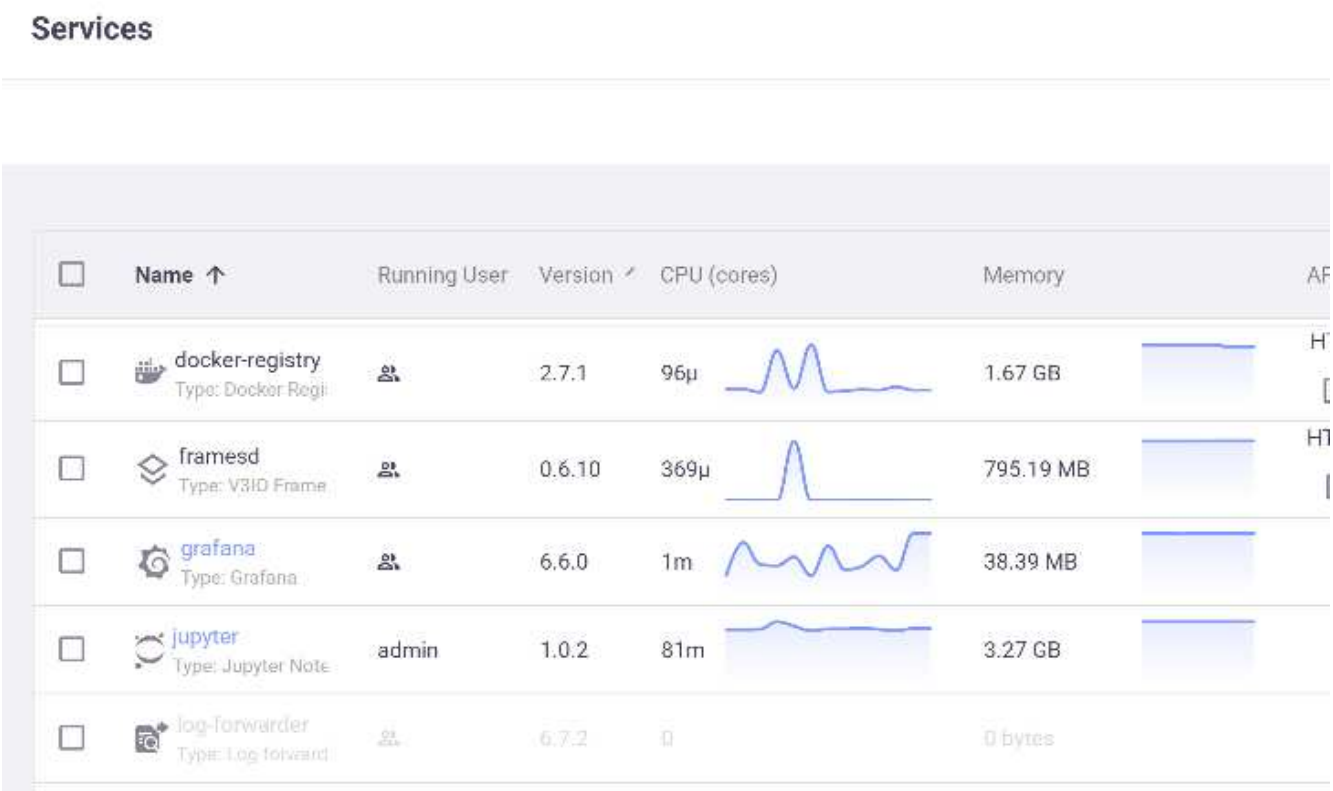
In questo documento sono disponibili ulteriori informazioni su MLRun. Gli artefatti di al, inclusa la definizione delle fasi e delle funzioni, possono essere salvati nel database API, con versione e richiamati singolarmente o come progetto completo. I progetti possono anche essere salvati e inviati a Git per un utilizzo successivo. Ti invitiamo a scoprire di più su "Sito MLRun GitHub".

Implementare Grafana Dashboard

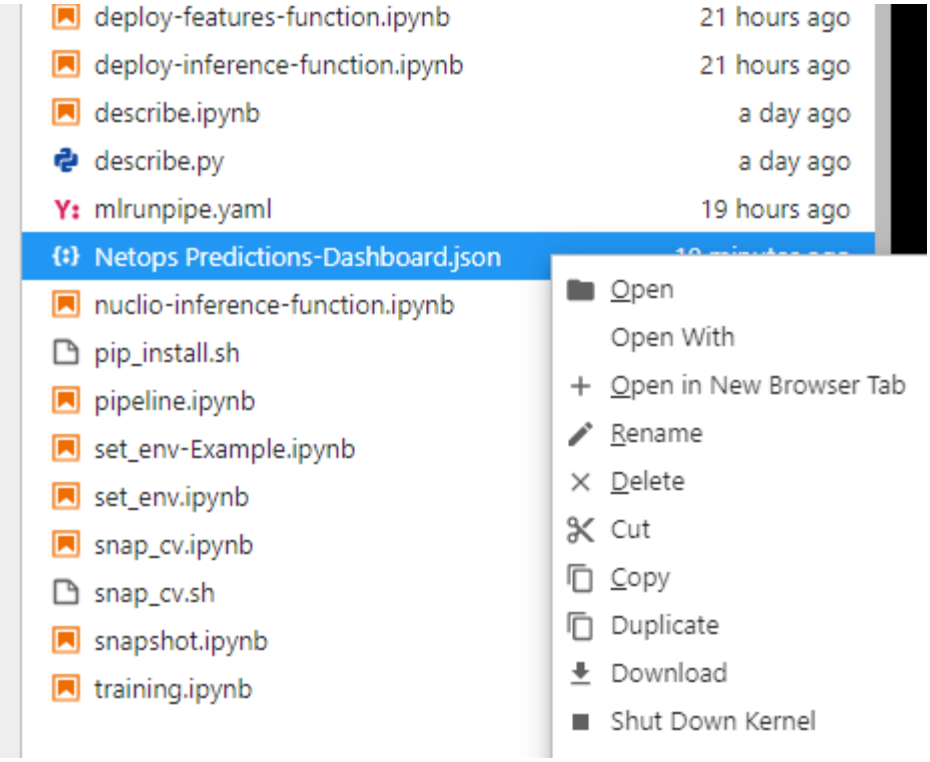
Una volta implementato tutto, eseguiamo le inferenze sui nuovi dati. I modelli prevedono guasti sulle apparecchiature di rete. I risultati della previsione vengono memorizzati in una tabella Iguazio TimeSeries. È possibile visualizzare i risultati con Grafana nella piattaforma integrata con la policy di sicurezza e accesso ai dati di Iguazio.

È possibile implementare la dashboard importando il file JSON fornito nelle interfacce Grafana del cluster.

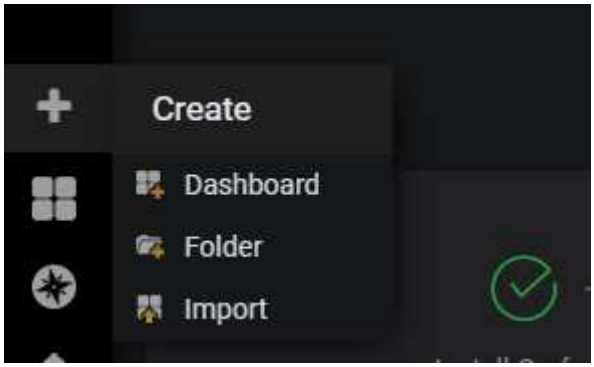
1. Per verificare che il servizio Grafana sia in esecuzione, consultare la sezione servizi.



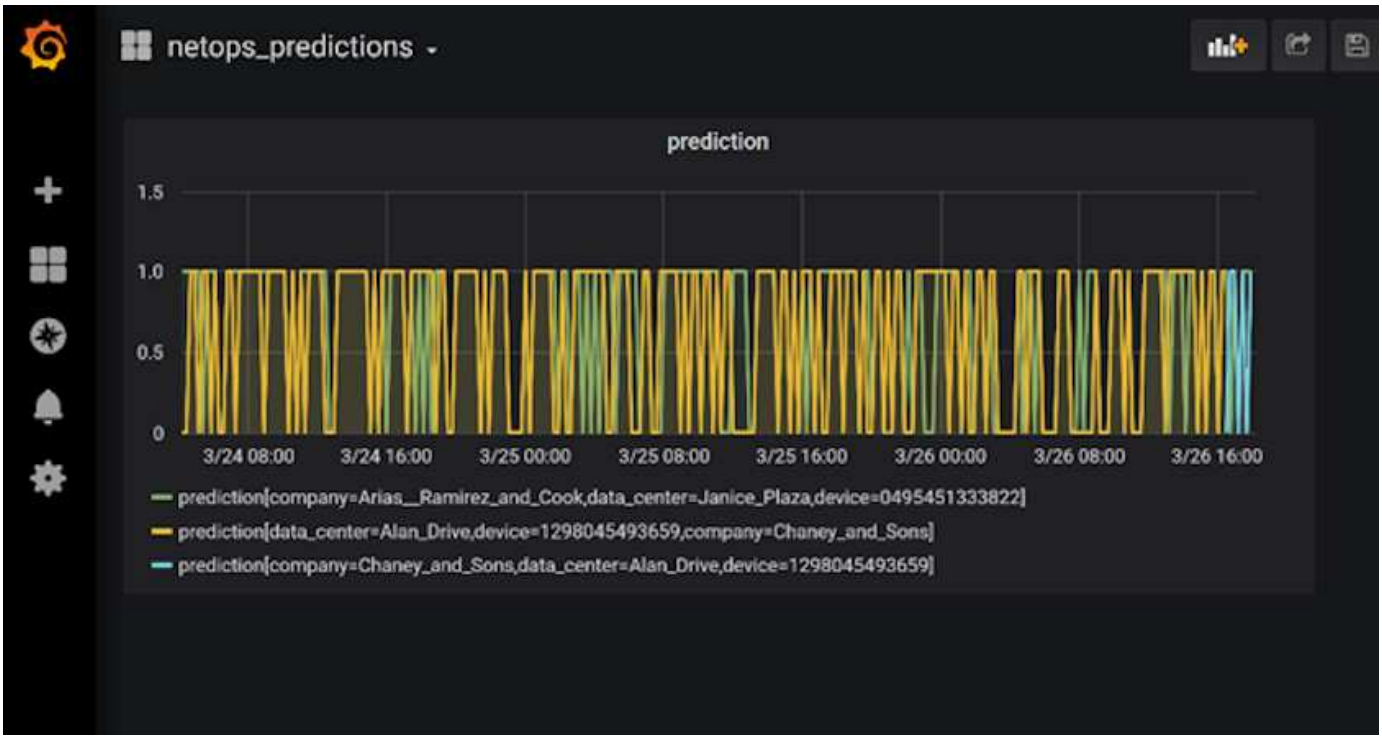
2. Se non è presente, distribuire un'istanza dalla sezione servizi:
   a. Fare clic su nuovo servizio.
   b. Selezionare Grafana dall'elenco.
   c. Accettare le impostazioni predefinite.
   d. Fare clic su Avanti.
   e. Inserire l'ID utente.
   f. Fare clic su Salva servizio.
   g. Fare clic su Apply Changes (Applica modifiche) nella parte superiore.
3. Per implementare la dashboard, scaricare il file `NetopsPredictions-Dashboard.json` Tramite l'interfaccia Jupyter.



4. Aprire Grafana dalla sezione servizi e importare la dashboard.



5. Fare clic su carica `*.json` E selezionare il file scaricato in precedenza (`NetopsPredictions-Dashboard.json`). La dashboard viene visualizzata al termine del caricamento.



Implementare la funzione di pulizia

Quando si generano molti dati, è importante mantenere le cose pulite e organizzate. A tale scopo, implementare la funzione di pulizia con `cleanup.ipynb` notebook.

Benefici

NetApp e Iguazio accelerano e semplificano l'implementazione delle applicazioni ai e ML creando framework essenziali come Kubeflow, Apache Spark e TensorFlow, oltre a tool di orchestrazione come Docker e Kubernetes. Unificando la pipeline di dati end-to-end, NetApp e Iguazio riducono la latenza e la complessità inerenti a molti carichi di lavoro di calcolo avanzati, colmando efficacemente il divario tra sviluppo e operazioni. I data scientist possono eseguire query su set di dati di grandi dimensioni e condividere in modo sicuro dati e modelli algoritmici con utenti autorizzati durante la fase di training. Dopo che i modelli containerizzati sono pronti per la produzione, è possibile spostarli facilmente dagli ambienti di sviluppo agli ambienti operativi.

Conclusione

Quando si creano pipeline ai/ML personalizzate, configurare l'integrazione, la gestione, la sicurezza e l'accessibilità dei componenti in un'architettura è un'attività complessa. Fornire agli sviluppatori l'accesso e il controllo del proprio ambiente presenta un'altra serie di sfide.

La combinazione di NetApp e Iguazio riunisce queste tecnologie come servizi gestiti per accelerare l'adozione della tecnologia e migliorare il time-to-market per le nuove applicazioni ai/ML.

TR-4915: Spostamento dei dati con e-Series e BeeGFS per i flussi di lavoro di ai e analytics

Cody Harryman e Ryan Rodine, NetApp

TR-4915 descrive come spostare i dati da qualsiasi repository di dati in un file system BeeGFS supportato dallo storage SAN NetApp e-Series. Per le applicazioni di intelligenza artificiale (ai) e machine learning (ML), i clienti potrebbero dover spostare regolarmente grandi set di dati che superano molti petabyte di dati nei cluster BeeGFS per lo sviluppo dei modelli. In questo documento viene spiegato come ottenere tutto questo utilizzando gli strumenti di copia e sincronizzazione BlueXP di NetApp XCP e NetApp.

"TR-4915: Spostamento dei dati con e-Series e BeeGFS per i flussi di lavoro di ai e analytics"