



資料管道、資料湖區與管理 NetApp Solutions

NetApp
April 12, 2024

This PDF was generated from https://docs.netapp.com/zh-tw/netapp-solutions/ai/mlops_fsxn_s3_integration.html on April 12, 2024. Always check docs.netapp.com for the latest.

目錄

資料管道、資料湖區與管理	1
適用於 MLOps 的 AWS FSx for NetApp ONTAP （ FSxN ）	1
採用 Domino Data Lab 和 NetApp 的混合式多雲端 MLOps	35
NVIDIA AI Enterprise搭配NetApp和VMware	49
TR-4851：NetApp StorageGRID 的資料湖適用於自動驅動工作負載-解決方案設計	59
NetApp AI Control Plane	59
採用Iguazio的MLRun Pipeline	110
TR-4915：E系列和BeeGFS的資料搬移功能、適用於AI和分析工作流程	136

資料管道、資料湖區與管理

適用於 MLOps 的 AWS FSx for NetApp ONTAP (FSxN)

- 作者：*
NetApp 資深資料與應用科學家 Jian Jian (Ken)

本節將深入探討 AI 基礎架構開發的實際應用、提供使用 FSxN 建構 MLOps 管線的端點對端逐步解說。它包含三個完整的範例、可引導您透過這個強大的資料管理平台來滿足 MLOps 需求。

這些文章著重於：

1. "第 1 部分：將 AWS FSx for NetApp ONTAP (FSxN) 整合為私有 S3 儲存區、並整合至 AWS SageMaker"
2. "第 2 部分：運用 AWS FSx for NetApp ONTAP (FSxN) 做為 SageMaker 模型訓練的資料來源"
3. "第 3 部分：建立簡化的 MLOps 管道 (CI/CT/CD) "

在本節結束時、您將深入瞭解如何使用 FSxN 來簡化 MLOps 程序。

第 1 部分：將 AWS FSx for NetApp ONTAP (FSxN) 整合為私有 S3 儲存區、並整合至 AWS SageMaker

- 作者：*
NetApp 資深資料與應用科學家 Jian Jian (Ken)

簡介

以 SageMaker 為例、本頁提供將 FSxN 設定為私有 S3 儲存區的指引。

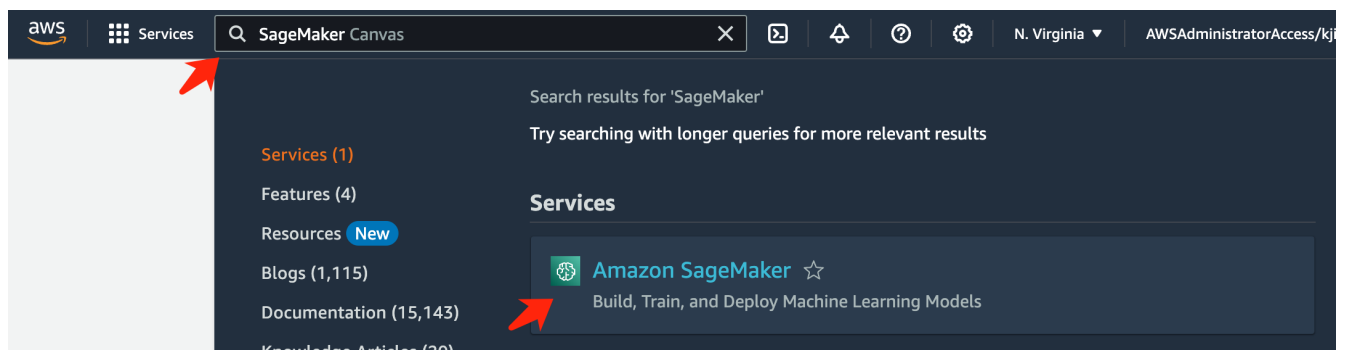
如需 FSxN 的詳細資訊、請參閱本簡報 ("[視訊連結](#)")

使用者指南

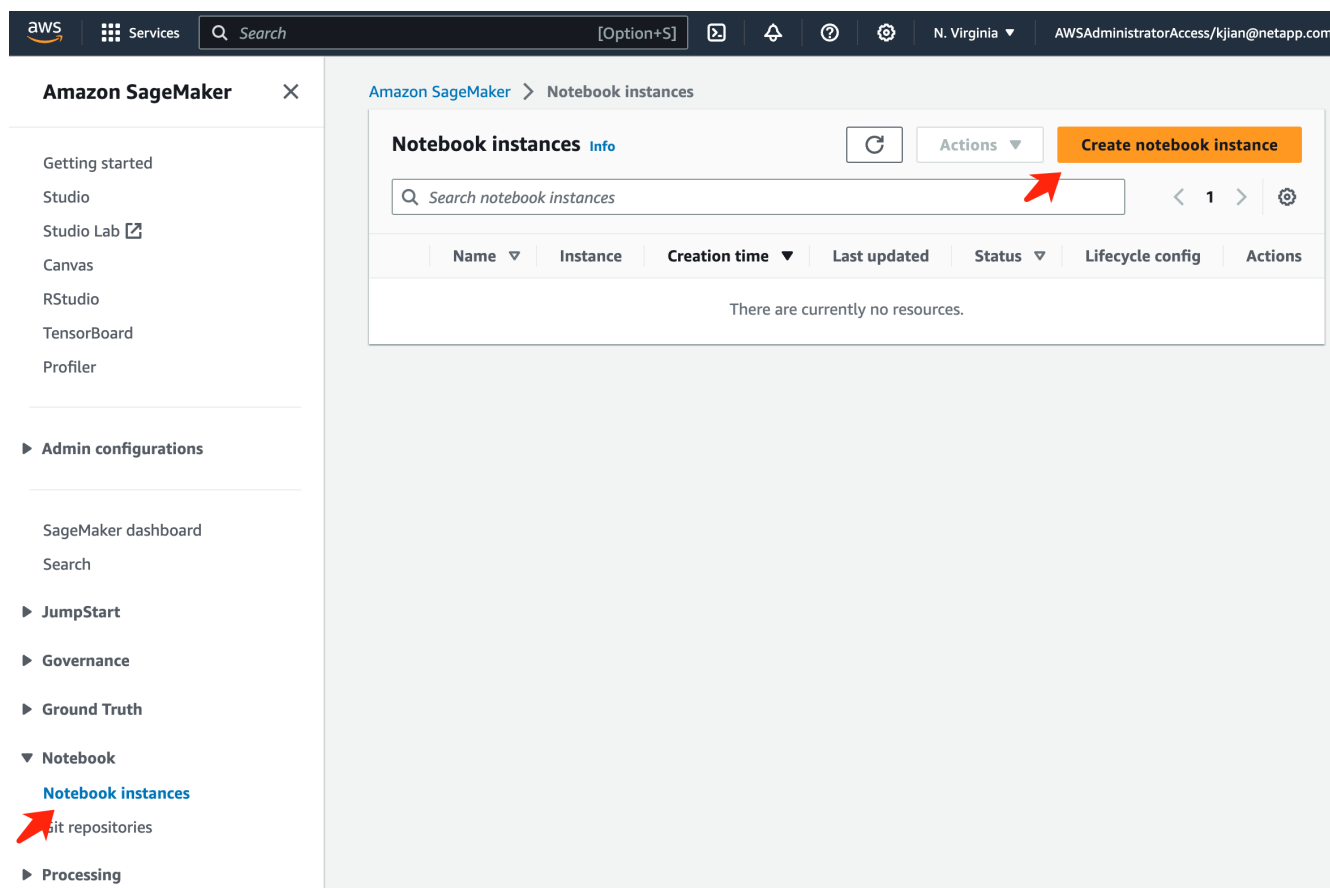
伺服器建立

建立 SageMaker 筆記型電腦執行個體

1. 開啟 AWS 主控台。在搜尋面板中、搜尋 SageMaker、然後按一下服務 * Amazon SageMaker*。



2. 開啟 [筆記本] 索引標籤下的 * 筆記本執行個體 * ，按一下橘色按鈕 * 建立筆記本執行個體 * 。



3. 在建立頁面中、
輸入 * 筆記本執行個體名稱 *
展開 **Network** 面板
保留其他項目的預設值、然後選取 * VPC* 、 * 子網路 * 和 * 安全性群組 * 。（此 **VPC** 和 * 子網路 * 將於稍後用於建立 FSxN 檔案系統）
按一下右下方的橘色按鈕 * 建立筆記本執行個體 * 。

Create notebook instance

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

Notebook instance settings

Notebook instance name

fsxn-demo

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

Notebook instance type

ml.t3.medium

Elastic Inference [Learn more](#)

none

Platform identifier [Learn more](#)

Amazon Linux 2, Jupyter Lab 3

► Additional configuration

Permissions and encryption

IAM role

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

AmazonSageMakerServiceCatalogProductsUseRole

Create role using the role creation wizard

Root access - optional

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

Encryption key - optional

Encrypt your notebook data. Choose an existing KMS key or enter a key's ARN.

No Custom Encryption

▼ Network - optional

VPC - optional

Default vpc-0df3956ab1fca2ec9 (172.31.0.0/16)

Subnet

Choose a subnet in an availability zone supported by Amazon SageMaker.

subnet-00060df0d0f562672 (172.31.16.0/20) | us-east-1a

Security group(s)

sg-0a39b3985770e9256 (default) X

Direct internet access

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

► Git repositories- optional

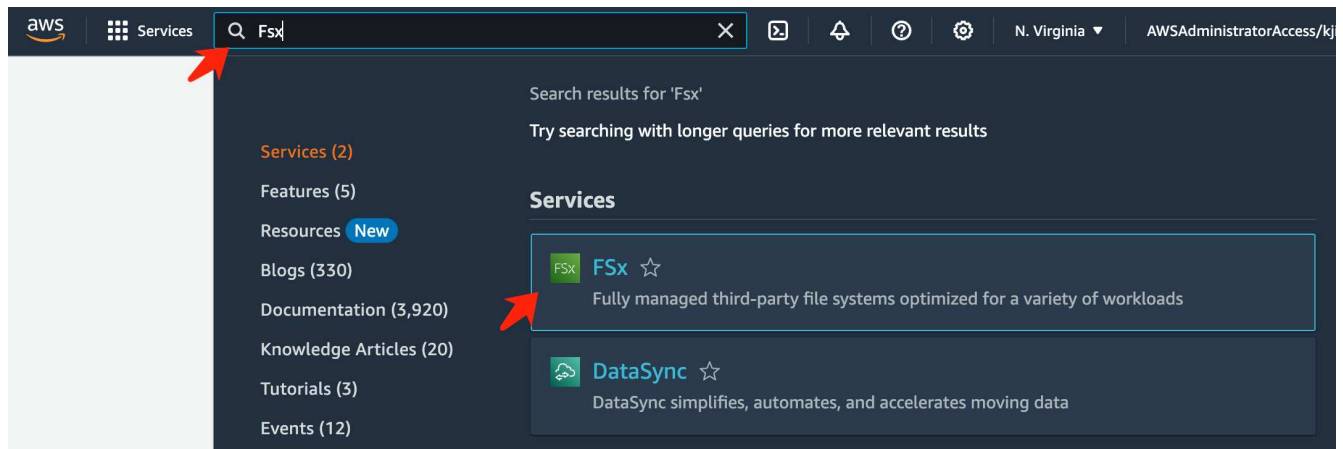
► Tags - optional

Cancel

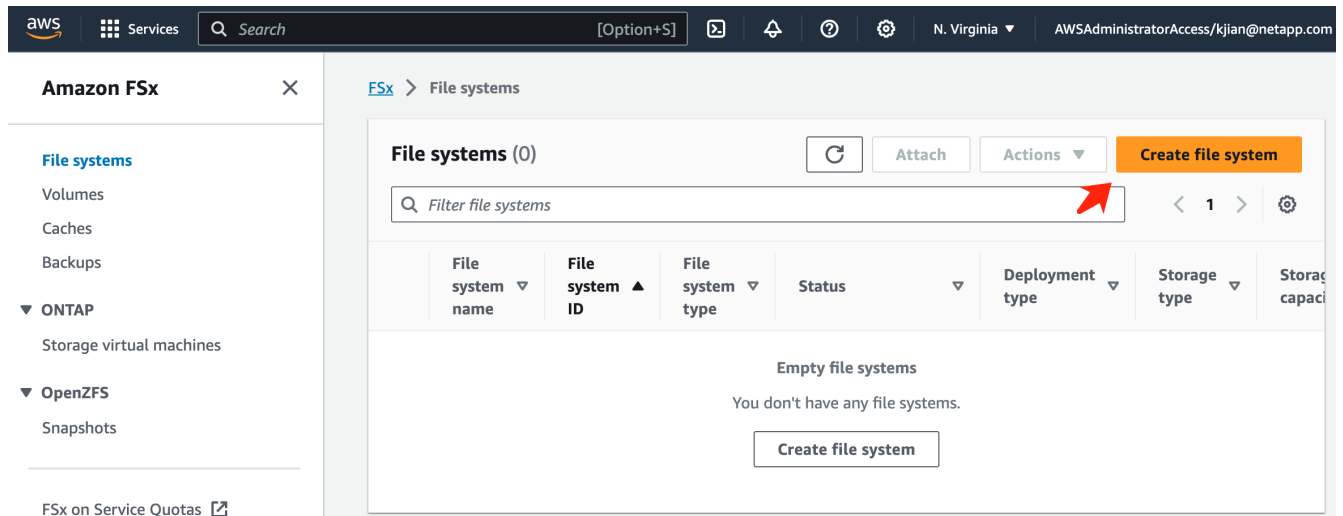
Create notebook instance

建立 FSxN 檔案系統

1. 開啟 AWS 主控台。在搜尋面板中、搜尋 FSX 並按一下服務 * fsx* 。



2. 按一下 * 建立檔案系統 * 。



3. 選擇第一張卡 *FSX for NetApp ONTAP* 、然後按一下 * 下一步 * 。

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

FSx > File systems > Create file system

Step 1
Select file system type

Step 2
Specify file system details

Step 3
Review and create

Select file system type

File system options

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

Amazon FSx for NetApp ONTAP

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

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

Cancel Next

4. 在詳細資料組態頁面中。
- a. 選擇 * 標準 cre* 選項。

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

FSx > File systems > Create file system

Step 1
Select file system type

Step 2
Specify file system details

Step 3
Review and create

Specify file system details

Creation method

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

- b. 輸入 * 檔案系統名稱 * 和 * SSD 儲存容量 * 。

File system details

File system name - optional [Info](#)

fsxn-demo

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

Deployment type [Info](#)

- ☒ Multi-AZ
☐ Single-AZ

SSD storage capacity [Info](#)

1024

GiB

Minimum 1024 GiB; Maximum 192 TiB.

Provisioned SSD IOPS

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

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

Throughput capacity [Info](#)

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

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

c. 確保使用 **VPC** 和 * 子網 * 與 **SageMaker Notebook** 實例相同。

Network & security

Virtual Private Cloud (VPC) [Info](#)

Specify the VPC from which your file system is accessible.

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

VPC Security Groups [Info](#)

Specify VPC Security Groups to associate with your file system's network interfaces.

Choose VPC security group(s) ▼

sg-0a39b3985770e9256 (default) ✕

Preferred subnet [Info](#)

Specify the preferred subnet for your file system.

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

Standby subnet

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

VPC route tables [Info](#)

Specify the VPC route tables to associate with your file system.

- ☒ VPC's main route table
- ☐ Select one or more VPC route tables

Endpoint IP address range [Info](#)

Specify the IP address range in which the endpoints to access your file system will be created

- ☒ Unallocated IP address range from your VPC
Simplest option for access from other AWS services or peered / on-premises networks
- ☐ Floating IP address range outside your VPC
- ☐ Enter an IP address range

d. 輸入 * 儲存虛擬機器 * 名稱、並 * 指定 SVM （儲存虛擬機器）的密碼 * 。

Default storage virtual machine configuration

Storage virtual machine name

Info

fsxn-svm-demo

SVM administrative password

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

☐ Don't specify a password

☒ Specify a password

Password

.....

Confirm password

.....

Volume security style

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

Unix (Linux)

Active Directory

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

☒ Do not join an Active Directory

☐ Join an Active Directory

e. 保留其他項目的預設值、然後按一下右下方的橘色按鈕 * 下一步 * 。

► Backup and maintenance - optional

► Tags - optional

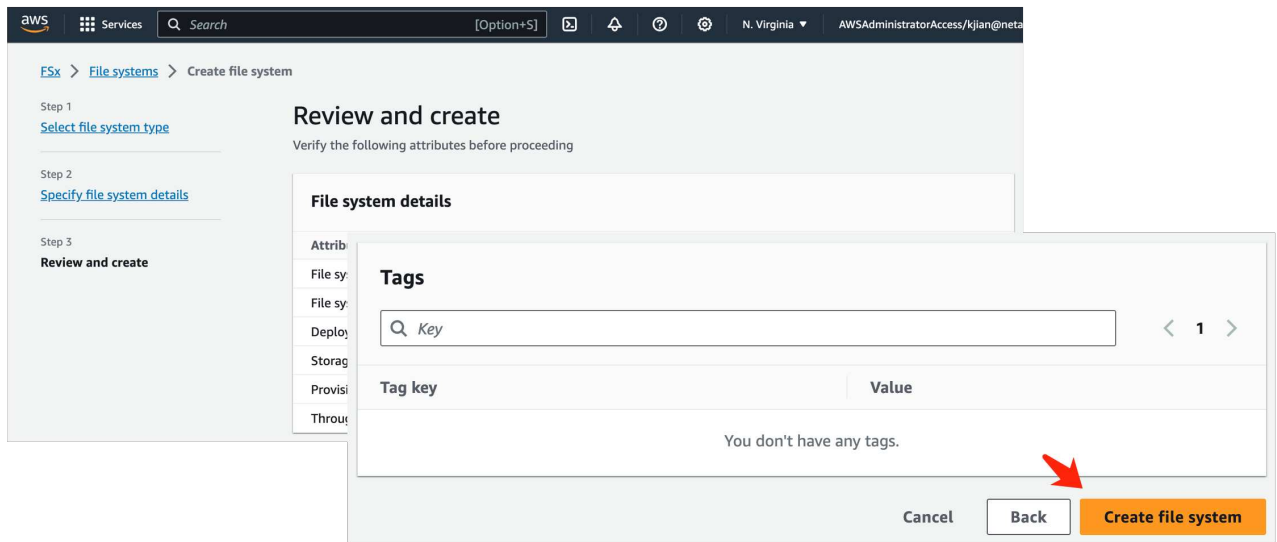
Cancel

Back

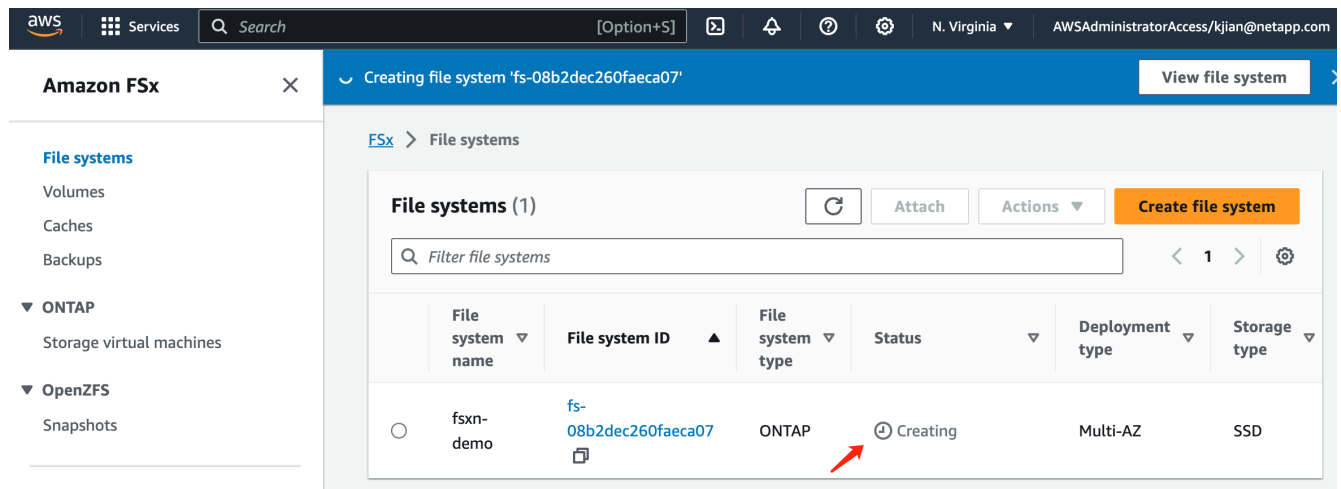
Next

f. 按一下檢閱頁面右下角的橘色按鈕 * 建立檔案系統 * 。

8



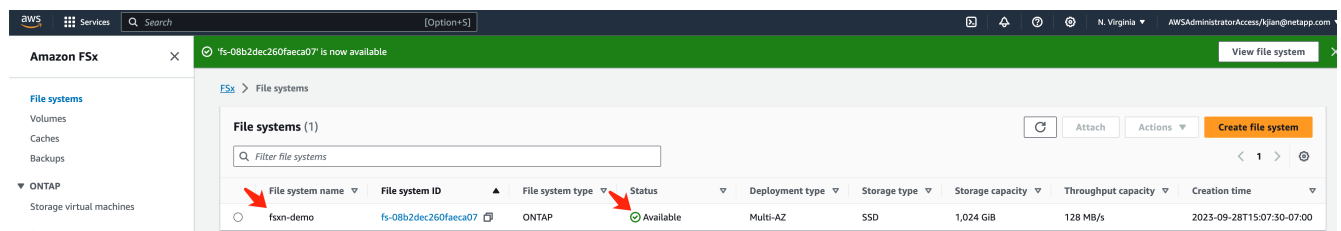
5. 增加 FSX 檔案系統可能需要大約 *20-40 分鐘*。



伺服器組態

ONTAP 組態

1. 開啟建立的 FSX 檔案系統。請確定狀態為 *可用*。



2. 選擇 **Administration (管理)** * 選項卡並保留 *Management Endpoint - IP address* 和 ONTAP 管理員用戶名*。

Amazon FSx fsxn-demo (fs-08b2dec260faeca07)

Summary

File system ID fs-08b2dec260faeca07	SSD storage capacity 1024 GiB	Availability Zones us-east-1a (Preferred) us-east-1b (Standby)
Lifecycle state Creating	Throughput capacity 128 MB/s	Creation time 2023-09-28T14:41:50-07:00
File system type ONTAP	Provisioned IOPS 3072	
Deployment type Multi-AZ		

ONTAP administration

Management endpoint - DNS name management.fs-08b2dec260faeca07.fsx.us-east-1.amazonaws.com	Management endpoint - IP address 172.31.255.250	ONTAP administrator username fsxadmin
Inter-cluster endpoint - DNS name intercluster.fs-08b2dec260faeca07.fsx.us-east-1.amazonaws.com	Inter-cluster endpoint - IP address 172.31.31.157	ONTAP administrator password <input type="button" value="Update"/>

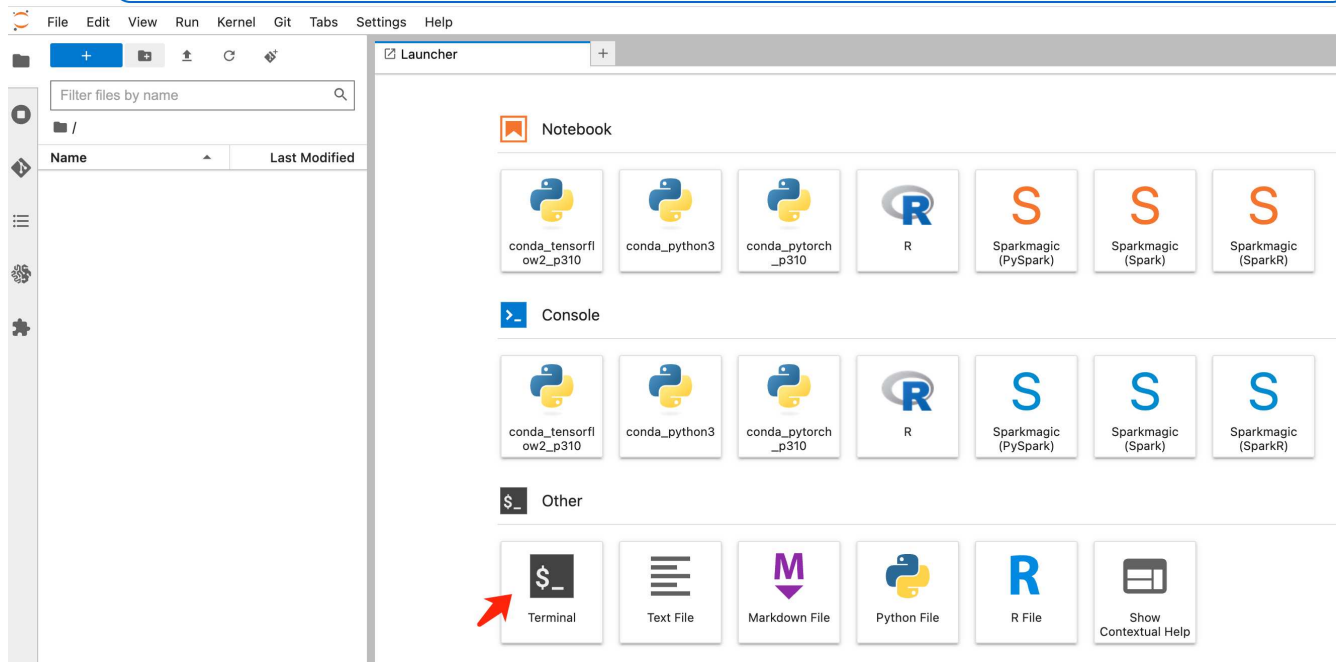
3. 開啟建立的 * SageMaker Notebook instance* 、然後按一下 * 開啟 JupyterLab* 。

Amazon SageMaker Notebook instances

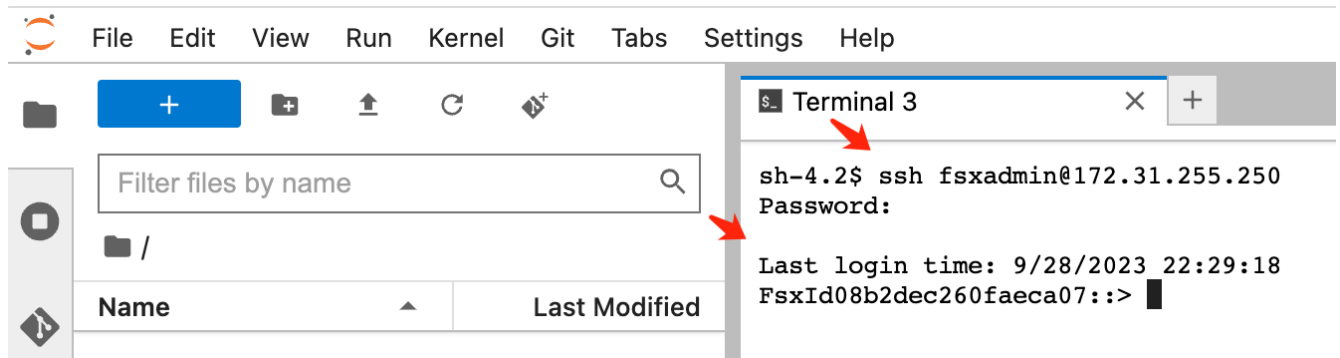
Notebook instances

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

4. 在 Jupyter Lab 頁面中，打開一個新的 * 終端 * 。



5. 輸入 ssh 命令 `ssh <admin user name>@<ONTAP server IP>` 以登入 FSxN ONTAP 檔案系統。（從步驟 2 擷取使用者名稱和 IP 位址）
請使用建立 * 儲存虛擬機器 * 時所使用的密碼。



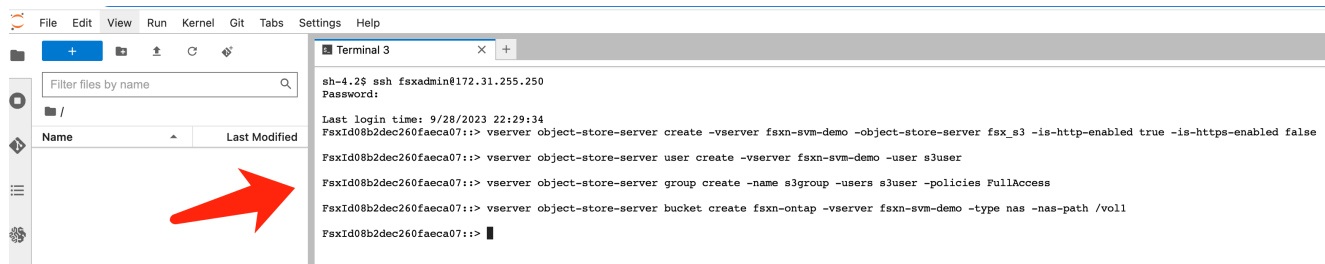
6. 依照下列順序執行命令。
我們使用 **fsxn-ONTAP** 作為 *FSxN Private S3 儲存區名稱* 的名稱。
請將 * 儲存虛擬機器名稱 * 用於 * 虛擬伺服器 * 引數。

```
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. 執行下列命令以擷取 FSxN Private S3 的端點 IP 和認證。

```

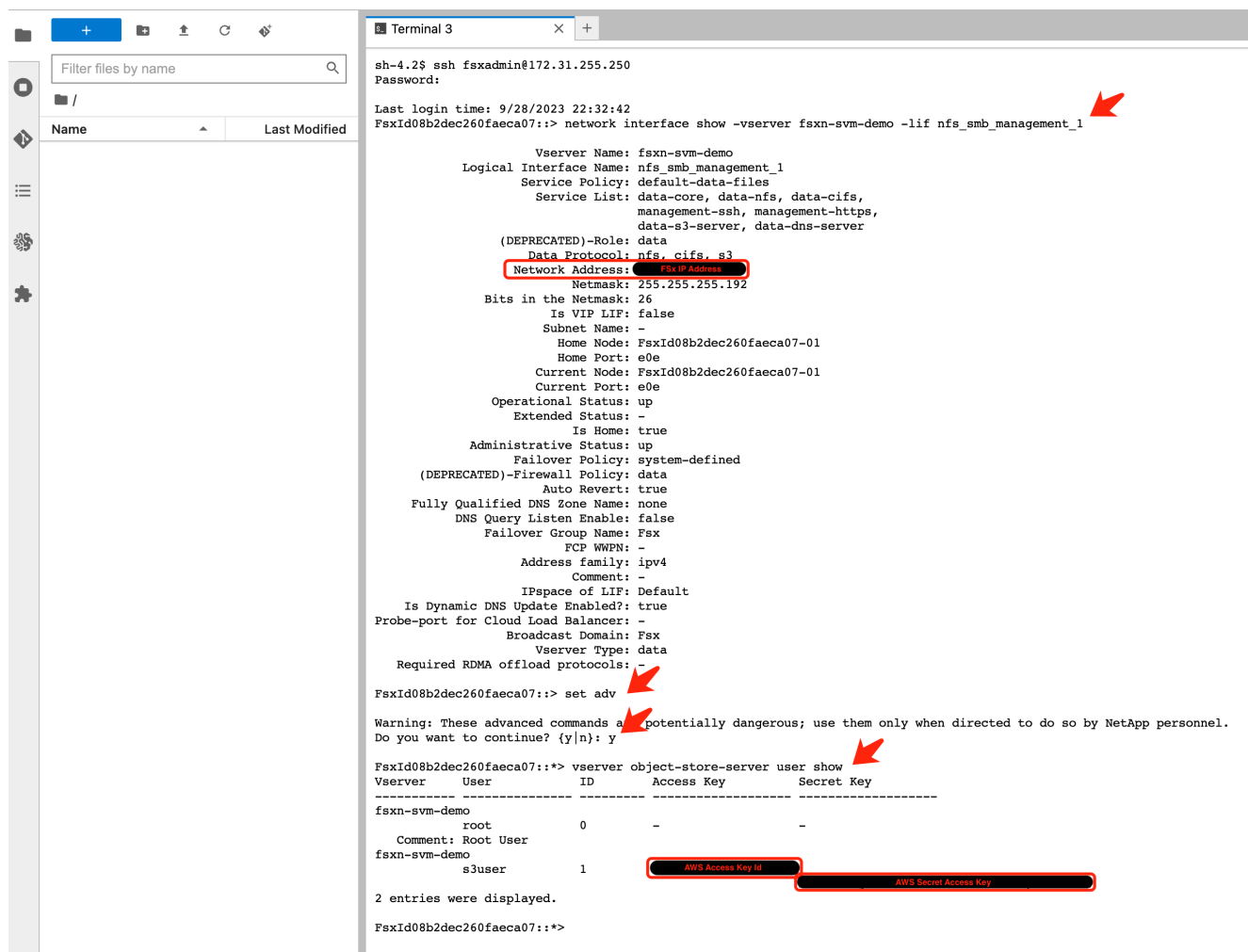
network interface show -vserver fsxn-svm-demo -lif nfs_smb_management_1

set adv

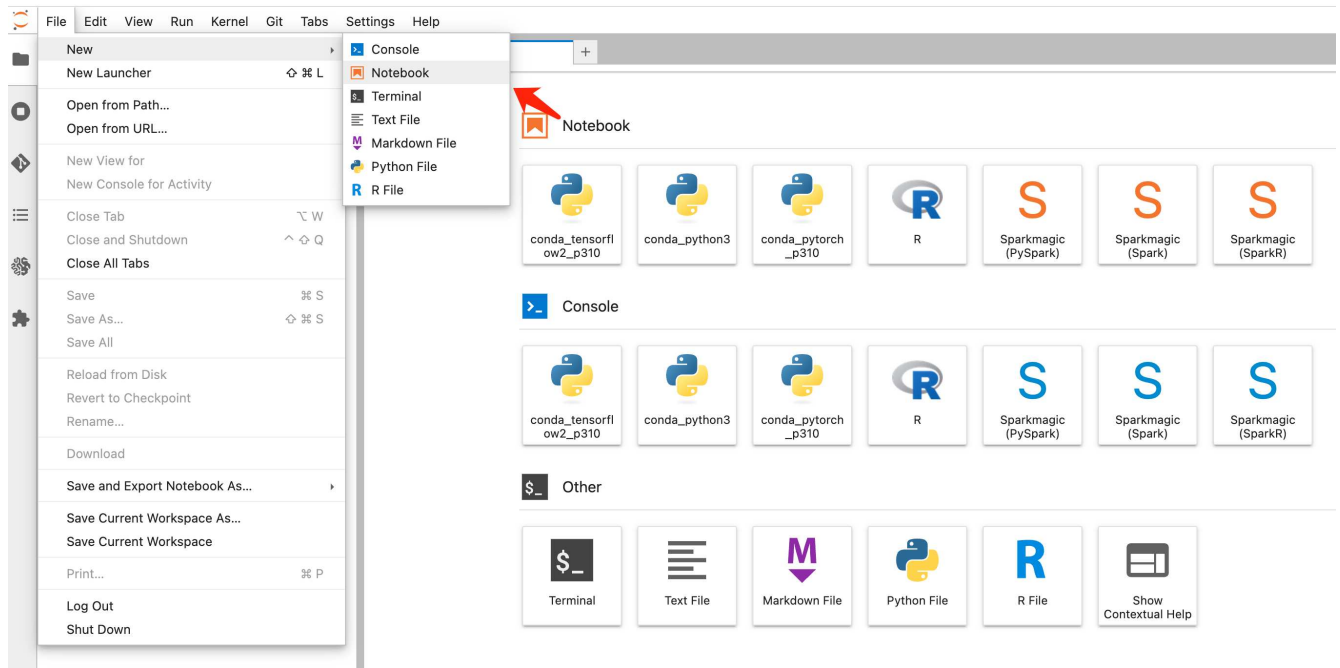
vserver object-store-server user show

```

8. 保留端點 IP 和認證、以供未來使用。



1. 在 SageMaker Notebook 執行個體中、建立新的 Jupyter 筆記本。



2. 使用下列程式碼做為解決方案的解決方案、將檔案上傳至 FSxN 私有 S3 儲存區。
如需完整的程式碼範例、請參閱本筆記型電腦。

"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']

```

FSxN 與 SageMaker 執行個體之間的整合到此結束。

實用的除錯檢查清單

- 確保 SageMaker 筆記型電腦執行個體和 FSxN 檔案系統位於同一個 VPC 中。

- 請記得在 ONTAP 上執行 **set dev** 命令、將權限等級設為 **dev** 。

常見問題集（從 2023 年 9 月 27 日起）

問：為什麼在上傳檔案至 FSxN 時、我會收到錯誤「* An error occurred (NotImplemented) when calling the CreateMultipartUpload operation: The S3 command you requested is not Implemate*」（* 發生錯誤（未實作））？

答：作為私有 S3 儲存區、FSxN 支援上傳高達 100MB 的檔案。使用 S3 傳輸協定時、大於 100MB 的檔案會分成 100MB 的區塊、並會呼叫「CreateMultipartUpload」功能。不過、FSxN Private S3 目前的實作並不支援此功能。

問：為什麼在將檔案上傳至 FSxN 時、我在呼叫 PutObject 作業時收到錯誤「* 發生錯誤（AccessDenied）」（* 發生錯誤（AccessDenied））？

答：若要從 SageMaker 筆記型電腦執行個體存取 FSxN 私有 S3 儲存區、請將 AWS 認證切換至 FSxN 認證。不過、若要授予執行個體寫入權限、則需要一種因應措施解決方案、其中包括掛載儲存區、並執行「chmod」Shell 命令來變更權限。

問：如何將 FSxN 私有 S3 儲存區與其他 SageMaker ML 服務整合？

答：很抱歉、SageMaker 服務 SDK 無法提供方法來指定私有 S3 儲存區的端點。因此、FSxN S3 與 SageMaker 服務不相容、例如 Sagemaker Data Wrangler、Sagemaker、SIGemaker、Sagemaker Glue、Sagemaker Athena、Sagemaker AutoML、及其他。

第 2 部分：運用 AWS FSx for NetApp ONTAP（FSxN）做為 SageMaker 模型訓練的資料來源

- 作者：*
NetApp 資深資料與應用科學家 Jian Jian（Ken）

簡介

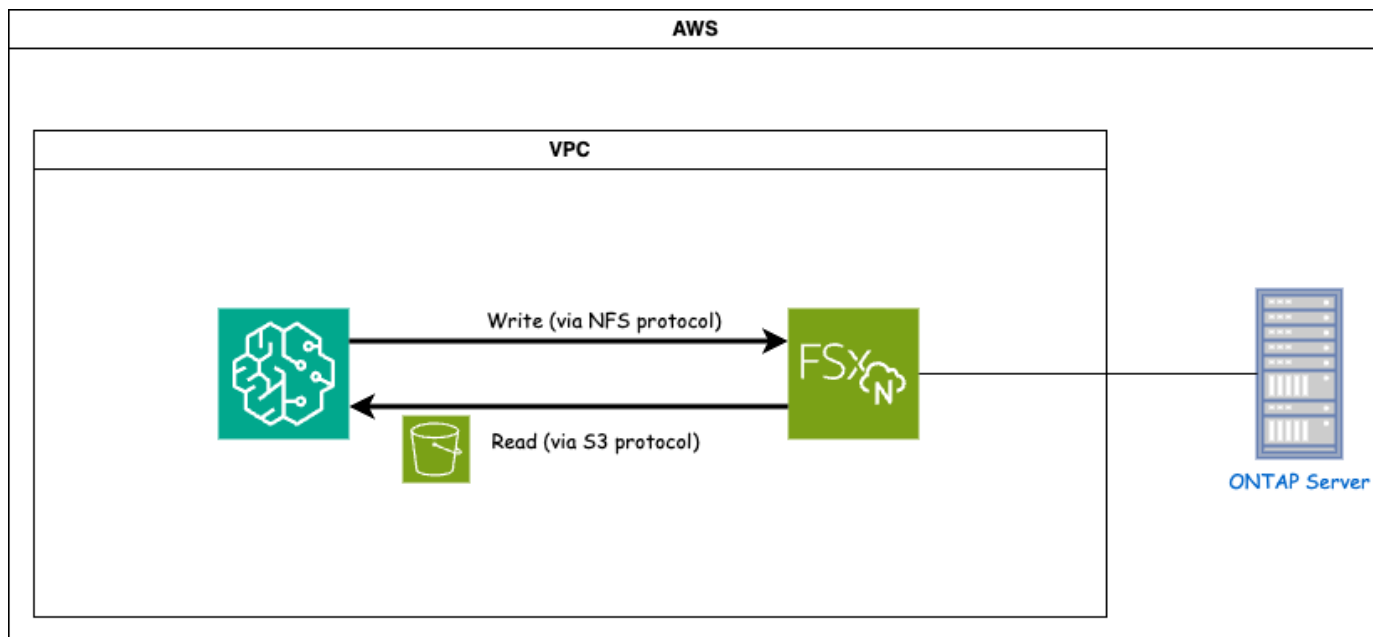
本教學課程提供電腦視覺分類專案的實用範例、提供建置 ML 模型的實際經驗、這些模型將 FSxN 作為 SageMaker 環境中的資料來源。此專案著重於使用深層學習架構 PyTorch、根據輪胎影像來分類輪胎品質。它強調使用 FSxN 做為 Amazon SageMaker 資料來源的機器學習模型開發。

什麼是 FSxN

Amazon FSx for NetApp ONTAP 確實是 AWS 提供的完全託管儲存解決方案。它利用 NetApp 的 ONTAP 檔案系統來提供可靠且高效能的儲存設備。支援 NFS、SMB 和 iSCSI 等傳輸協定、可從不同的運算執行個體和容器進行無縫存取。此服務旨在提供卓越的效能、確保快速且有效率的資料作業。它也提供高可用度和耐用度、確保資料保持可存取和保護的狀態。此外、Amazon FSx for NetApp ONTAP 的儲存容量可擴充、可讓您根據自己的需求輕鬆調整。

先決條件

網路環境



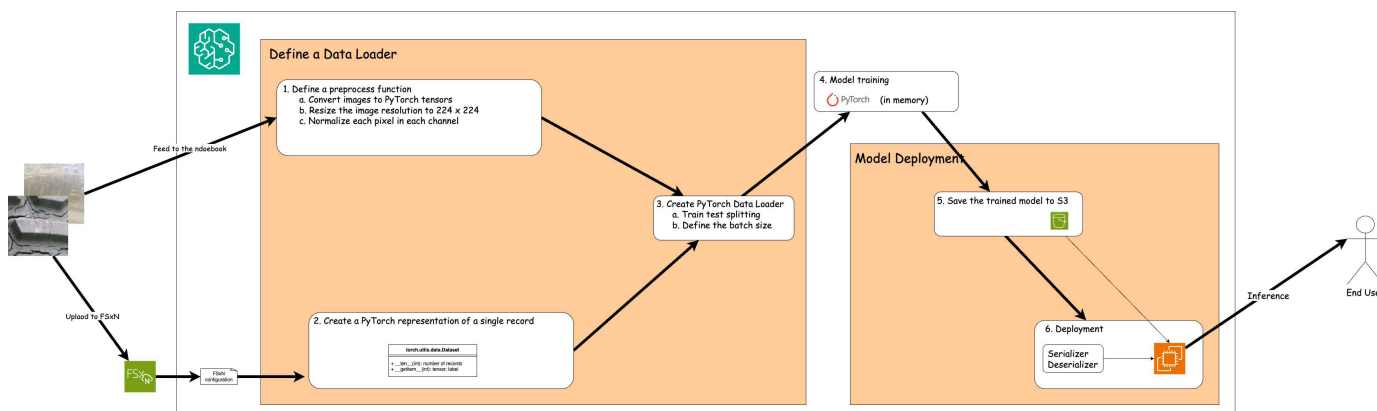
FSxN (Amazon FSx for NetApp ONTAP) 是 AWS 儲存服務。其中包括在 NetApp ONTAP 系統上執行的檔案系統、以及連接到它的 AWS 託管系統虛擬機器 (SVM)。在所提供的圖表中、AWS 管理的 NetApp ONTAP 伺服器位於 VPC 之外。SVM 是 SageMaker 和 NetApp ONTAP 系統之間的中介、可接收來自 SageMaker 的作業要求、並將其轉送至基礎儲存設備。若要存取 FSxN、SageMaker 必須與 FSxN 部署位於同一個 VPC 內。此組態可確保 SageMaker 和 FSxN 之間的通訊和資料存取。

資料存取

在實際案例中、資料科學家通常會利用儲存在 FSxN 中的現有資料來建置機器學習模型。不過、為了進行示範、由於 FSxN 檔案系統在建立後一開始是空的、因此必須手動上傳訓練資料。這可以透過將 FSxN 當作 Volume 安裝到 SageMaker 來達成。檔案系統成功掛載後、您可以將資料集上傳至掛載位置、以便在 SageMaker 環境中訓練您的模型。此方法可讓您在與 SageMaker 合作進行模型開發與訓練時、善用 FSxN 的儲存容量與功能。

資料讀取程序包括將 FSxN 設定為私有 S3 儲存區。若要瞭解詳細的組態指示、請參閱 ["第 1 部分：將 AWS FSx for NetApp ONTAP \(FSxN\) 整合為私有 S3 儲存區、並整合至 AWS SageMaker"](#)

整合概述



使用 FSxN 中的訓練資料在 SageMaker 中建立深度學習模型的工作流程、可歸納為三個主要步驟：資料載入程式定義、模型訓練和部署。在高層級、這些步驟是 MLOps 管道的基礎。不過、每個步驟都包含數個詳細的子步驟、以便全面實作。這些子步驟涵蓋各種工作、例如資料預先處理、資料集分割、模型組態、超參數調整、模型評估、和模型部署。這些步驟可確保在 SageMaker 環境中、使用 FSxN 的訓練資料、建立及部署深度學習模型

的完整且有效的程序。

逐步整合

資料載入器

為了訓練具有資料的 PyTorch 深度學習網路、我們建立了一個資料載入器、以利資料的輸入。資料載入程式不僅會定義批次大小、也會決定在批次中讀取及預先處理每筆記錄的程序。藉由設定資料載入器、我們可以分批處理資料處理、以便訓練深度學習網路。

資料載入器包含 3 個部分。

預處理功能

```
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]
    )
])
```

上述程式碼片段示範使用 * torchvision.transformation* 模組來定義影像處理前轉換。在此轉換中、會建立處理前物件以套用一系列的轉換。首先， **ToTensor()** 轉換會將影像轉換成張量表示法。隨後， **Resize(224,224)** 轉換會將影像調整為 224 x 224 像素的固定大小。最後、* 標準化 () * 轉換會減去平均值、並除以每個通道的標準差、以標準化張量值。用於正規化的平均和標準差值通常用於預先訓練的神經網路模型。整體而言、此程式碼會將影像資料轉換成張量、調整大小、以及將像素值正規化、以準備影像資料、以便進一步處理或輸入到預先訓練的模型中。

PyTorch Dataset Class

```

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

```

此類別提供的功能可取得資料集中的記錄總數、並定義讀取每筆記錄資料的方法。在 `getertis_` 函數中，代碼使用 boto3 S3 儲存區物件從 FSxN 擷取二進位資料。從 FSxN 存取資料的程式碼樣式類似於從 Amazon S3 讀取資料。隨後的說明將深入到私有 S3 物件 * 儲存庫 * 的建立程序。

FSxN 做為私有 S3 儲存庫


```

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 ---

```

若要從 SageMaker 中的 FSxN 讀取資料、會建立一個處理常式、使用 S3 傳輸協定指向 FSxN 儲存設備。如此可將 FSxN 視為私有 S3 儲存區。處理常式組態包括指定 FSxN SVM 的 IP 位址、貯體名稱和必要的認證。如需取得這些組態項目的完整說明、請參閱上的文件 ["第 1 部分：將 AWS FSx for NetApp ONTAP \(FSxN\) 整合為私有 S3 儲存區、並整合至 AWS SageMaker"](#)。

在上述範例中、貯體物件用於產生 PyTorch 資料集物件。後續章節將進一步說明 DataSet 物件。

PyTorch Data Loader

```

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)

```

在所提供的範例中、會指定 64 個批次大小、表示每個批次將包含 64 個記錄。結合 PyTorch * Dataset* 課程、預處理功能和訓練批次大小、我們獲得訓練用的資料載入器。此資料載入器可協助在訓練階段中分批重複資料集的程序。

示範訓練

```

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')

```

此程式碼可實作標準的 PyTorch 訓練程序。它定義了一個稱為 **TyreQualityClassifier** 的神經網路模型、使用卷積層和線性層來分類輪胎品質。訓練循環會反覆循環資料批次、計算遺失、並使用反向傳播和最佳化來更新模型參數。此外、它會列印目前時間、時期、批次和遺失、以供監控。

建構部署模式

部署

```

import io
import os
import tarfile
import sagemaker

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

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

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

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

```

此程式碼會將 PyTorch 模型儲存至 * Amazon S2* 、因為 SageMaker 需要將模型儲存在 S3 中以進行部署。將模型上傳至 * Amazon S2* 、即可讓 SageMaker 存取、讓部署模型的部署和推斷得以實現。

```

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
)

```

此程式碼有助於在 SageMaker 上部署 PyTorch 模型。它定義了自訂序列化器 *TyreQualitySerializer*、可將輸入資料預先處理並序列化為 PyTorch Tensor。*TyreQualityPredictor* 類是一種自定義的謂詞，它使用定義的序列化器和 *JSONDeserializer*。程式碼也會建立一個 *PyTorchModel* 物件、以指定模型的 S3 位置、IAM 角色、架構版本和推斷的進入點。程式碼會產生時間戳記、並根據模型和時間戳記來建構端點名稱。最後、使用部署方法來部署模型、指定執行個體數、執行個體類型和產生的端點名稱。如此一來、即可部署並存取 PyTorch 模型、以供 SageMaker 的推斷。

推斷

```
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)
```

這是使用已部署端點進行推斷的範例。

第 3 部分：建立簡化的 MLOps 管道（CI/CT/CD）

- 作者：*
NetApp 資深資料與應用科學家 Jian Jian（Ken）

簡介

在本教學課程中、您將學習如何運用各種 AWS 服務來建構簡單的 MLOps 管道、其中包括持續整合（CI）、持續訓練（CT）和持續部署（CD）。與傳統的 DevOps 管線不同、MLOps 需要額外考量才能完成營運週期。在本教學課程之後、您將深入瞭解如何將 CT 整合至 MLOps 迴圈、以便持續訓練您的模型、並進行無縫部署以進行推斷。本教學課程將引導您完成使用 AWS 服務建立此端點對端點 MLOps 管道的程序。

資訊清單

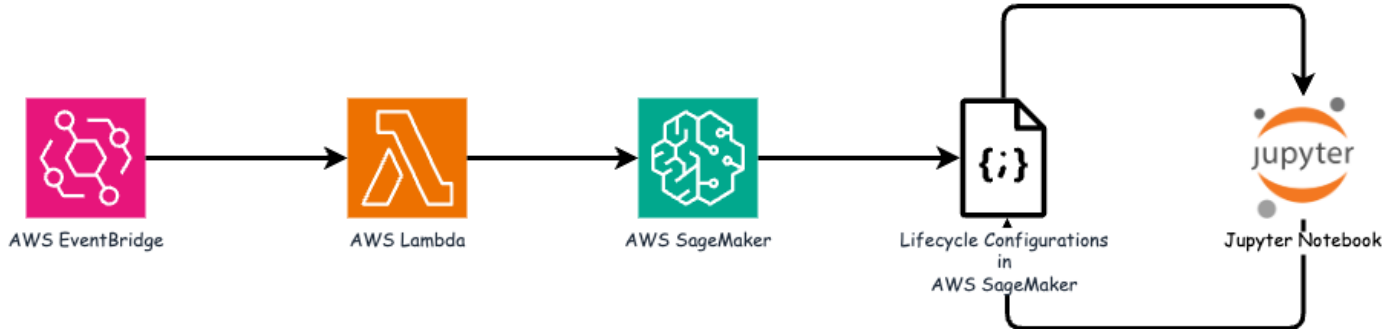
功能	名稱	留言
資料儲存	AWS FSxN	請參閱 "第 1 部分：將 AWS FSx for NetApp ONTAP（FSxN）整合為私有 S3 儲存區、並整合至 AWS SageMaker" 。
資料科學 IDE	AWS SageMaker	本教學課程是以 Jupyter 筆記型電腦為基礎、如所示 "第 2 部分：運用 AWS FSx for NetApp ONTAP（FSxN）做為 SageMaker 模型訓練的資料來源" 。
觸發 MLOps 管道的功能	AWS Lambda 功能	-
cron 工作觸發程式	AWS EventBridge	-
深度學習架構	PyTorch	-
AWS Python SDK	boto3.	-
程式設計語言	Python	v3.10.

先決條件

- 預先設定的 FSxN 檔案系統。本教學課程將 FSxN 中儲存的資料用於訓練程序。

- * SageMaker 筆記型電腦執行個體 *、設定為與上述 FSxN 檔案系統共用相同的 VPC。
- 在觸發 *AWS Lambda 功能* 之前、請確定 *SageMaker 筆記型電腦執行個體* 處於 *停止* 狀態。
- 需要 * ml.g4dn.xlarg* 執行個體類型才能運用深度神經網路運算所需的 GPU 加速。

架構



此 MLOps 管道是一種實際實作、利用 cron 工作觸發無伺服器功能、進而執行以生命週期回撥功能登錄的 AWS 服務。**AWS EventBridge** 充當 cron 工作。它會定期調用一個負責重新訓練和重新部署模型的 *AWS Lambda 功能*。此程序涉及將 **AWS SageMaker Notebook** 執行個體旋轉、以執行必要的工作。

逐步組態

生命週期組態

若要設定 AWS SageMaker 筆記型電腦執行個體的生命週期回撥功能、您可以使用 *生命週期組態*。此服務可讓您定義在啟動筆記型電腦執行個體時要執行的必要動作。具體而言、您可以在 *生命週期組態* 中實作 Shell 指令碼、以便在訓練與部署程序完成後、自動關閉筆記型電腦執行個體。這是必要的組態、因為成本是 MLOps 的主要考量之一。

請務必注意 *生命週期組態* 的組態必須事先設定。因此、建議您在繼續進行其他 MLOps 管道設定之前、先優先設定此方面的設定。

1. 若要設定生命週期組態、請開啟 **Sagemaker** 面板、然後瀏覽至 *管理組態* 一節下的 *生命週期組態*。

aws

Services

Q Search

S3

Amazon SageMaker

×

Getting started

Studio

Studio Lab

Canvas

RStudio

TensorBoard

Profiler

▼ Admin configurations

Domains

Role manager

Images

Lifecycle configurations

SageMaker dashboard

Search

► JumpStart

Amazon SageMaker > Domains

Domains

Info

A domain includes an associated Amazon SageMaker notebook instance. Each domain receives a personal and private Amazon SageMaker notebook instance.

► Domain structure diagram

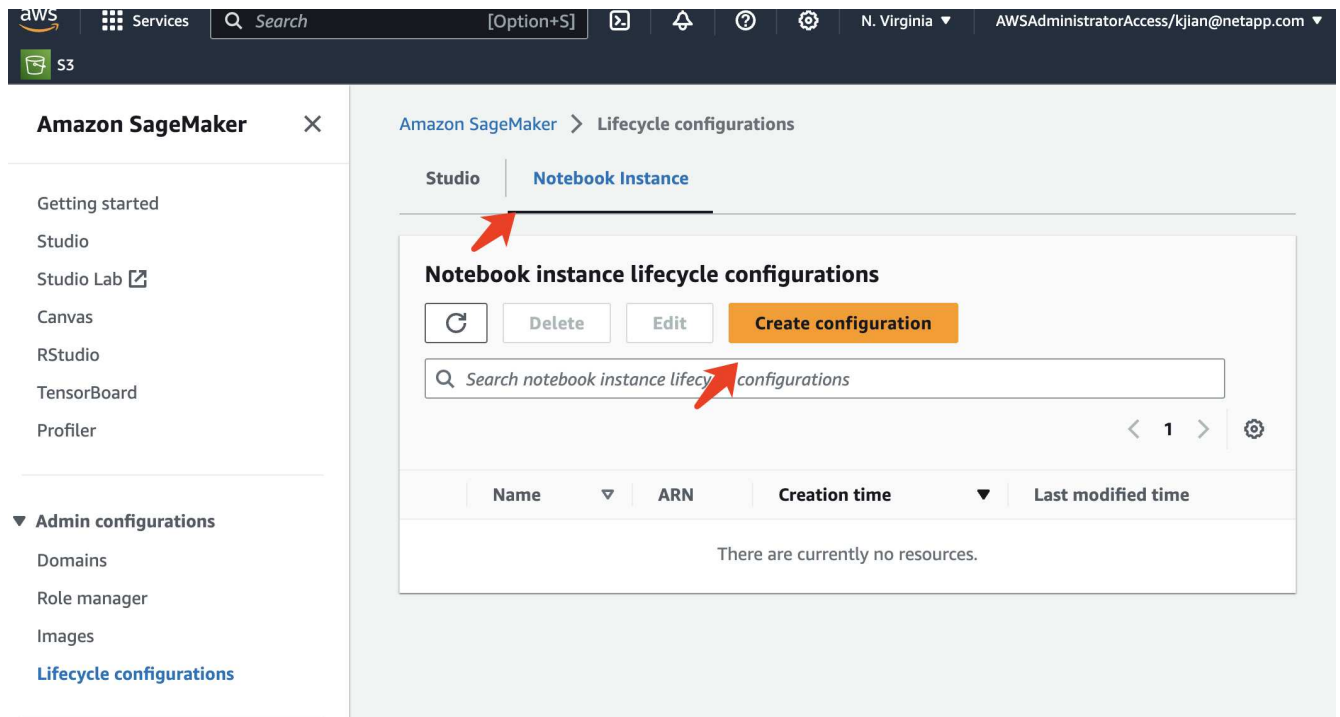
Domains (4)

Info

Q Find domain name

	Name	
<input type="radio"/>	rdsml-east-1	
<input type="radio"/>	rdsml-east-2	
<input type="radio"/>	rdsml-east-3	
<input type="radio"/>	rdsml-east-4	

2. 選取 * 筆記型電腦執行個體 * 標籤、然後按一下 * 建立組態 * 按鈕



3. 將下列程式碼貼到輸入區。

```
#!/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. 此指令碼會執行 Jupyter Notebook 、以處理模型的重新訓練和重新部署以供推斷。執行完成後、筆記型電腦將在 5 分鐘內自動關機。若要深入瞭解問題陳述和程式碼實作、請參閱 ["第 2 部分：運用 AWS FSx for NetApp ONTAP \(FSxN\) 做為 SageMaker 模型訓練的資料來源"](#)。

aws Services Search [Option+S]

S3

Amazon SageMaker > Lifecycle configurations > Create lifecycle configuration

Create lifecycle configuration

Configuration setting

Name

fsxn-demo-lifecycle-callback

Alphanumeric characters and "-", no spaces. Maximum 63 characters.

Scripts

Start notebook Create notebook

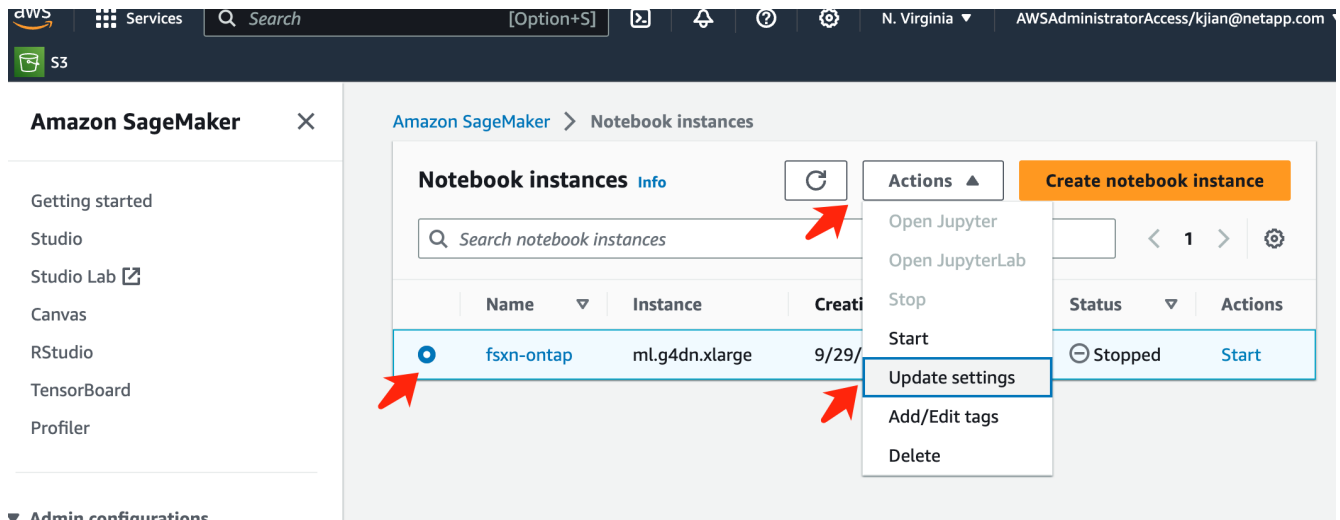
This script will be run each time an associated notebook instance is started, including during initial creation. If the associated notebook instance is already started, it will be run the next time it is stopped and started. [a curated list of sample scripts](#)

```
1 #!/bin/bash
2
3 set -e
4 sudo -u ec2-user -i <<'EOF'
5 # 1. Retraining and redeploying the model
6 NOTEBOOK_FILE=/home/ec2-user/SageMaker/tyre_quality_classification_local_training.ipynb
7 echo "Activating conda env"
8 source /home/ec2-user/anaconda3/bin/activate pytorch_p310
9 nohup jupyter nbconvert "$NOTEBOOK_FILE" --ExecutePreprocessor.kernel_name=python --execute --to nbconvert_pid=$!
10 nbconvert_pid=$!
11 conda deactivate
12
13 # 2. Scheduling a job to shutdown the notebook to save the cost
14 PYTHON_DIR=/home/ec2-user/anaconda3/envs/JupyterSystemEnv/bin/python3.10
15 echo "Starting the autostop script in cron"
16 (crontab -l 2>/dev/null; echo "*/5 * * * * bash -c 'if ps -p $nbconvert_pid > /dev/null; then echo
17 EOF
```

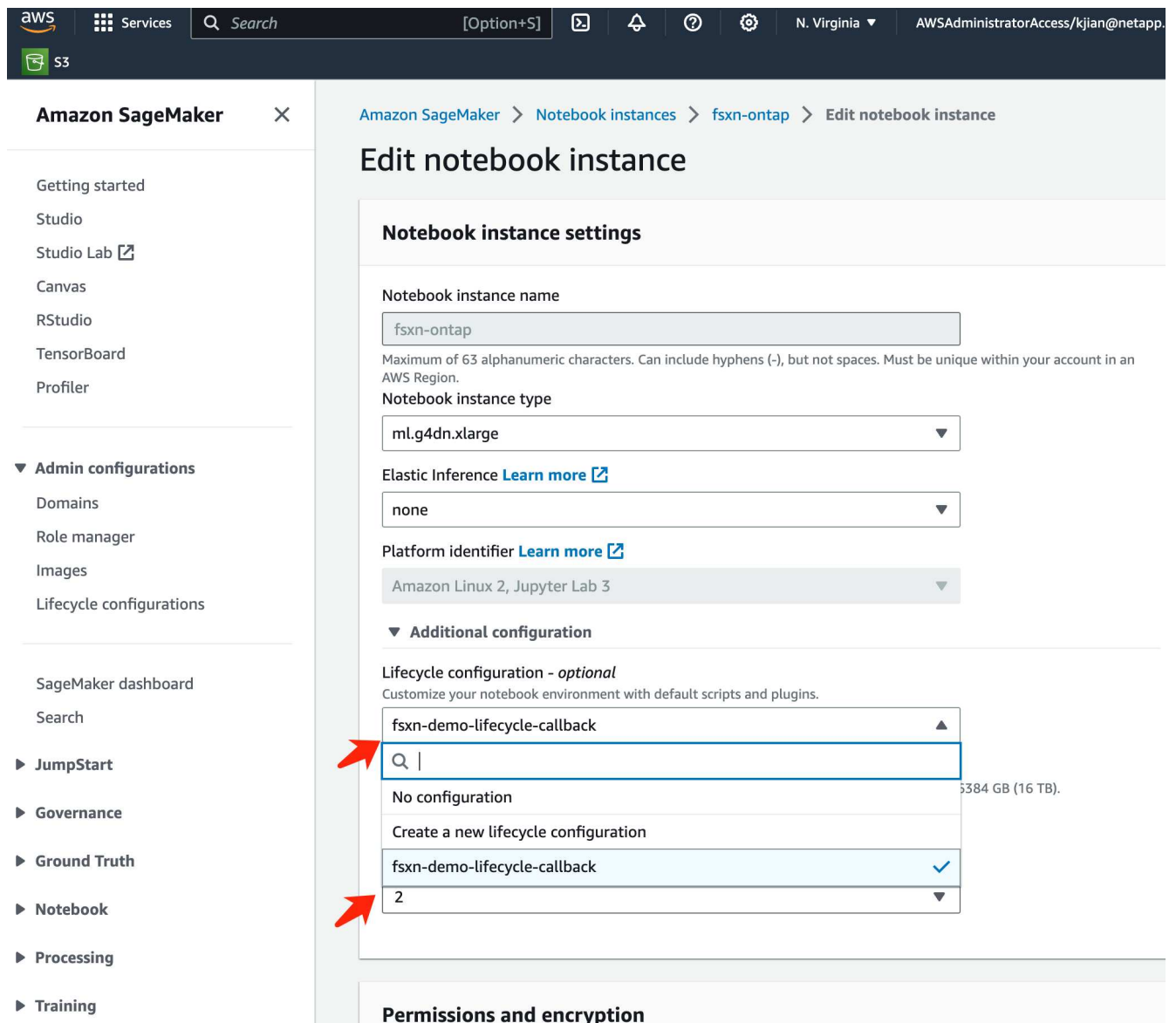
Cancel Create configuration

CloudShell Feedback

5. 建立之後、請瀏覽至「筆記本執行個體」、選取目標執行個體、然後按一下「動作」下拉式清單下的「* 更新設定 *」。



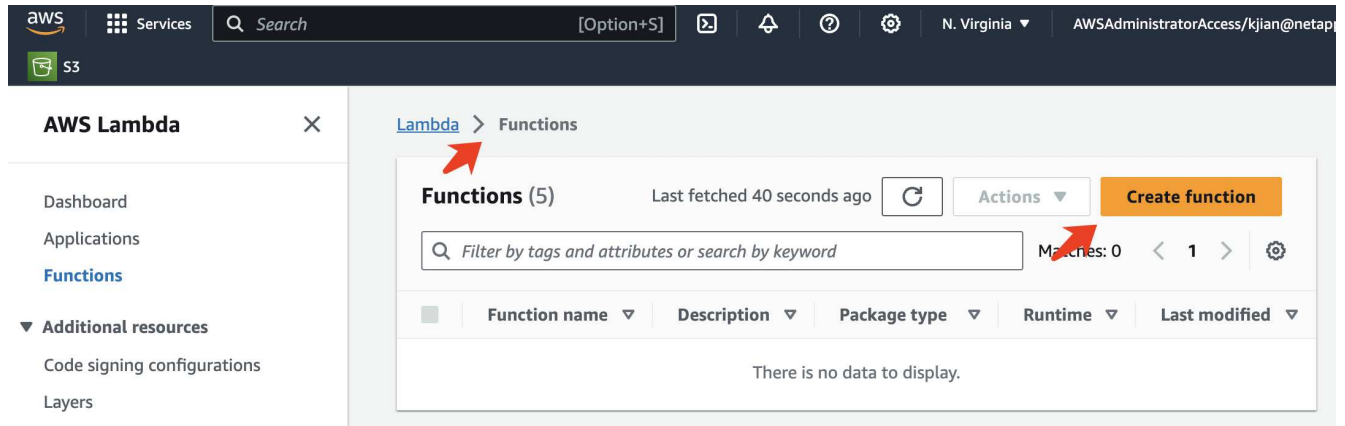
6. 選取建立的 * 生命週期組態 * 、然後按一下 * 更新筆記型電腦執行個體 * 。



AWS Lambda 無伺服器功能

如前所述、* AWS Lambda Funct* 負責加速 *AWS SageMaker 筆記型電腦執行個體* 。

1. 若要建立 * AWS Lambda 功能* 、請瀏覽至個別面板、切換至 * 功能* 索引標籤、然後按一下 * 建立功能* 。



2. 請將所有必要項目歸檔在頁面上、並記得將執行時間切換為 * Python 3.10* 。

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

S3

Lambda > Functions > Create function

Create function [Info](#)

AWS Serverless Application Repository applications have moved to [Create application](#).

☒ **Author from scratch**
Start with a simple Hello World example.

☐ **Use a blueprint**
Build a Lambda application from sample code and configuration presets for common use cases.

☐ **Container image**
Select a container image to deploy for your function.

Basic information

Function name
Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.
☒ x86_64
☐ arm64

Permissions [Info](#)
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

3. 請驗證指定的角色是否具有所需的權限 **amazonSageMakerFullAccess** 、然後按一下 * 建立功能 * 按鈕。

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

S3

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.10

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.
☒ x86_64
☐ arm64

Permissions [Info](#)
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ **Change default execution role**

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☐ Create a new role with basic Lambda permissions
☒ Use an existing role
☐ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
service-role/fsxn-demo-mlops-role-585jzdny
[View the fsxn-demo-mlops-role-585jzdny role](#) on the IAM console.

► **Advanced settings**

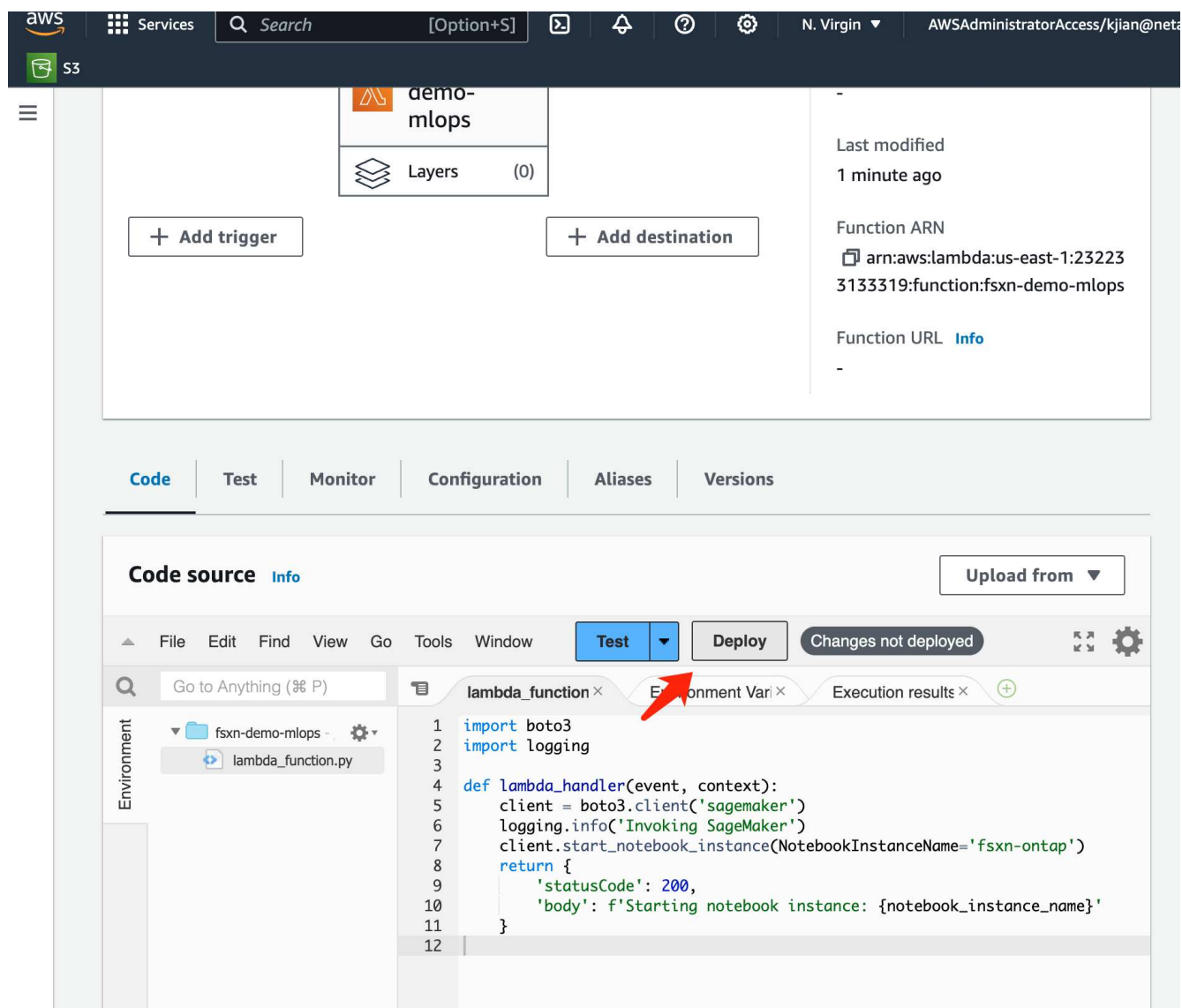
Cancel Create function

4. 選取建立的 Lambda 函數。在「程式碼」索引標籤中、複製下列程式碼並貼到文字區域。此程式碼會啟動名為 * fsxn-ONTAP* 的筆記型電腦執行個體。

```
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. 按一下 * 部署 * 按鈕以套用此程式碼變更。



6. 若要指定如何觸發 AWS Lambda 功能、請按一下「新增觸發按鈕」。

The screenshot shows the AWS Lambda console interface for a function named 'fsxn-demo-mlops'. The breadcrumb navigation at the top indicates the path: [Lambda](#) > [Functions](#) > fsxn-demo-mlops. The function name 'fsxn-demo-mlops' is prominently displayed at the top left of the main content area. To the right of the name are three buttons: 'Throttle', 'Copy ARN', and 'Actions'. Below the function name, there is a section titled 'Function overview' with an 'Info' link. This section contains a central card for the function, showing the Lambda icon, the name 'fsxn-demo-mlops', and a 'Layers' section with '(0)' layers. Below this card are two buttons: '+ Add trigger' and '+ Add destination'. A red arrow points directly to the '+ Add trigger' button. To the right of the central card, there is a sidebar with details: 'Description' (empty), 'Last modified' (2 minutes ago), 'Function ARN' (arn:aws:lambda:us-east-1:232233133319:function:fsxn-demo-mlops), and 'Function URL' (empty with an 'Info' link).

7. 從下拉式功能表中選取 EventBridge、然後按一下標有「建立新規則」的選項按鈕。在排程運算式欄位中、輸入 `rate(1 day)`、然後按一下「新增」按鈕、以建立這個新的 cron 工作規則、並將其套用至 AWS Lambda 功能。

aws Services Search [Option+S] N. Virginia AWSAdministratorAccess

S3

[Lambda](#) > Add trigger

Add trigger

Trigger configuration Info

EventBridge (CloudWatch Events)
aws asynchronous schedule management-tools

Rule
Pick an existing rule, or create a new one.

☒ Create a new rule
☐ Existing rules

Rule name
Enter a name to uniquely identify your rule.

mlops-retraining-trigger

Rule description
Provide an optional description for your rule.

Rule type
Trigger your target based on an event pattern, or based on an automated schedule.

☐ Event pattern
☒ Schedule expression

Schedule expression
Self-trigger your target on an automated schedule using [Cron or rate expressions](#). Cron expressions are in UTC.

rate(1 day)

e.g. rate(1 day), cron(0 17 ? * MON-FRI *)

Lambda will add the necessary permissions for Amazon EventBridge (CloudWatch Events) to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

Cancel Add

完成雙步驟組態後、**AWS Lambda** 功能 * 每天會啟動 ***SageMaker Notebook**、使用 **FSxN** 儲存庫中的資料執行模型重新訓練、將更新的模型重新部署至正式作業環境、並自動關閉 *SageMaker 筆記型電腦執行個體 * 以最佳化成本。如此可確保模型保持在最新狀態。

本課程將為您介紹如何開發 MLOps 管道。

採用 **Domino Data Lab** 和 **NetApp** 的混合式多雲端 MLOps

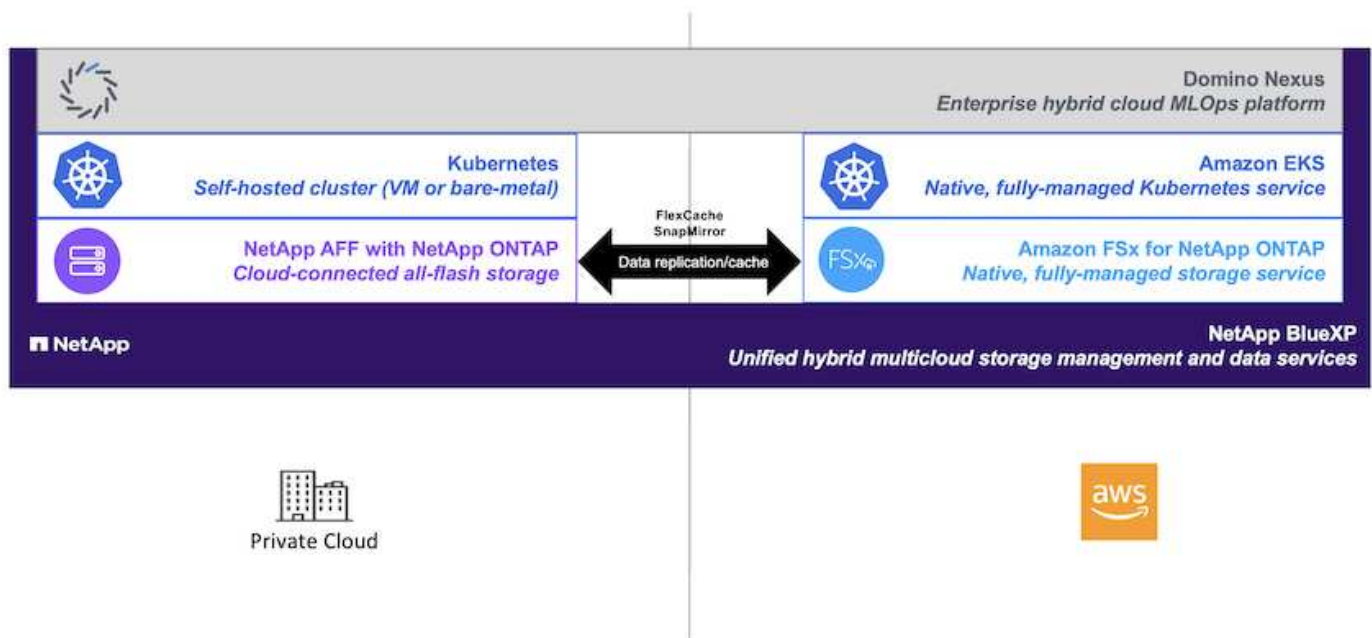
採用 Domino Data Lab 和 NetApp 的混合式多雲端 MLOps

Mike Oglesby、NetApp

全球各地的組織目前都採用 AI 來轉型業務和流程。因此、AI 就緒的運算基礎架構通常供應不足。企業正採用混合式多雲端 MLOps 架構、以便在不同區域、資料中心和雲端之間充分利用可用的運算環境、並在成本、可用度和效能之間取得平衡。

Domino Data Lab 的 Domino Nexus 是統一的 MLOps 控制平台、可讓您在任何雲端、區域或內部部署的任何運算叢集上執行資料科學和機器學習工作負載。它將整個企業的資料科學封閉環境統一起來、讓您有一個地方可以建置、部署和監控模型。同樣地、無論資料在何處執行、NetApp 的混合雲資料管理功能都能讓您將資料帶到工作和工作空間。當您將 Domino Nexus 與 NetApp 配對時、您可以靈活地在不同環境中排程工作負載、而無需擔心資料可用性。換句話說、您可以將工作負載和資料傳送到適當的運算環境、讓您在瀏覽資料隱私權和主權相關法規的同時、加速 AI 部署。

此解決方案展示了統一的 MLOps 控制平面的部署、其中包含內部部署的 Kubernetes 叢集、以及在 Amazon Web Services (AWS) 中執行的 Elastic Kubernetes Service (EKS) 叢集。



技術總覽

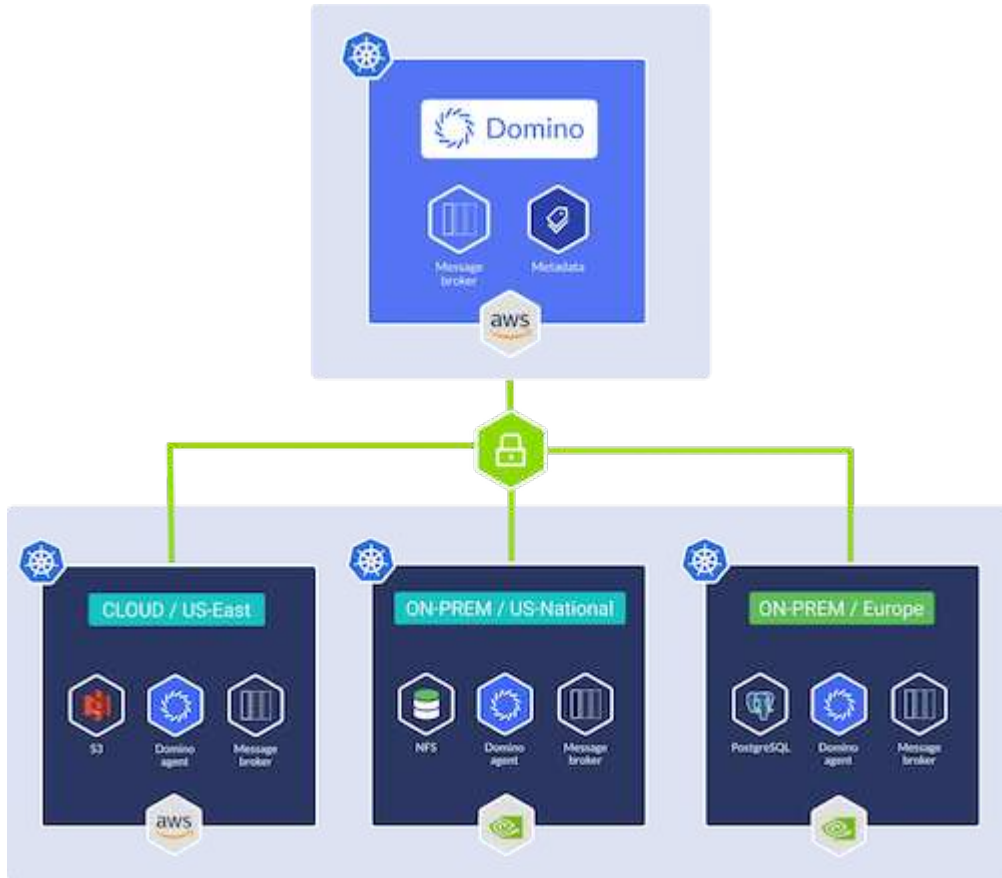
Domino Data Lab

Domino Data Lab 以領先業界的 Enterprise AI 平台為模式導向企業提供動力、這是財星百大企業中 20% 以上的企業所信賴的平台。Domino 加速資料科學工作的開發與部署、同時增加協同作業與治理。有了 Domino、全球企業就能開發更好的藥物、種植更多生產作物、打造更好的汽車等。Domino 成立於 2013 年、以 Coatue Management、Great Hill Partners、Highland Capital、Sequoia Capital 及其他主要投資者為後盾。

Domino 可讓企業及其資料科學家在統一的端對端平台上建置、部署及管理 AI、快速、負責且具成本效益。團隊可在任何環境中存取所需的所有資料、工具、運算、模型和專案、以便協同作業、重複使用過去的工作、追蹤生產模式以提高準確度、以最佳實務標準化、並使 AI 負責並加以管理。

- * 開放且靈活：* 存取最廣泛的開放原始碼和商業工具生態系統、以及基礎架構、以獲得最佳創新成果、而不受廠商束縛。

- * 記錄系統：* 企業中 AI 營運與知識的中央中樞、可實現最佳實務做法、跨職能協同作業、加速創新與效率。
- * 整合式：* 整合式工作流程與自動化、專為企業流程、控管與治理所打造、可滿足您的法規遵循與法規要求。
- * 混合式多雲端：* 在任何地點（內部部署、混合式、任何雲端或多雲端）、都能在接近您資料的位置執行 AI 工作負載、以降低成本、達到最佳效能並符合法規要求。



Domino Nexus

Domino Nexus 是單一窗口、可讓您在任何雲端、區域或內部部署的任何運算叢集上執行資料科學和機器學習工作負載。它將整個企業的資料科學封閉環境統一起來、讓您有一個地方可以建置、部署和監控模型。

NetApp BlueXP

NetApp BlueXP 將所有 NetApp 的儲存與資料服務統一化為單一工具、讓您建置、保護及管理混合式多雲端資料資產。它為內部部署和雲端環境的儲存和資料服務提供統一的體驗、並透過 AIOps 的強大功能、提供靈活的使用參數和整合式保護功能、讓營運變得更簡單。

NetApp ONTAP

NetApp最新一代的儲存管理軟體、即支援企業將基礎架構現代化、並移轉至雲端就緒的資料中心。ONTAP利用領先業界的資料管理功能ONTAP、無論資料位於何處、只要使用一組工具、即可管理及保護資料。您也可以自由地將資料移至任何需要的位置：邊緣、核心或雲端。支援眾多功能、可簡化資料管理、加速及保護關鍵資料、並在混合雲架構中提供新一代基礎架構功能。ONTAP

簡化資料管理

資料管理對於企業IT營運和資料科學家而言至關重要、因此可將適當的資源用於AI應用程式和訓練AI/ML資料集。下列關於NetApp技術的其他資訊超出此驗證範圍、但可能會因您的部署而有所差異。

包含下列功能的資料管理軟體、可簡化及簡化作業、並降低您的總營運成本：ONTAP

- 即時資料精簡與擴充重複資料刪除技術。資料壓縮可減少儲存區塊內的空間浪費、重複資料刪除技術可大幅提升有效容量。這適用於本機儲存的資料、以及分層至雲端的資料。
- 最低、最大及可調適的服務品質（AQO）。精細的服務品質（QoS）控制有助於維持高共享環境中關鍵應用程式的效能等級。
- NetApp FabricPool自動將冷資料分層至公有和私有雲端儲存選項、包括Amazon Web Services（AWS）、Azure和NetApp StorageGRID 等儲存解決方案。如需FabricPool 更多有關資訊、請參閱 ["TR-4598：FabricPool 最佳實務做法"](#)。

加速並保護資料

提供優異的效能與資料保護、並以下列方式擴充這些功能：ONTAP

- 效能與較低的延遲。以最低的延遲提供最高的處理量。ONTAP
- 資料保護：支援所有平台的通用管理功能、可提供內建的資料保護功能。ONTAP
- NetApp Volume Encryption（NVE）。支援內建和外部金鑰管理、提供原生Volume層級的加密功能。ONTAP
- 多租戶和多因素驗證。支援以最高安全等級共享基礎架構資源。ONTAP

符合未來需求的基礎架構

下列功能可協助滿足嚴苛且不斷變化的業務需求：ONTAP

- 無縫擴充與不中斷營運。支援在不中斷營運的情況下、將容量新增至現有控制器和橫向擴充叢集。ONTAP客戶可以升級至最新技術、例如NVMe和32GB FC、而不需進行昂貴的資料移轉或中斷運作。
- 雲端連線：ONTAP 是最具雲端連線能力的儲存管理軟體、可在所有公有雲中選擇軟體定義儲存設備和雲端原生執行個體。
- 與新興應用程式整合。利用支援現有企業應用程式的相同基礎架構、為新一代平台和應用程式提供企業級資料服務、例如自動駕駛車輛、智慧城市和產業4.0。ONTAP

Amazon FSX for NetApp ONTAP 產品

Amazon FSX for NetApp ONTAP 是第一方、完全託管的 AWS 服務、可提供高度可靠、可擴充、高效能及功能豐富的檔案儲存設備、以 NetApp 熱門的 ONTAP 檔案系統為基礎。FSX for ONTAP VMware將NetApp檔案系統熟悉的功能、效能、功能和API作業、以及完全託管的AWS服務的敏捷度、擴充性和簡易性結合在一起。

NetApp Astra Trident

Astra Trident 可在所有常見的 NetApp 儲存平台、公有雲或內部部署（包括 ONTAP（AFF、FAS、Select、Cloud、Amazon FSX for NetApp ONTAP）、Element 軟體（NetApp HCI、SolidFire）、Azure NetApp Files 服務、以及 Google Cloud 上的 Cloud Volumes Service。Astra Trident 是符合 Container Storage Interface（CSI）規範的動態儲存協調器、可與 Kubernetes 原生整合。

Kubernetes

Kubernetes是開放原始碼的分散式容器協調平台、最初由Google設計、現在由Cloud Native Computing Foundation（CNCF）維護。Kubernetes 可為容器化應用程式自動化部署、管理及擴充功能、是企業環境中最主要的容器協調作業平台。

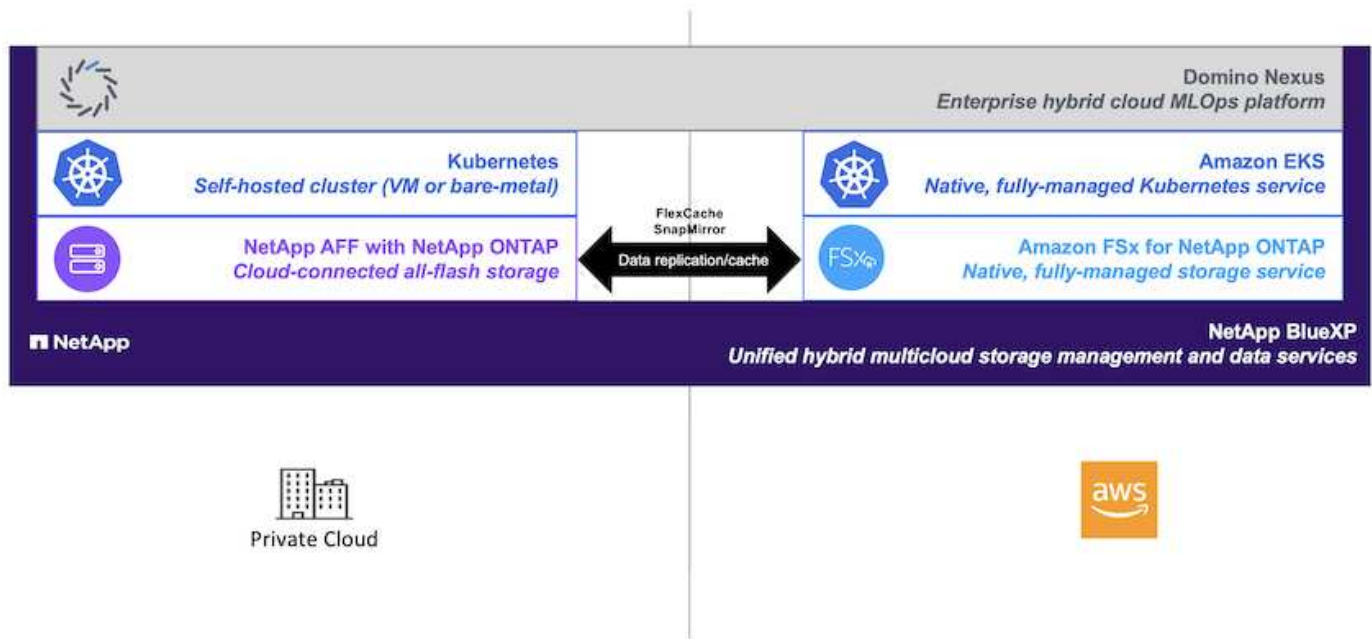
Amazon Elastic Kubernetes Service（EKS）

Amazon Elastic Kubernetes Service（Amazon EKS）是 AWS 雲端中的管理 Kubernetes 服務。Amazon EKS 可自動管理 Kubernetes 控制平面節點的可用度與擴充性、這些節點負責排程容器、管理應用程式可用度、儲存叢集資料及其他重要工作。有了 Amazon EKS、您就能充分利用 AWS 基礎架構的所有效能、規模、可靠性和可用度、以及與 AWS 網路和安全服務的整合。

架構

此解決方案結合 Domino Nexus 的混合式多雲端工作負載排程功能與 NetApp 資料服務、打造統一的混合雲 MLOps 平台。如需詳細資料、請參閱下表。

元件	名稱	環境
MLOps 控制平面	"採用 Domino Nexus 的 Domino Enterprise AI 平台"	AWS
MLOps 平台運算環境	"Domino Nexus Data Planes"	AWS、內部部署資料中心
內部部署運算平台	"Kubernetes" 與 "NetApp Astra Trident"	內部部署資料中心
雲端運算平台	"Amazon Elastic Kubernetes Service（EKS）" 與 "NetApp Astra Trident"	AWS
內部部署資料平台	"NetApp 儲存設備" 技術支援 "NetApp ONTAP"	內部部署資料中心
雲端資料平台	"Amazon FSX for NetApp ONTAP 產品"	AWS



初始設定

本節說明在結合內部部署資料中心和 AWS 的混合式環境中、為了將 Domino Nexus 與 NetApp 資料服務搭配使用而需要執行的初始設定工作。

先決條件

在您執行本節概述的步驟之前、我們假設您已執行下列工作：

- 您已部署並設定內部部署的 NetApp ONTAP 儲存平台。如需詳細資訊、請參閱 ["NetApp 產品文件"](#)。
- 您已在 AWS 中配置 Amazon FSx for NetApp ONTAP 執行個體。如需詳細資訊、請參閱 ["Amazon FSx for NetApp ONTAP 產品頁面"](#)。
- 您已在內部部署資料中心內配置 Kubernetes 叢集。如需詳細資訊、請參閱 ["Domino 管理指南"](#)。
- 您已在 AWS 中佈建 Amazon EKS 叢集。如需詳細資訊、請參閱 ["Domino 管理指南"](#)。
- 您已在內部部署的 Kubernetes 叢集中安裝 NetApp Astra Trident。此外、您已將此 Trident 執行個體設定為在資源配置和管理時、使用內部部署的 NetApp ONTAP 儲存平台。如需詳細資訊、請參閱 ["NetApp Astra Trident 文件"](#)。
- 您已在 Amazon EKS 叢集中安裝 NetApp Astra Trident。此外、您已將此 Trident 執行個體設定為在資源配置和管理儲存資源時、使用 Amazon FSx for NetApp ONTAP 執行個體。如需詳細資訊、請參閱 ["NetApp Astra Trident 文件"](#)。
- 您必須在內部部署資料中心和 AWS 中的虛擬私有雲（VPC）之間建立雙向網路連線。如需實作此選項的詳細資訊、請參閱 ["Amazon 虛擬私有網路（VPN）文件"](#)。

在 AWS 中安裝 Domino Enterprise AI Platform

若要在 AWS 中安裝 Domino Enterprise MLOps Platform、請遵循中所述的指示 ["Domino 管理指南"](#)。您必須將 Domino 部署在先前佈建的同一個 Amazon EKS 叢集中。此外、NetApp Astra Trident 必須已在此 EKS 叢集中安裝及設定、而且您必須將 Trident 託管儲存類別指定為 Domino.yml 安裝組態檔案中的共用儲存類別。



請參閱 ["Domino 安裝組態參考指南"](#) 如需如何在 domino.yml 安裝組態檔案中指定共用儲存類別的詳細資訊、



["技術報告 TR-4952."](#) 逐步在 AWS 中部署 Domino、搭配 Amazon FSX for NetApp ONTAP、是疑難排解任何問題的實用參考資料。

啟用 Domino Nexus

接下來、您必須啟用 Domino Nexus。請參閱 ["Domino 管理指南"](#) 以取得詳細資料。

在內部部署資料中心部署 Domino Data Plane

接下來、您必須在內部部署資料中心部署 Domino Data Plane。您必須將此資料層部署在先前已配置的內部部署 Kubernetes 叢集。此外、NetApp Astra Trident 必須已在此 Kubernetes 叢集中安裝及設定。請參閱 ["Domino 管理指南"](#) 以取得詳細資料。

將現有的 NetApp Volume 暴露於 Domino 中

本節說明將現有 NetApp ONTAP NFS 磁碟區公開至 Domino MLOps 平台所需執行的工作。這些相同步驟同時適用於內部部署和 AWS。

為何要將 NetApp ONTAP Volume 暴露於 Domino 中？

將 NetApp Volume 搭配 Domino 使用可提供下列優點：

- 您可以利用 NetApp ONTAP 的橫向擴充功能、針對極為龐大的資料集執行工作負載。
- 您可以跨多個運算節點執行工作負載、而無需將資料複製到個別節點。
- 您可以利用 NetApp 的混合式多雲端資料移動與同步功能、在多個資料中心和 / 或雲端之間存取資料。
- 您希望能夠在不同的資料中心或雲端中、快速輕鬆地建立資料快取。

揭露 Astra Trident 未配置的現有 NFS Volume

如果您現有的 NetApp ONTAP NFS 磁碟區並非由 Astra Trident 配置、請遵循本小節所述的步驟。

在 Kubernetes 建立 PV 和 PVC



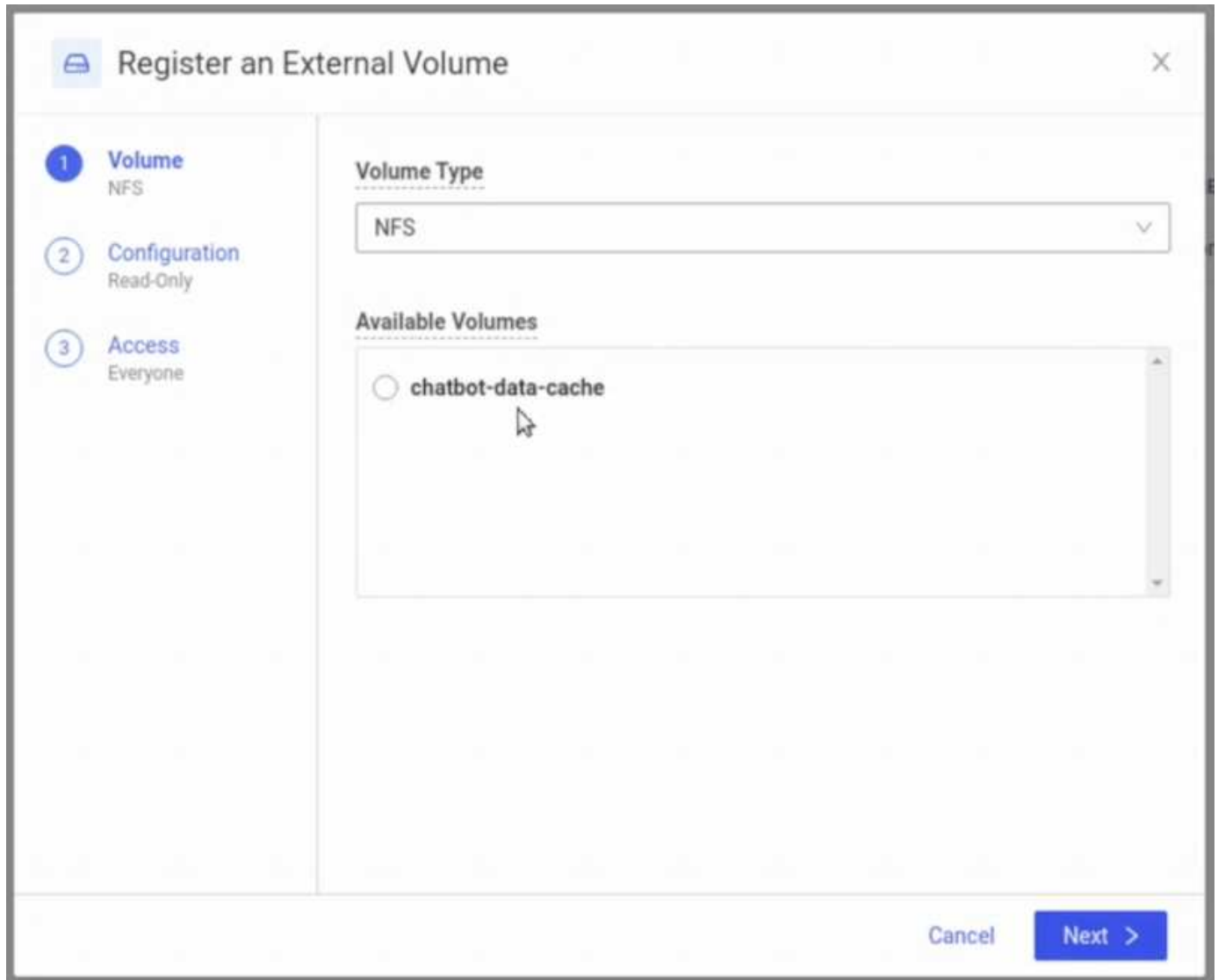
對於內部部署磁碟區、請在內部部署的 Kubernetes 叢集中建立 PV 和 PVC。對於 Amazon FSX for NetApp ONTAP Volume、請在 Amazon EKS 中建立 PV 和 PVC。

首先、您必須在 Kubernetes 叢集中建立持續性 Volume（PV）和持續性 Volume Claim（PVC）。若要建立 PV 和 PVC、請使用 ["NFS PV/PVC 範例"](#) 從 Domino 管理指南更新值、以反映您的環境。請務必為指定正確的值 namespace、nfs.path 和 nfs.server 欄位。此外、我們建議您提供 PV 和 PVC 的唯一名稱、以代表儲存在對應 ONTAP NFS Volume 上的資料性質。例如、如果 Volume 包含製造瑕疵的影像、您可以將 PV 命名為 pv-mfg-defect-images 和 PVC、pvc-mfg-defect-images。

在 Domino 中註冊外部資料 Volume

接下來、您必須在 Domino 中登錄外部資料 Volume。若要登錄外部資料 Volume、請參閱 ["說明"](#) 在 Domino 管

理指南中。登錄 Volume 時、請務必從「Volume Type」（Volume 類型）下拉式功能表中選取「NFS」（NFS）。選取「NFS」之後、您應該會在「可用磁碟區」清單中看到您的 PVC。



Register an External Volume

1 **Volume**
NFS

2 **Configuration**
Read-Only

3 **Access**
Everyone

Volume Type

NFS

Available Volumes

☒ chatbot-data-cache

Cancel Next >

揭露 Astra Trident 所佈建的現有 Volume

如果您現有的磁碟區是由 Astra Trident 配置、請遵循本子節所述的步驟。

編輯現有的 PVC

如果您的 Volume 是由 Astra Trident 配置、則您的 Volume Claim（PVC）已對應於您的 Volume。若要將此 Volume 暴露於 Domino 中、您必須編輯 PVC 並將下列標籤新增至中的標籤清單 `metadata.labels` 欄位：

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

在 Domino 中註冊外部資料 Volume

接下來、您必須在 Domino 中登錄外部資料 Volume。若要登錄外部資料 Volume、請參閱 ["說明"](#) 在 Domino 管理指南中。登錄 Volume 時、請務必從「Volume Type」（Volume 類型）下拉式功能表中選取「Generic」

(一般)。選取「一般」之後、您應該會在「可用磁碟區」清單中看到您的 PVC。

在不同環境中存取相同的資料

本節說明在不同運算環境中存取相同資料所需執行的工作。在 Domino MLOps 平台中、運算環境稱為「資料平面」。如果資料位於一個資料平面的 NetApp Volume 上、但您需要在另一個資料平面中存取、請遵循本節所述的工作。這類情境通常稱為「爆增」、或是當目的地環境是雲端時、稱為「雲端爆增」。處理受限或超額訂閱的運算資源時、通常需要此功能。例如、如果內部部署運算叢集訂閱過度、您可能會想要將工作負載排程到雲端、以便立即啟動。

存取位於不同資料平面的 NetApp Volume 有兩個建議選項。以下子節概述這些選項。根據您的特定需求、選擇其中一個選項。下表說明這兩個選項的優點和缺點。

選項	效益	缺點
選項 1 - 快取	<ul style="list-style-type: none">- 簡化工作流程- 能夠根據需求快取資料子集- 能夠將資料寫回來源- 沒有要管理的遠端複本	<ul style="list-style-type: none">- 隨著快取增加、初始資料存取的延遲也隨之增加。
選項 2 - 鏡射	<ul style="list-style-type: none">- 來源 Volume 的完整複本- 快取補水不會增加延遲（鏡射作業完成後）	<ul style="list-style-type: none">- 必須等待鏡射作業完成後、才能存取資料- 必須管理遠端複本- 無法回寫至來源

選項 1：建立位於不同 Data Plane 的 Volume 快取

與 ["NetApp FlexCache 技術"](#)、您可以建立位於不同資料平面的 NetApp Volume 快取。例如、如果內部部署資料層中有 NetApp Volume、而您需要在 AWS 資料層面中存取該 Volume、則可以在 AWS 中建立該 Volume 的快取。本節概述為了建立位於不同資料層的 NetApp Volume 快取而必須執行的工作。

在目的地環境中建立 FlexCache Volume



如果目的地環境是內部部署的資料中心、您將在內部部署的 ONTAP 系統上建立 FlexCache Volume。如果目的地環境是 AWS、您將在 Amazon FSX for NetApp ONTAP 執行個體上建立 FlexCache Volume。

首先、您必須在目的地環境中建立 FlexCache Volume。

我們建議您使用 BlueXP 來建立 FlexCache Volume。若要使用 BlueXP 建立 FlexCache Volume、請遵循中所述的指示進行 ["BlueXP Volume 快取文件"](#)。

如果您不想使用 BlueXP、可以使用 ONTAP 系統管理員或 ONTAP CLI 來建立 FlexCache Volume。若要使用系統管理員建立 FlexCache Volume、請參閱中所述的指示 ["本文檔 ONTAP"](#)。若要使用 ONTAP CLI 建立 FlexCache Volume、請參閱中所述的指示 ["本文檔 ONTAP"](#)。

如果您想要將此程序自動化、可以使用 ["BlueXP API"](#)、["靜態API ONTAP"](#)或 ["ONTAP Ansible 系列"](#)。



Amazon FSX for NetApp ONTAP 不提供 System Manager。

將 FlexCache Volume 暴露於 Domino 中

接下來、您必須將 FlexCache 磁碟區公開至 Domino MLOps 平台。若要將 FlexCache Volume 暴露於 Domino 中、請遵循中「公開未由 Astra Trident 配置的現有 NFS Volume」子區段中所述的指示 "「將現有的 NetApp Volume 暴露於 Domino 中」一節"。

現在、您可以在目的地資料平面中啟動工作和工作區時、掛載 FlexCache Volume、如下面的螢幕擷取畫面所示。

建立 FlexCache Volume 之前

Start a Job

✓ Execution
FILE: main.py
ENV: Domino Sta...

✓ Compute Cluster
(optional)

✓ Data

Data that will be mounted

NAME	DATA TYPE	DATA PLANE	KIND
quick-start	Dataset	Local	Project
image-data	EDV	rtp-aillab-kube02 ...	Nfs

Unavailable in selected Dataplane

Change your Hardware Tier to mount currently unavailable data.

NAME	DATA TYPE	DATA PLANE	KIND
chatbot-data	EDV	rtp-aillab-kube02	Nfs

Cancel

< Back

Start

將 FlexCache Volume 暴露於 Domino 之後

Start a Job

✓

Execution

FILE: model.py

ENV: Domino Sta...

✓

Compute Cluster

(optional)

3

Data

Data that will be mounted

NAME	DATA TYPE	DATA PLANE	KIND
quick-start	Dataset	Local	Project
image-data	EDV	rtp-aillab-kube02	Nfs
chatbot-data	EDV	rtp-aillab-kube02	Nfs

Unavailable in selected Dataplane

Change your Hardware Tier to mount currently unavailable data.

NAME	DATA TYPE	DATA PLANE	KIND
No data found			

Cancel

< Back

Start

選項 2：複寫位於不同 Data Plane 的 Volume

與 "NetApp SnapMirror 資料複寫技術"、您可以建立位於不同資料平面的 NetApp Volume 複本。例如、如果內部部署資料層中有 NetApp Volume、而您需要在 AWS 資料層面中存取該 Volume、則可以在 AWS 中建立該 Volume 的複本。本節概述建立位於不同資料平面的 NetApp Volume 複本時、需要執行的工作。

建立SnapMirror關係

首先、您必須在來源 Volume 和目的地環境中的新目的地 Volume 之間建立 SnapMirror 關係。請注意、目的地磁碟區將在建立 SnapMirror 關係的過程中建立。

45

我們建議您使用 BlueXP 建立 SnapMirror 關係。若要與 BlueXP 建立 SnapMirror 關係、請遵循中所述的指示 ["BlueXP 複寫文件"](#)。

如果您不想使用 BlueXP、可以使用 ONTAP 系統管理員或 ONTAP CLI 來建立 SnapMirror 關係。若要與 System Manager 建立 SnapMirror 關係、請參閱中所述的指示 ["本文檔 ONTAP"](#)。若要與 ONTAP CLI 建立 SnapMirror 關係、請參閱中所述的指示 ["本文檔 ONTAP"](#)。

如果您想要將此程序自動化、可以使用 ["BlueXP API"](#)、["靜態API ONTAP"](#)或 ["ONTAP Ansible 系列"](#)。



Amazon FSX for NetApp ONTAP 不提供 System Manager。

打破 SnapMirror 關係

接下來、您必須中斷 SnapMirror 關係、才能啟動目的地 Volume 以進行資料存取。請等到初始複寫完成後再執行此步驟。



您可以檢查 BlueXP、ONTAP 系統管理員或 ONTAP CLI 中的鏡射狀態、判斷複寫是否完成。複寫完成後、鏡射狀態將會是「快照鏡射」。

我們建議您使用 BlueXP 來打破 SnapMirror 關係。若要中斷與 BlueXP 的 SnapMirror 關係、請遵循中所述的指示 ["BlueXP 複寫文件"](#)。

如果您不想使用 BlueXP、可以使用 ONTAP 系統管理員或 ONTAP CLI 來中斷 SnapMirror 關係。若要中斷與 System Manager 的 SnapMirror 關係、請參閱中所述的指示 ["本文檔 ONTAP"](#)。若要中斷與 ONTAP CLI 的 SnapMirror 關係、請參閱中所述的指示 ["本文檔 ONTAP"](#)。

如果您想要將此程序自動化、可以使用 ["BlueXP API"](#)、["靜態API ONTAP"](#)或 ["ONTAP Ansible 系列"](#)。

將目的地 Volume 公開至 Domino

接下來、您必須將目的地 Volume 公開至 Domino MLOps 平台。若要將目的地 Volume 暴露於 Domino 中、請遵循中「公開未由 Astra Trident 配置的現有 NFS Volume」子區段中所述的指示 ["「將現有的 NetApp Volume 暴露於 Domino 中」一節"](#)。

現在、您可以在目的地資料平面中啟動工作和工作區時、掛載目的地 Volume、如下面的螢幕擷取畫面所示。

建立 SnapMirror 關係之前

Start a Job

✓

Execution

FILE: main.py

ENV: Domino Sta...

✓

Compute Cluster

(optional)

✓

Data

Data that will be mounted

NAME	DATA TYPE	DATA PLANE	KIND
quick-start	Dataset	Local	Project
image-data	EDV	rtp-aillab-kube02 ...	Nfs

Unavailable in selected Dataplane

Change your Hardware Tier to mount currently unavailable data.

NAME	DATA TYPE	DATA PLANE	KIND
chatbot-data	EDV	rtp-aillab-kube02	Nfs

Cancel

< Back

Start

將目的地 **Volume** 暴露於 **Domino** 之後

47

Start a Job

✓

Execution

FILE: model.py

ENV: Domino Sta...

✓

Compute Cluster

(optional)

3

Data

Data that will be mounted

NAME	DATA TYPE	DATA PLANE	KIND
quick-start	Dataset	Local	Project
image-data	EDV	rtp-aillab-kube02	Nfs
chatbot-data	EDV	rtp-aillab-kube02	Nfs

Unavailable in selected Dataplane

Change your Hardware Tier to mount currently unavailable data.

NAME	DATA TYPE	DATA PLANE	KIND
No data found			

Cancel

< Back

Start

何處可找到其他資訊

若要深入瞭解本文件所述資訊、請參閱下列文件和/或網站：

- Domino Data Lab

["https://domino.ai"](https://domino.ai)

- Domino Nexus

["https://domino.ai/platform/nexus"](https://domino.ai/platform/nexus)

48

- NetApp BlueXP

["https://bluexp.netapp.com"](https://bluexp.netapp.com)

- NetApp ONTAP 資料管理軟體

["https://www.netapp.com/data-management/ontap-data-management-software/"](https://www.netapp.com/data-management/ontap-data-management-software/)

- NetApp AI 解決方案

["https://www.netapp.com/artificial-intelligence/"](https://www.netapp.com/artificial-intelligence/)

感謝

- Domino Data Lab 技術聯盟 SA 總監 Josh Mineroff
- Domino Data Lab 現場技術長 Nicholas Jablonski
- NetApp 解決方案架構設計師 Prabu Arjunan
- Brian Young 、 NetApp 技術聯盟合作夥伴全球聯盟總監

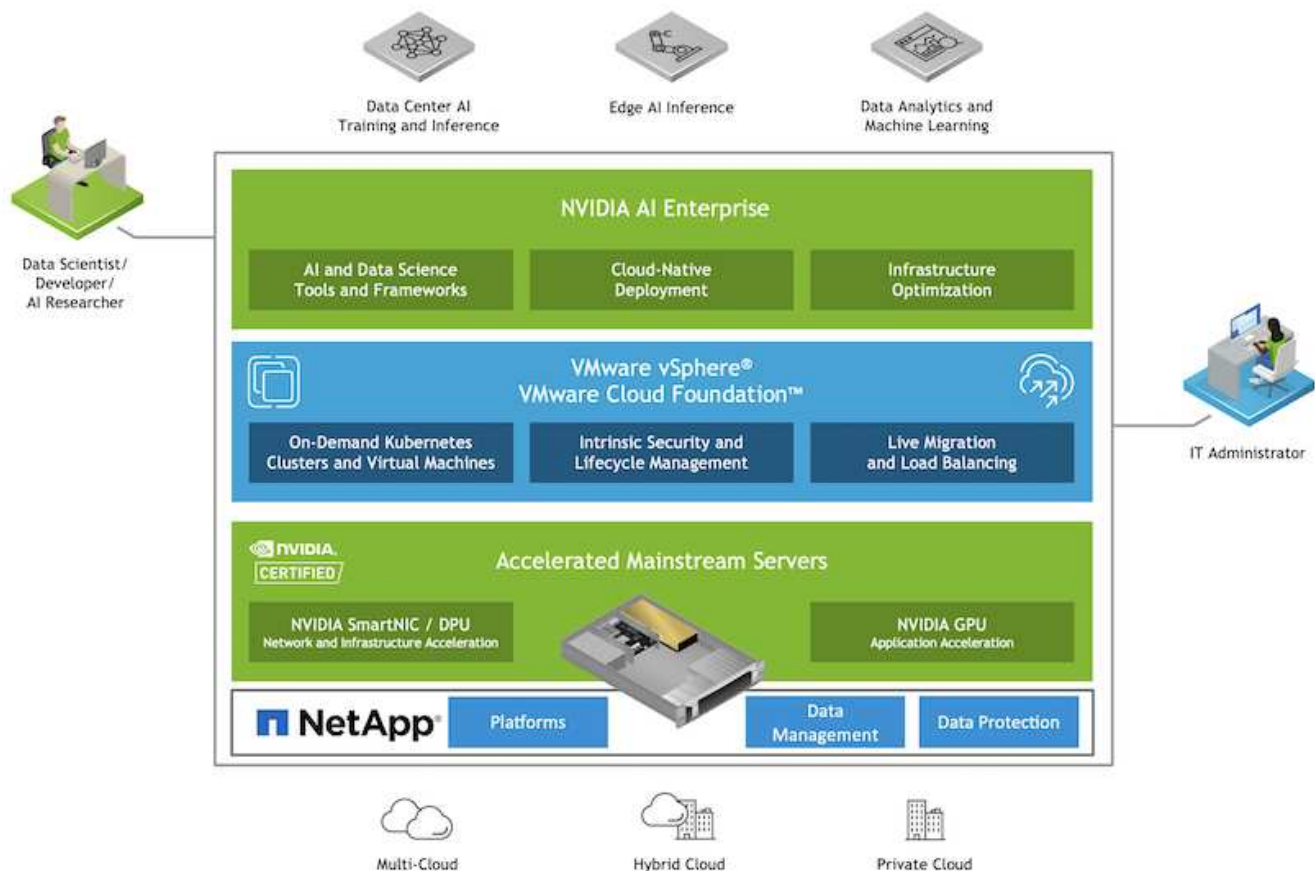
NVIDIA AI Enterprise搭配NetApp和VMware

NVIDIA AI Enterprise搭配NetApp和VMware

Mike Oglesby 、 NetApp

對於IT架構設計師和管理員而言、AI工具可能複雜且不熟悉。此外、許多AI平台還未做好企業準備。NVIDIA AI Enterprise採用NetApp和VMware技術、提供精簡的企業級AI架構。

NVIDIA AI Enterprise是一套端點對端點、雲端原生的AI與資料分析軟體套件、經過NVIDIA最佳化、認證及支援、可在採用NVIDIA認證系統的VMware vSphere上執行。此軟體可在現代化的混合雲環境中、輕鬆快速地部署、管理及擴充AI工作負載。NVIDIA AI Enterprise採用NetApp與VMware技術、以簡化且熟悉的套件提供企業級AI工作負載與資料管理功能。



技術總覽

NVIDIA AI Enterprise

NVIDIA AI Enterprise是一套端點對端點、雲端原生的AI與資料分析軟體套件、經過NVIDIA最佳化、認證及支援、可在採用NVIDIA認證系統的VMware vSphere上執行。此軟體可在現代化的混合雲環境中、輕鬆快速地部署、管理及擴充AI工作負載。

NVIDIA GPU雲端（NGC）

NVIDIA NGC主打GPU最佳化軟體目錄、讓AI從業者得以開發AI解決方案。此外、它還能存取各種AI服務、包括NVIDIA Base Command for Model訓練、NVIDIA車隊Command for Deploy and Monitor model、以及NGC Private登錄、以安全地存取及管理專屬AI軟體。此外、NVIDIA AI Enterprise客戶也可以透過NGC入口網站要求支援。

VMware vSphere

VMware vSphere是VMware的虛擬化平台、可將資料中心轉換成彙總式運算基礎架構、其中包括CPU、儲存設備和網路資源。vSphere將這些基礎架構管理為統一化的作業環境、並提供系統管理員工具來管理參與該環境的資料中心。

vSphere的兩個核心元件為ESXi和vCenter Server。ESXi是系統管理員建立及執行虛擬機器和虛擬應用裝置的虛擬化平台。vCenter Server是一項服務、可讓系統管理員管理網路和集區主機資源中連線的多個主機。

NetApp ONTAP

NetApp最新一代的儲存管理軟體、即支援企業將基礎架構現代化、並移轉至雲端就緒的資料中心。ONTAP利用領先業界的資料管理功能ONTAP、無論資料位於何處、只要使用一組工具、即可管理及保護資料。您也可以自由地將資料移至任何需要的位置：邊緣、核心或雲端。支援眾多功能、可簡化資料管理、加速及保護關鍵資料、並在混合雲架構中提供新一代基礎架構功能。ONTAP

簡化資料管理

資料管理對於企業IT營運和資料科學家而言至關重要、因此可將適當的資源用於AI應用程式和訓練AI/ML資料集。下列關於NetApp技術的其他資訊超出此驗證範圍、但可能會因您的部署而有所差異。

包含下列功能的資料管理軟體、可簡化及簡化作業、並降低您的總營運成本：ONTAP

- 即時資料精簡與擴充重複資料刪除技術。資料壓縮可減少儲存區塊內的空間浪費、重複資料刪除技術可大幅提升有效容量。這適用於本機儲存的資料、以及分層至雲端的資料。
- 最低、最大及可調適的服務品質（AQO）。精細的服務品質（QoS）控制有助於維持高共享環境中關鍵應用程式的效能等級。
- NetApp FabricPool自動將冷資料分層至公有和私有雲端儲存選項、包括Amazon Web Services（AWS）、Azure和NetApp StorageGRID等儲存解決方案。如需FabricPool更多有關資訊、請參閱 ["TR-4598：FabricPool 最佳實務做法"](#)。

加速並保護資料

提供優異的效能與資料保護、並以下列方式擴充這些功能：ONTAP

- 效能與較低的延遲。以最低的延遲提供最高的處理量。ONTAP
- 資料保護：支援所有平台的通用管理功能、可提供內建的資料保護功能。ONTAP
- NetApp Volume Encryption（NVE）。支援內建和外部金鑰管理、提供原生Volume層級的加密功能。ONTAP
- 多租戶和多因素驗證。支援以最高安全等級共享基礎架構資源。ONTAP

符合未來需求的基礎架構

下列功能可協助滿足嚴苛且不斷變化的業務需求：ONTAP

- 無縫擴充與不中斷營運。支援在不中斷營運的情況下、將容量新增至現有控制器和橫向擴充叢集。ONTAP客戶可以升級至最新技術、例如NVMe和32GB FC、而不需進行昂貴的資料移轉或中斷運作。
- 雲端連線：NetApp是最具雲端連線能力的儲存管理軟體、可在所有公有雲中選擇軟體定義儲存（AI）和雲端原生執行個體（NetApp）ONTAP ONTAP Select Cloud Volumes Service。
- 與新興應用程式整合。利用支援現有企業應用程式的相同基礎架構、為新一代平台和應用程式提供企業級資料服務、例如自動駕駛車輛、智慧城市和產業4.0。ONTAP

NetApp DataOps工具套件

NetApp DataOps Toolkit是一款以Python為基礎的工具、可簡化開發/訓練工作區和推斷伺服器管理、這些工作區都以高效能橫向擴充的NetApp儲存設備為後盾。主要功能包括：

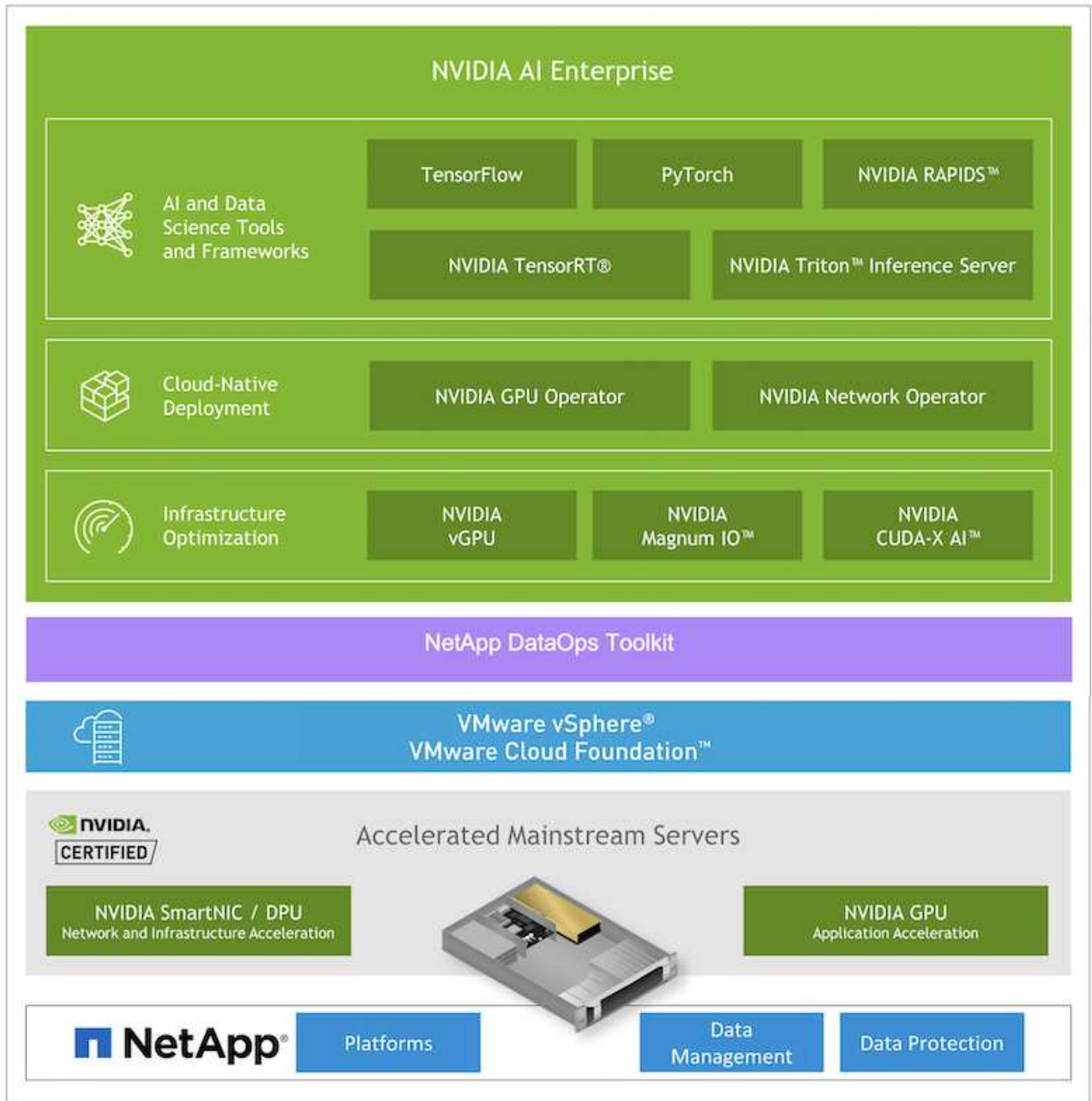
- 快速配置以高效能橫向擴充NetApp儲存設備為後盾的新高容量JupyterLab工作區。

- 快速配置以企業級NetApp儲存設備為後盾的全新NVIDIA Triton Inference Server執行個體。
- 近乎即時地複製高容量JupyterLab工作區、以進行實驗或快速迭代。
- 近乎即時地儲存高容量JupyterLab工作區的快照、以供備份和/或追蹤/基準化。
- 近乎即時地配置、複製及快照高容量、高效能的資料磁碟區。

架構

本解決方案以NetApp、VMware及NVIDIA認證系統為基礎、打造出備受肯定且熟悉的架構。如需詳細資料、請參閱下表。

元件	詳細資料
AI與資料分析軟體	"NVIDIA AI Enterprise for VMware"
虛擬化平台	"VMware vSphere"
運算平台	"NVIDIA認證系統"
資料管理平台	"NetApp ONTAP"



初始設定

本節說明在NetApp和VMware上使用NVIDIA AI Enterprise時、必須執行的初始設定工作。

先決條件

在您執行本節所述步驟之前、我們假設您已部署VMware vSphere和NetApp ONTAP VMware。請參閱 ["NVIDIA AI企業產品支援對照表"](#) 如需支援vSphere版本的詳細資訊、請參閱 ["NetApp與VMware解決方案文件"](#) 如需部署VMware vSphere搭配NetApp ONTAP 功能的詳細資訊、

安裝NVIDIA AI Enterprise Host軟體

若要安裝NVIDIA AI Enterprise主機軟體、請依照第1-4節所述的指示進行 ["NVIDIA AI企業快速入門指南"](#)。

使用NVIDIA NGC軟體

本節說明在NVIDIA AI Enterprise環境中使用NVIDIA NGC企業軟體所需執行的工作。

設定

本節說明在NVIDIA AI Enterprise環境中使用NVIDIA NGC企業軟體所需執行的初始設定工作。

先決條件

在您執行本節所述步驟之前、我們假設您已依照中所述的指示部署NVIDIA AI Enterprise主機軟體 ["初始設定"](#) 頁面。

使用vGPU建立Ubuntu Guest虛擬機器

首先、您必須使用vGPU建立Ubuntu 20.04客體VM。若要使用vGPU建立Ubuntu 20.04客體VM、請遵循中的指示大綱 ["NVIDIA AI企業部署指南"](#)。

下載並安裝NVIDIA Guest軟體

接下來、您必須在先前步驟所建立的客體VM中安裝必要的NVIDIA客體軟體。若要在客體VM內下載及安裝所需的NVIDIA客體軟體、請遵循中5.1-5.4節所述的指示 ["NVIDIA AI企業快速入門指南"](#)。



執行第5.4節所述的驗證工作時、您可能需要使用不同的CUDA Container映像版本標記、因為CUDA Container映像自撰寫指南以來就已更新。在我們的驗證中、我們使用了「nvidia/CUDA : 11.0.3-base-ubuntu20.04」。

下載AI /分析架構容器

接下來、您必須從NVIDIA NGC下載所需的AI或分析架構容器映像、以便在您的客體VM中使用。若要在客體VM內下載架構容器、請遵循中所述的指示 ["NVIDIA AI企業部署指南"](#)。

安裝及設定NetApp DataOps Toolkit

接下來、您必須在客體VM內安裝適用於傳統環境的NetApp DataOps Toolkit。NetApp DataOps Toolkit可用於直接從ONTAP 客體VM內的終端機、管理您的一套系統上的橫向擴充資料磁碟區。若要在客體VM內安裝NetApp DataOps Toolkit、請執行下列工作。

1. 安裝Pip。

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

2. 登出客體VM終端機、然後重新登入。

3. 設定NetApp DataOps Toolkit。若要完成此步驟、ONTAP 您需要針對您的整套系統提供API存取詳細資料。您可能需要向儲存管理員取得這些資訊。

```
$ 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
```

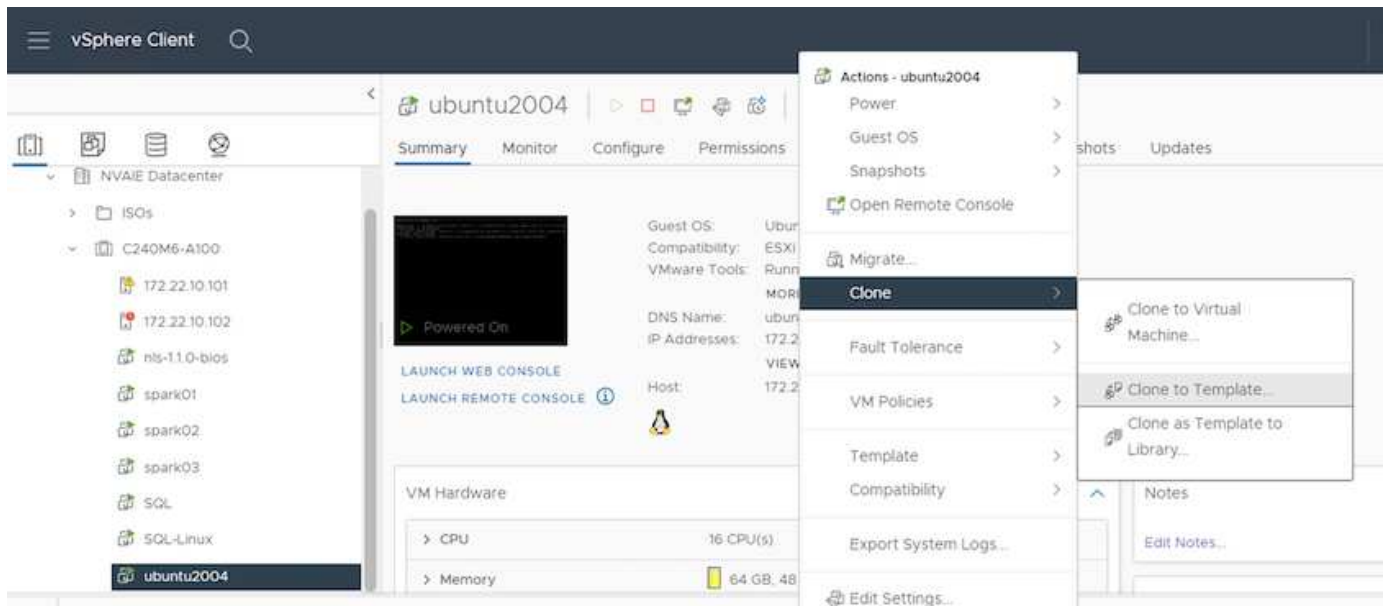
```
Do you intend to use this toolkit to push/pull from S3? (yes/no): no
```

```
Created config file: '/home/user/.netapp_dataops/config.json'.
```

建立來賓VM範本

最後、您必須根據客體VM建立VM範本。您可以使用此範本快速建立來賓VM、以使用NVIDIA NGC軟體。

若要根據客體VM建立VM範本、請登入VMware vSphere、按一下客體VM名稱、選擇「Clone（複製）」、選擇「Clone to Template（複製到範本）...」、然後依照精靈進行。



範例使用案例- TensorFlow訓練工作

本節說明在NVIDIA AI Enterprise環境中執行TensorFlow訓練工作所需執行的工作。

先決條件

在您執行本節所述步驟之前、我們假設您已依照中所述的指示建立客體VM範本 "設定" 頁面。

從範本建立來賓VM

首先、您必須從上一節建立的範本建立新的來賓VM。若要從範本建立新的來賓VM、請登入VMware vSphere、按一下範本名稱、選擇「New VM from this Template ... (從此範本新增VM ...)」、然後依照精靈進行。

vSphere Client

172.22.10.100

NVAIE Datacenter

Discovered virtual machine

vCLS

nls-1.1.0-bios

spark01

spark02

spark03

SQL

SQL-Linux

ubuntu2004

vgpu-client-ubuntu2004

Recent Tasks

Alarms

Task Name	Target
Delete virtual machine	
Clone virtual machine	

All

More Tasks

vgpu-client-ubuntu2004

SummaryMonitorConfiguration

Guest OS: Ubuntu 20.04 LTS (64-bit)
Compatibility Mode
VMware Tools: Installed and running

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

建立及掛載資料Volume

接下來、您必須建立新的資料量、以便儲存訓練資料集。您可以使用NetApp DataOps Toolkit快速建立新的資料Volume。以下命令範例顯示建立容量為2 TB的名為「imagenet」的磁碟區。

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

您必須先在客體VM內掛載資料、才能在資料磁碟區中填入資料。您可以使用NetApp DataOps Toolkit快速掛載資料磁碟區。以下命令範例顯示在上一個步驟中建立的磁碟區遠移。

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

填入資料Volume

新磁碟區完成資源配置和掛載之後、即可從來源位置擷取訓練資料集、並放在新磁碟區上。這通常需要從S3或Hadoop資料湖提取資料、有時需要資料工程師提供協助。

執行TensorFlow訓練工作

現在、您已準備好執行TensorFlow訓練工作。若要執行TensorFlow訓練工作、請執行下列工作。

1. 拉出NVIDIA NGC企業級TensorFlow容器映像。

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

2. 啟動NVIDIA NGC企業級TensorFlow容器的執行個體。使用「-v」選項將資料磁碟區附加至容器。

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

3. 在容器內執行TensorFlow訓練方案。以下命令範例顯示執行容器映像所包含的ResNet-50訓練程式範例。

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

何處可找到其他資訊

若要深入瞭解本文件所述資訊、請參閱下列文件和/或網站：

- NetApp ONTAP 數據管理軟體ONTAP —資訊庫

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

- NetApp DataOps工具套件

["https://github.com/NetApp/netapp-dataops-toolkit"](https://github.com/NetApp/netapp-dataops-toolkit)

- NVIDIA AI Enterprise搭配VMware

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

感謝

- Bobby Oommen、資深NetApp經理
- NetApp系統管理員Ramesh Issac
- NetApp技術行銷工程師Raney Daniel

TR-4851：NetApp StorageGRID 的資料湖適用於自動驅動工作負載-解決方案設計

NetApp公司David Arignette

TR-4851展示如何使用NetApp StorageGRID 的物件式儲存設備作為資料儲存庫和管理系統、以利機器學習（ML）和深度學習（DL）軟體開發。本白皮書說明自主車輛軟體開發的資料流程與需求、以及StorageGRID 簡化資料生命週期的各種功能。此解決方案適用於ML與DL開發程序中的任何一般多階段資料管線工作流程。

["TR-4851：NetApp StorageGRID 的資料湖適用於自動驅動工作負載-解決方案設計"](#)

NetApp AI Control Plane

TR-4798：NetApp AI Control Plane

Mike Oglesby、NetApp

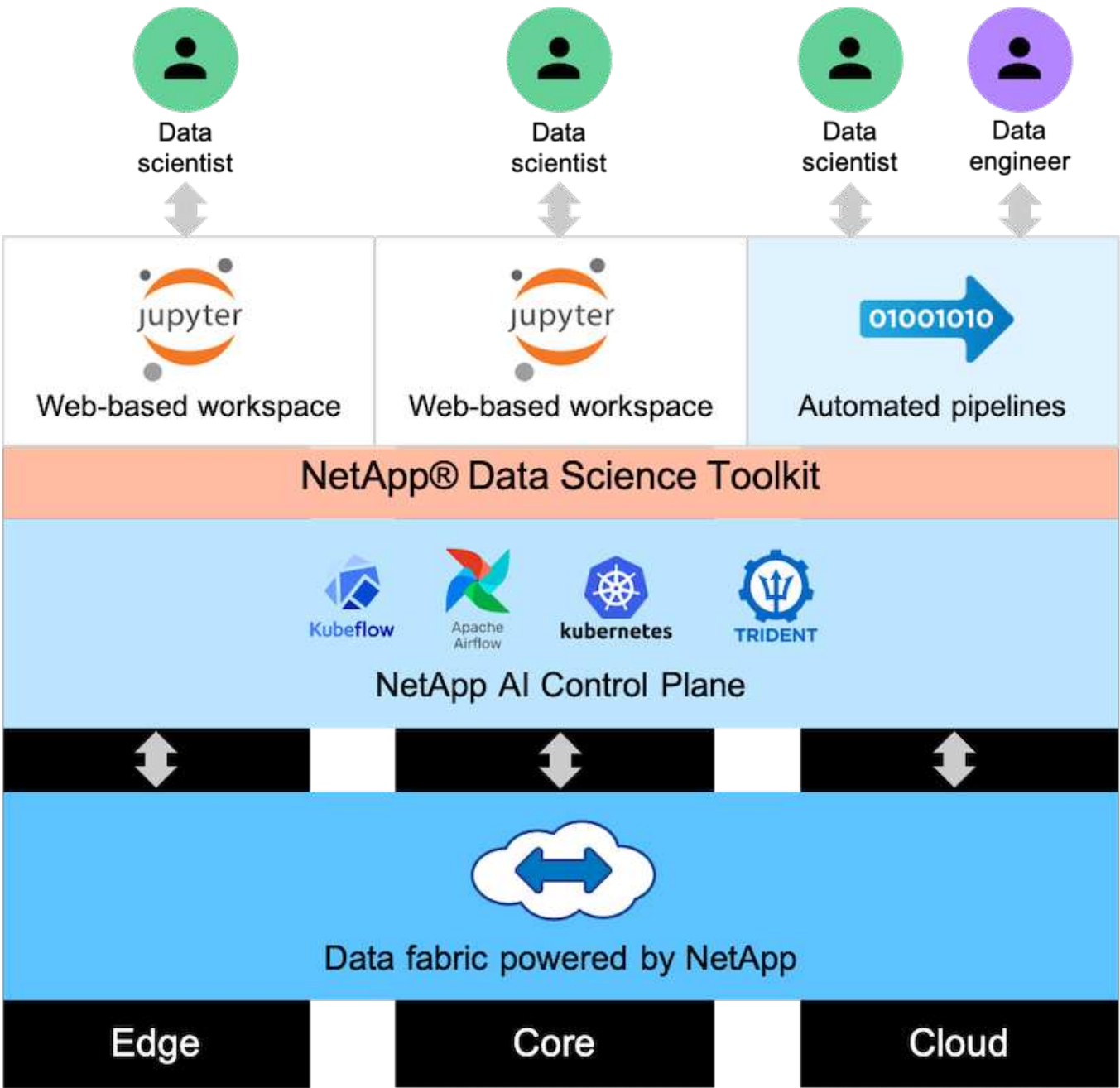
各種規模的公司和組織、以及許多產業、紛紛轉向人工智慧（AI）、機器學習（ML）和深度學習（DL）、以解決實際問題、提供創新產品和服務、並在競爭日益激烈的市場中獲得優勢。隨著企業組織增加AI、ML和DL的使用率、他們面臨許多挑戰、包括工作負載擴充性和資料可用度。本文件說明如何使用NetApp AI Control Plane解決這些挑戰、這套解決方案將NetApp資料管理功能與熱門的開放原始碼工具和架構配對。

本報告說明如何快速複製資料命名空間。同時也說明如何在站台和區域之間無縫複寫資料、以建立一致且統一的AI/ML/DL資料傳輸途徑。此外、它還會引導您完成AI、ML和DL訓練工作流程的定義與實作、這些工作流程整合了近乎即時的資料建立與模型基準、以利追蹤及版本管理。有了這套解決方案、您可以追蹤每個模型訓練回溯到用來訓練和/或驗證模型的確切資料集。最後、本文件說明如何快速配置Jupyter Notebook工作區、並存取大量資料集。

附註：針對需要共用存取相同資料集的大量GPU伺服器大規模的HPC形式分散式訓練、或是如果您需要/偏好平行檔案系統、請查看 ["TR-4890"](#)。本技術報告說明如何納入 ["NetApp完全支援的平行檔案系統解決方案BeeGFS"](#) 作為NetApp AI Control Plane的一部分。此解決方案可從少數NVIDIA DGX A100系統擴充至完整的140節

點SupermPOD。

NetApp AI Control Plane的目標對象是資料科學家和資料工程師、ONTAP 因此需要最少的NetApp或NetApp®專業知識。有了這套解決方案、您就能使用簡單且熟悉的工具和介面來執行資料管理功能。如果您的環境中已經有NetApp儲存設備、您可以立即試用NetApp AI控制面板。如果您想試用解決方案、但還沒有NetApp儲存設備、請造訪 "cloud.netapp.com"而且您可以在幾分鐘內使用雲端型NetApp儲存解決方案來啟動和執行。下圖提供解決方案的視覺化功能。



概念與元件

人工智慧

AI是一項電腦科學訓練、訓練電腦模擬人類思維的認知功能。AI開發人員訓練電腦、以類似甚至優於人類的方式學習及解決問題。深度學習和機器學習是AI的子領域。企業組織越來越採用AI、ML和DL來支援其關鍵業務需求。以下是一些範例：

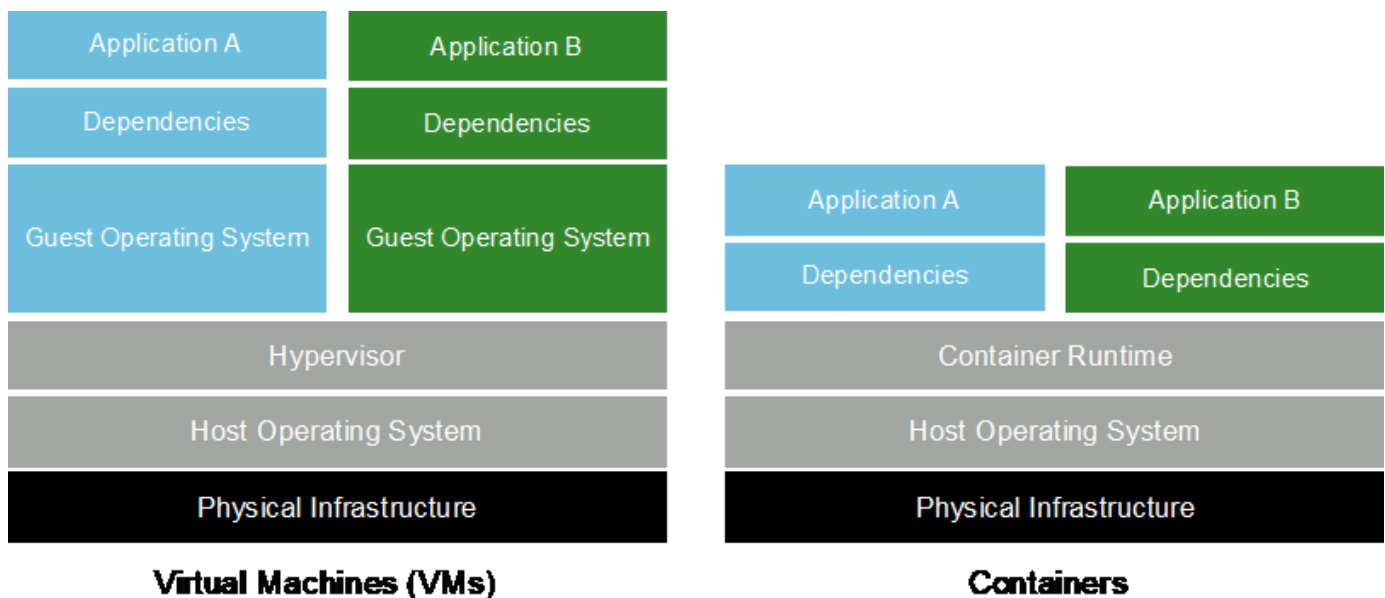
- 分析大量資料、發掘先前未知的商業洞見
- 使用自然語言處理功能直接與客戶互動
- 自動化各種業務流程與功能

現代化的AI訓練和推斷工作負載需要大量平行運算功能。因此、GPU越來越常用於執行AI作業、因為GPU的平行處理能力遠優於通用CPU。

容器

容器是獨立的使用者空間執行個體、可在共享主機作業系統核心上執行。容器的採用率正在迅速增加。Container提供許多與虛擬機器（VM）相同的應用程式沙箱效益。不過、由於虛擬機器所仰賴的Hypervisor和客體作業系統層已經被淘汰、因此容器的重量遠較輕。下圖說明虛擬機器與容器的視覺化。

容器也能直接透過應用程式、有效封裝應用程式相依性、執行時間等項目。最常用的容器包裝格式是Docker容器。以Docker Container格式容器化的應用程式、可在任何能夠執行Docker Container的機器上執行。即使應用程式的相依性並不存在於機器上、也一樣、因為所有相依性都封裝在容器本身。如需詳細資訊、請參閱 "[Docker網站](#)"。



Kubernetes

Kubernetes是開放原始碼的分散式容器協調平台、最初由Google設計、現在由Cloud Native Computing Foundation（CNCF）維護。Kubernetes可將容器化應用程式的部署、管理及擴充功能自動化。近年來、Kubernetes已成為主要的容器協調平台。雖然支援其他容器封裝格式和執行時間、但Kubernetes通常是Docker容器的協調系統。如需詳細資訊、請參閱 "[Kubernetes網站](#)"。

NetApp Trident

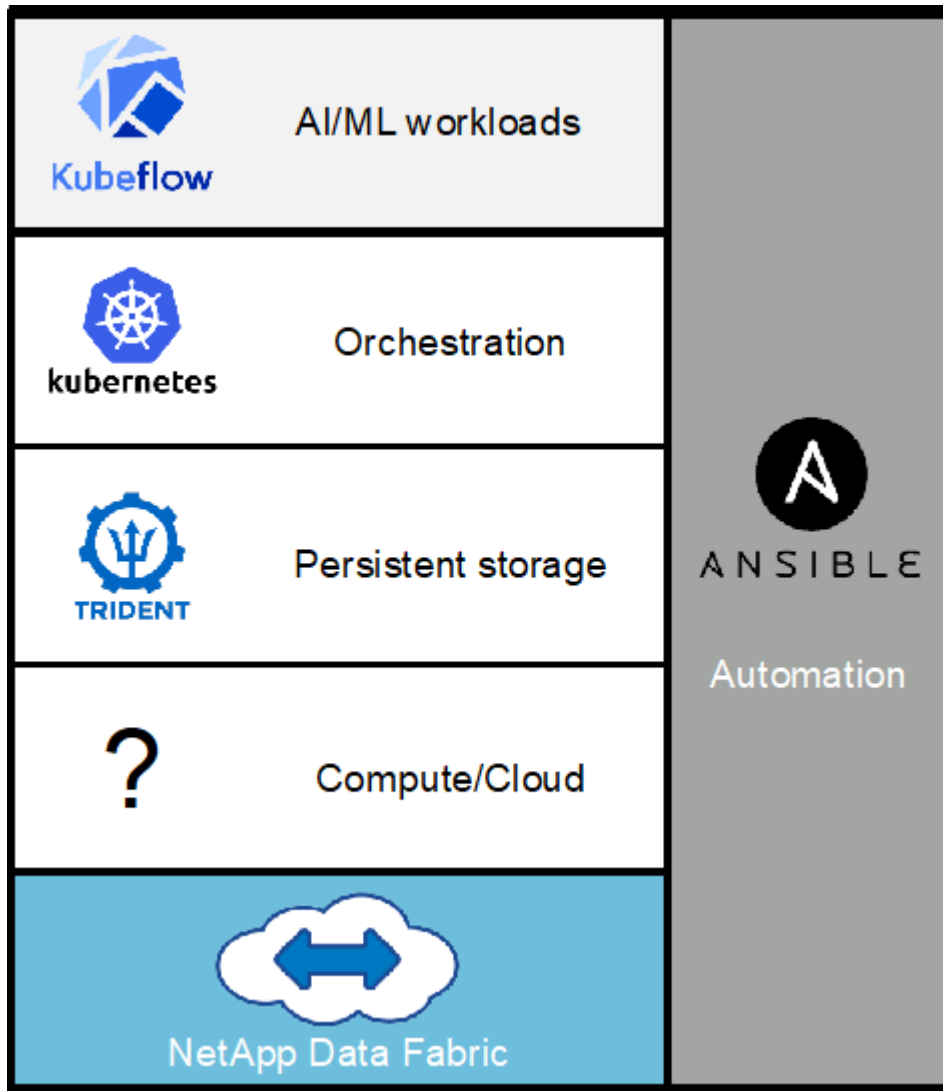
Trident是NetApp開發與維護的開放原始碼儲存協調工具、可大幅簡化Kubernetes工作負載的持續儲存設備建立、管理與使用。Trident本身是Kubernetes原生應用程式、直接在Kubernetes叢集內執行。Kubernetes使用者（開發人員、資料科學家、Kubernetes系統管理員等）可以使用他們已經熟悉的標準Kubernetes格式、建立、管理及與持續儲存磁碟區互動。同時、他們也能善用NetApp先進的資料管理功能、以及採用NetApp技術的資料架構。Trident將持續儲存設備的複雜度抽象化、使其易於使用。如需詳細資訊、請參閱 "[Trident網站](#)"。

NVIDIA DeepOps

DeepOps是NVIDIA的開放原始碼專案、使用Ansible可根據最佳實務做法、自動部署GPU伺服器叢集。DeepOps是模組化的、可用於各種部署工作。本文件及其所說明的驗證作業中、DeepOps用於部署Kubernetes叢集、其中包含GPU伺服器工作節點。如需詳細資訊、請參閱 "[DeepOps網站](#)"。

Kubeflow

Kubeflow是Kubernetes的開放原始碼AI和ML工具套件、最初由Google開發。Kubeflow專案讓Kubernetes上的AI和ML工作流程部署變得簡單、可攜且可擴充。Kubeflow將Kubernetes複雜的資料擷取出來、讓資料科學家能夠專注於他們最瞭解的資料科學。如需視覺化功能、請參閱下圖。由於企業IT部門越來越標準化Kubernetes、Kubeflow的發展也越來越顯著。如需詳細資訊、請參閱 "[Kubeflow網站](#)"。



Kubeflow Pipelines

Kubeflow Pipelines是Kubeflow的重要元件。Kubeflow Pipelines是定義及部署可攜式且可擴充的AI和ML工作流程的平台和標準。如需詳細資訊、請參閱 "[官方Kubeflow文件](#)"。

Jupyter筆記型電腦伺服器

Jupyter Notebook Server是開放原始碼的網路應用程式、可讓資料科學家建立名為Jupyter Notebooks的維基類

文件、其中包含即時程式碼和描述性測試。Jupyter筆記型電腦在AI和ML社群中廣為使用、可用來記錄、儲存及分享AI和ML專案。Kubeflow簡化了Kubernetes上Jupyter筆記型電腦伺服器的資源配置與部署。如需Jupyter筆記型電腦的詳細資訊、請參閱 "[Jupyter網站](#)"。如需Kubeflow內容中Jupyter Notebooks的詳細資訊、請參閱 "[官方Kubeflow文件](#)"。

Apache Airflow

Apache Airflow是開放原始碼的工作流程管理平台、可針對複雜的企業工作流程、進程式化的撰寫、排程及監控。它通常用於自動化ETL和資料管線工作流程、但不限於這些類型的工作流程。氣流專案是由Airbnb發起、但後來在業界廣受歡迎、現在由Apache Software Foundation贊助。氣流是以Python撰寫、氣流工作流程是透過Python指令碼建立、氣流是以「組態為程式碼」的原則設計。許多企業氣流使用者現在都在Kubernetes上執行氣流。

定向Acyclic圖表（DAG）

在氣流中、工作流程稱為「導向Acyclic Graphs (DAG)（導向型Acyclic Graphs (DAG)）」。DAG是由依順序、平行或兩者組合執行的工作所組成、視DAG定義而定。氣流排程器會在一組工作人員上執行個別工作、並遵循DAG定義中指定的工作層級相依性。DAG是透過Python指令碼來定義和建立。

NetApp ONTAP 產品9.

NetApp ONTAP 支援NetApp的最新一代儲存管理軟體、可讓像您這樣的企業將基礎架構現代化、並移轉至雲端就緒的資料中心。藉由領先業界的資料管理功能、ONTAP 無論資料位於何處、您都能使用單一工具組來管理及保護資料。您也可以自由地將資料移至任何需要的位置：邊緣、核心或雲端。包含許多功能、可簡化資料管理、加速並保護關鍵資料、以及跨越混合雲架構的符合未來需求的基礎架構。ONTAP

簡化資料管理

資料管理對企業IT營運至關重要、因此您可以將適當的資源用於應用程式和資料集。包含下列功能、可簡化及簡化作業、並降低營運總成本：ONTAP

- *即時資料壓縮與擴充重複資料刪除技術。*資料壓縮技術可減少儲存區塊內的空間浪費、重複資料刪除技術則可大幅提升有效容量。
- *服務品質（QoS）的最低、最大和調適性。*精細的QoS控制可協助維持高共享環境中關鍵應用程式的效能等級。
- *此功能可將冷資料自動分層至公有雲和私有雲儲存選項、包括Amazon Web Services（AWS）、Azure和NetApp以物件為基礎的儲存設備。ONTAP FabricPool StorageGRID

加速並保護資料

提供優異的效能與資料保護、並透過下列功能來延伸這些功能：ONTAP

- *高效能與低延遲。ONTAP *
- * NetApp ONTAP FlexGroup 功能* FlexGroup 。*一個功能強大的資料容器、可線性擴充至高達20PB和4000億個檔案、提供簡化資料管理的單一命名空間。
- 資料保護 ONTAP 效能*功能提供內建的資料保護功能、並可在所有平台上進行通用管理。
- * NetApp Volume Encryption。ONTAP *支援內建和外部金鑰管理、提供原生Volume層級的加密功能。

支援需求嚴苛且瞬息萬變的企業需求：ONTAP

- 無縫擴充與不中斷營運。ONTAP 支援在不中斷營運的情況下、為現有控制器和橫向擴充叢集增加容量。您可以升級至最新技術、例如NVMe和32GB FC、而不需進行昂貴的資料移轉或中斷運作。
- * Cloud connection。* ONTAP 效能不只是雲端連線能力最強的儲存管理軟體之一、ONTAP Select 還可在Cloud Volumes Service 所有公有雲中選擇軟體定義儲存（英文）和雲端原生執行個體（NetApp邊）。
- 整合新興應用程式。ONTAP 使用支援現有企業應用程式的相同基礎架構、即可為OpenStack、Hadoop和MongoDB等新一代平台和應用程式提供企業級資料服務。

NetApp Snapshot複本

NetApp Snapshot複本是磁碟區的唯讀時間點映像。此映像會佔用最小的儲存空間、並產生可忽略的效能負荷、因為它只會記錄自上次建立Snapshot複本以來所建立的檔案變更、如下圖所示。

Snapshot複本的效率歸功於核心ONTAP 的不穩定儲存虛擬化技術WAFL、亦即Write Anywhere File Layout（簡稱「Write Anywhere File Layout」、簡稱「Write Anywhere」）。如同資料庫、WAFL 利用中繼資料指向磁碟上的實際資料區塊。但是WAFL、不像資料庫、不像是使用什麼功能來覆寫現有的區塊。它會將更新的資料寫入新的區塊、並變更中繼資料。這是因為ONTAP 當我們建立Snapshot複本時、不需要複製資料區塊、而是參考中繼資料、所以Snapshot複本非常有效率。如此可免除其他系統在尋找要複製的區塊時所需的搜尋時間、以及複本本身的成本。

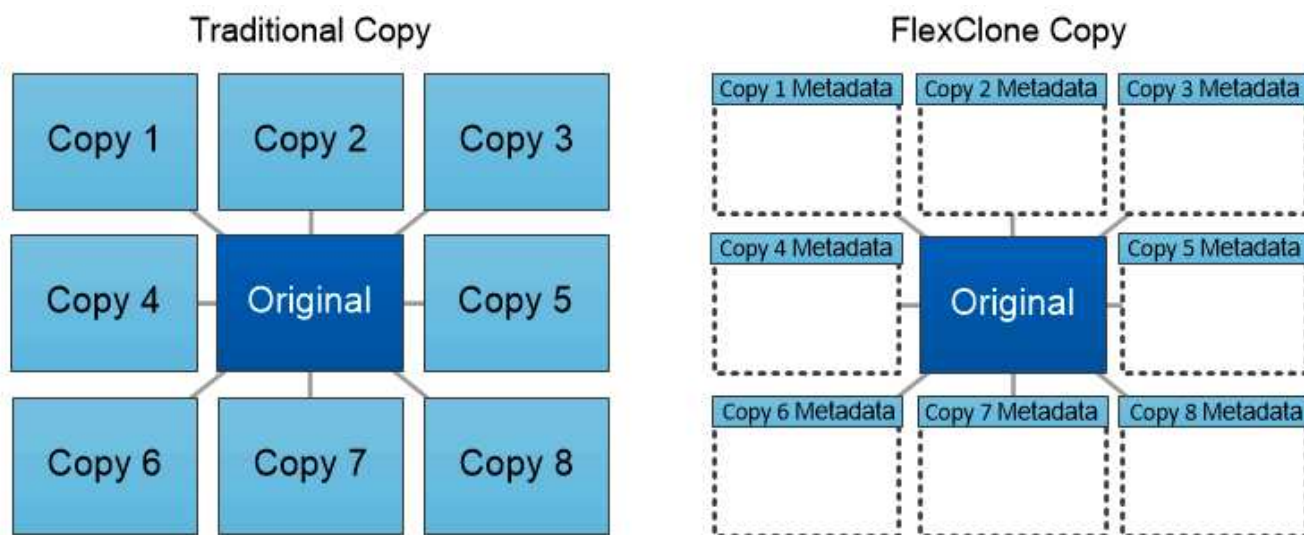
您可以使用Snapshot複本來還原個別檔案或LUN、或還原磁碟區的完整內容。此功能可將Snapshot複本中的指標資訊與磁碟上的資料進行比較、以重建遺失或損壞的物件、而不會造成停機或重大效能成本。ONTAP



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

NetApp FlexClone技術

NetApp FlexClone技術會參考Snapshot中繼資料、以建立磁碟區的可寫入時間點複本。複本會與父實體共用資料區塊、除非中繼資料需要的資料、否則不會佔用任何儲存空間、直到將變更寫入複本為止、如下圖所示。在傳統複本需要數分鐘甚至數小時才能建立的地方、FlexClone軟體可讓您幾乎即時複製最大的資料集。這使得它非常適合您需要多個相同資料集複本（例如開發工作區）或資料集暫存複本（針對正式作業資料集測試應用程式）的情況。

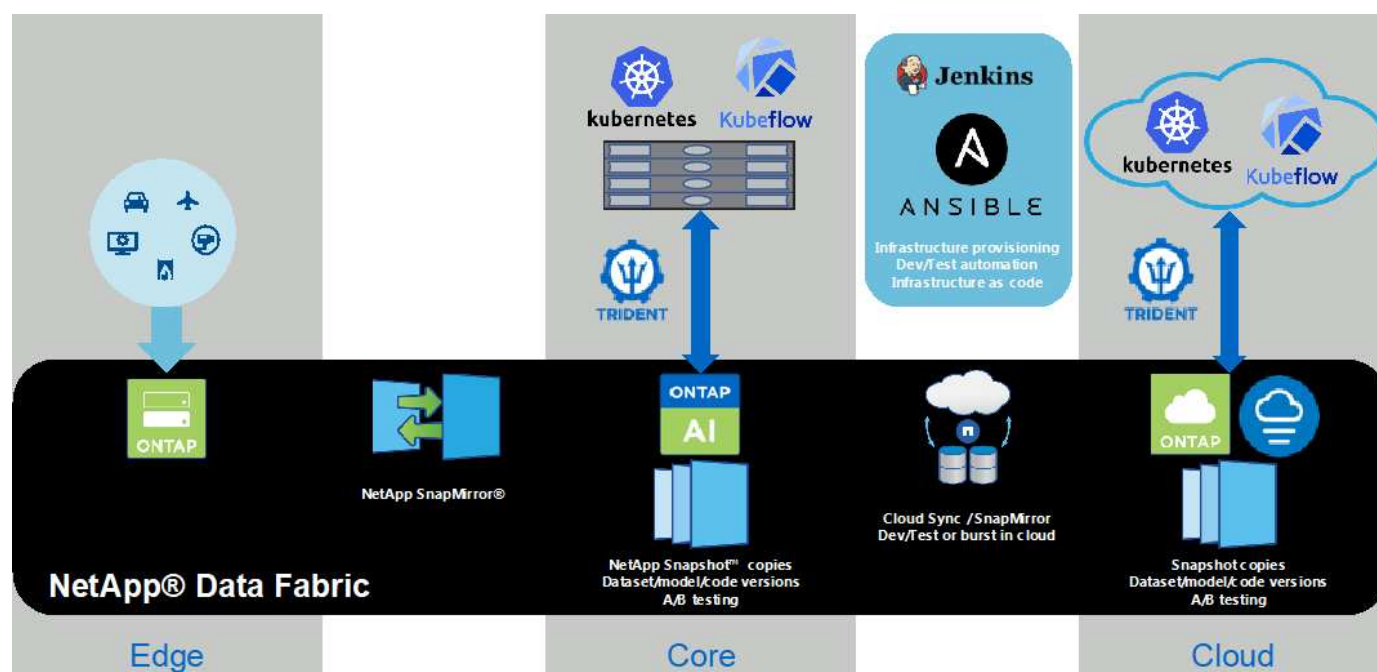


FlexClone copies share data blocks with their parents, consuming no storage except what is required for metadata.

NetApp SnapMirror資料複寫技術

NetApp SnapMirror軟體是一款具成本效益且易於使用的統一化複寫解決方案、適用於整個資料架構。它可透過LAN或WAN高速複寫資料。它可為各種應用程式提供高資料可用度及快速資料複寫、包括虛擬與傳統環境中的業務關鍵應用程式。當您將資料複寫到一或多個NetApp儲存系統、並持續更新次要資料時、資料會保持最新狀態、而且隨時可供使用。不需要外部複寫伺服器。請參閱下圖、瞭解運用SnapMirror技術的架構範例。

SnapMirror軟體透過ONTAP 網路僅傳送變更的區塊、充分發揮NetApp的效能。SnapMirror軟體也使用內建的網路壓縮功能來加速資料傳輸、並減少高達70%的網路頻寬使用率。有了SnapMirror技術、您可以利用單一精簡複寫資料串流來建立單一儲存庫、同時維護作用中鏡像和先前的時間點複本、最多可減少50%的網路流量。



NetApp BlueXP 複製與同步

BlueXP 複製與同步是 NetApp 服務、可快速安全地同步資料。無論您需要在內部部署的 NFS 或 SMB 檔案共用、NetApp StorageGRID、NetApp ONTAP S3、NetApp Cloud Volumes Service、Azure NetApp Files、AWS S3、AWS EFS、Azure Blob、Google Cloud Storage 或 IBM Cloud Object Storage、BlueXP 複製與同步功能可快速安全地將檔案移至所需的位置。

資料傳輸完成後、即可在來源和目標上完全使用。BlueXP 複製與同步可在觸發更新時隨需同步資料、或根據預先定義的排程持續同步資料。不過、BlueXP 複製與同步只會移動資料量、因此將用於資料複寫的時間與金錢降到最低。

BlueXP 複製與同步是一種軟體即服務（SaaS）工具、設定與使用極為簡單。BlueXP 複製與同步所觸發的資料傳輸是由資料代理人執行。BlueXP 複製與同步資料代理人可以部署在 AWS、Azure、Google Cloud Platform 或內部部署。

NetApp XCP

NetApp XCP是以用戶端為基礎的軟體、適用於任何對NetApp和NetApp對NetApp的資料移轉及檔案系統洞見。XCP的設計旨在利用所有可用的系統資源來處理大量資料集和高效能移轉、以擴充並達到最大效能。XCP可讓您利用產生報告的選項、全面掌握檔案系統。

NetApp XCP可在單一套件中取得、支援NFS和SMB傳輸協定。XCP包含適用於NFS資料集的Linux二進位檔、以及適用於SMB資料集的Windows執行檔。

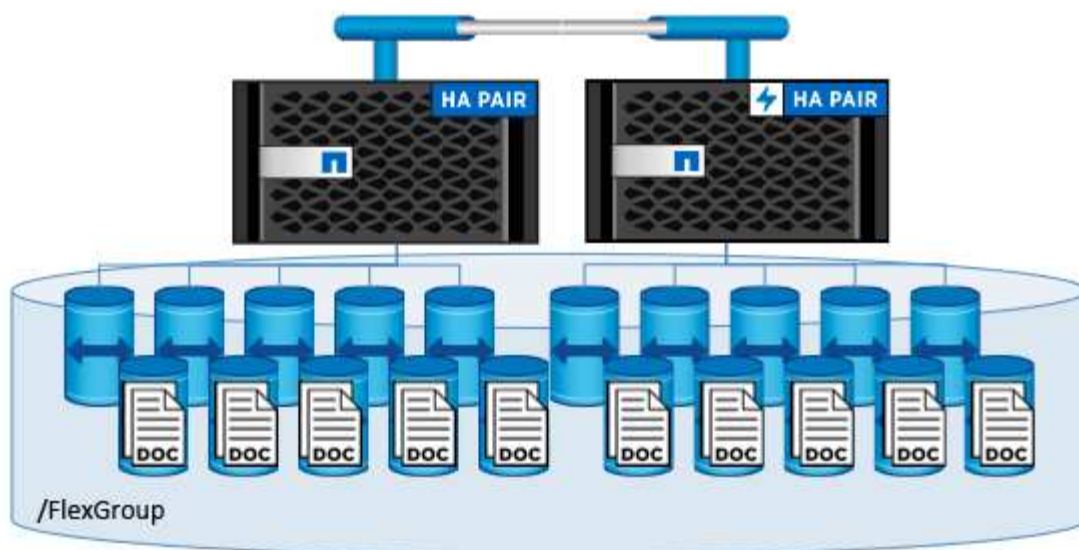
NetApp XCP檔案分析是以主機為基礎的軟體、可偵測檔案共用、在檔案系統上執行掃描、並提供檔案分析儀表板。XCP檔案分析可與NetApp和非NetApp系統相容、並可在Linux或Windows主機上執行、以提供NFS和SMB匯出檔案系統的分析功能。

NetApp ONTAP FlexGroup 產品區

訓練資料集可能是數十億個檔案的集合。檔案可以包含文字、音訊、視訊及其他形式的非結構化資料、這些資料必須儲存和處理才能並行讀取。儲存系統必須儲存大量的小型檔案、而且必須平行讀取這些檔案、才能執行連續和隨機I/O

例如下圖所示、一個包含多個組成成員磁碟區的單一命名空間。FlexGroup從儲存管理員的觀點來看、FlexGroup 可管理一個不實的功能、就像NetApp FlexVol 的一套功能。將某個資料區中的檔案FlexGroup 分配給個別成員磁碟區、而不會跨磁碟區或節點進行等量分佈。這些功能可實現下列功能：

- 支援多PB容量、可預測低延遲的高中繼資料工作負載。FlexGroup
- 在同一個命名空間中支援高達4000億個檔案。
- 它們支援跨CPU、節點、集合體及組成FlexVol 的等量資料磁碟區、在NAS工作負載中進行平行化作業。



硬體與軟體需求

NetApp AI Control Plane解決方案不依賴於此特定硬體。此解決方案可與Trident支援的任何NetApp實體儲存設備、軟體定義執行個體或雲端服務相容。範例包括NetApp AFF 的不完整儲存系統Azure NetApp Files、功能完善、NetApp Cloud Volumes Service 功能完善、NetApp ONTAP Select 功能完善、由軟體定義的儲存執行個體、或是NetApp Cloud Volumes ONTAP 的不完整執行個體。此外、只要Kubernetes和NetApp Trident支援所使用的Kubernetes版本、即可在任何Kubernetes叢集上實作解決方案。如需Kubernetes支援版本的清單、請參閱 "[官方Kubeflow文件](#)"。如需Trident支援的Kubernetes版本清單、請參閱 "[Trident文件](#)"。如需驗證解決方案所用環境的詳細資訊、請參閱下表。

基礎架構元件	數量	詳細資料	作業系統
部署跳接主機	1.	VM	Ubuntu 20.04.2 LTS
Kubernetes主節點	1.	VM	Ubuntu 20.04.2 LTS
Kubernetes工作節點	2.	VM	Ubuntu 20.04.2 LTS
Kubernetes GPU工作節點	2.	NVIDIA DGX-1 (裸機)	NVIDIA DGX OS 4.0.5 (以Ubuntu 18.04.2 LTS為基礎)
儲存設備	1個HA配對	NetApp AFF 解決方案-A220	NetApp ONTAP 產品9.7 P6

軟體元件	版本
Apache Airflow	2.0.1
Apache Airflow Helm圖表	8.0.8
Docker	19.03.12
Kubeflow	1.2

軟體元件	版本
Kubernetes	1.18.9.
NetApp Trident	21.01.2
NVIDIA DeepOps	主分支機構在提交時的Trident部署功能 " 61898cdfda " ；所有其他功能（版本21.03）

支援

NetApp不提供Apache Airflow、Docker、Kubeflow、Kubernetes或NVIDIA DeepOps的企業支援。如果您對具備類似NetApp AI Control Plane解決方案功能的完整支援解決方案感興趣、"[請聯絡NetApp](#)" 關於NetApp與合作夥伴共同提供的完全支援的AI/ML解決方案。

Kubernetes部署

本節說明部署Kubernetes叢集所需完成的工作、以實作NetApp AI Control Plane解決方案。如果您已經有Kubernetes叢集、則只要您執行Kubernetes版本、且Kubeflow和NetApp Trident均支援此版本、就可以跳過本節。如需Kubernetes支援版本的清單、請參閱 "[官方Kubeflow文件](#)"。如需Trident支援的Kubernetes版本清單、請參閱 "[Trident文件](#)"。

對於採用裸機節點的內部部署Kubernetes、採用NVIDIA GPU、NetApp建議使用NVIDIA的DeepOps Kubernetes部署工具。本節概述如何使用DeepOps部署Kubernetes叢集。

先決條件

在您執行本節所述的部署練習之前、我們假設您已經執行下列工作：

1. 您已根據ONTAP 標準組態指示、設定任何裸機Kubernetes節點（例如、NVIDIA DGX系統是整個AI Pod的一部分）。
2. 您已在所有Kubernetes主節點和工作節點、以及部署跨接主機上安裝支援的作業系統。如需DeepOps支援的作業系統清單、請參閱 "[DeepOps GitHub網站](#)"。

使用NVIDIA DeepOps來安裝及設定Kubernetes

若要使用NVIDIA DeepOps部署及設定Kubernetes叢集、請從部署跳接主機執行下列工作：

1. 依照上的指示下載NVIDIA DeepOps "[入門頁面](#)" 在NVIDIA DeepOps GitHub網站上。
2. 依照上的指示、在叢集中部署Kubernetes "[Kubernetes部署指南頁面](#)" 在NVIDIA DeepOps GitHub網站上。

NetApp Trident部署與組態

NetApp Trident部署與組態

本節說明在Kubernetes叢集中安裝及設定NetApp Trident所需完成的工作。

先決條件

在您執行本節所述的部署練習之前、我們假設您已經執行下列工作：

1. 您已經有一個正常運作的Kubernetes叢集、而且您正在執行Trident支援的Kubernetes版本。如需支援版本的清單、請參閱 ["Trident文件"](#)。
2. 您已經擁有Trident支援的有效NetApp儲存設備、軟體定義執行個體或雲端儲存服務。

安裝Trident

若要在Kubernetes叢集中安裝及設定NetApp Trident、請從部署跨接主機執行下列工作：

1. 使用下列其中一種方法部署Trident：
 - 如果您使用NVIDIA DeepOps來部署Kubernetes叢集、也可以使用NVIDIA DeepOps在Kubernetes叢集中部署Trident。若要部署Trident with DeepOps、請遵循 ["Trident部署指示"](#) 在NVIDIA DeepOps GitHub網站上。
 - 如果您沒有使用NVIDIA DeepOps來部署Kubernetes叢集、或是想要手動部署Trident、您可以遵循來部署Trident ["部署指示"](#) 在Trident文件中。請務必至少建立一個Trident Backend和至少一個Kubernetes StorageClass、以取得如何設定的詳細資訊 ["後端"](#) 和 ["儲存類別"](#) 請參閱NetApp文件中的連結小節。



如果您要在ONTAP 一個AI Pod上部署NetApp AI Control Plane解決方案、請參閱 ["Trident後端範例、適用於ONTAP AI部署"](#) 如需您可能想要建立和的不同Trident後端的一些範例 ["Kubernetes Storageclasses範例、可用於ONTAP 進行AI部署"](#) 如需您可能想要建立的不同Kubernetes StorageClass範例。

Trident後端範例、適用於ONTAP AI部署

您必須先建立一或多個Trident後端、才能使用Trident在Kubernetes叢集中動態配置儲存資源。以下範例代表在ONTAP 將NetApp AI Control Plane解決方案部署到AI Pod上時、您可能會想要建立的不同類型後端。如需後端的詳細資訊、請參閱 ["Trident文件"](#)。

1. NetApp建議針對您想要在NetApp AFF 供應系統上使用的每個資料LIF（提供資料存取的邏輯網路介面）、建立具有FlexGroup功能的Trident後端。這可讓您在所有生命期之間平衡Volume掛載

以下命令範例顯示為兩個不同的資料LIF建立兩個啟用FlexGroup的Trident後端、這些資料LIF與相同ONTAP的物件儲存虛擬機器（SVM）相關聯。這些後端使用「ontap-nas flexgroup」儲存驅動程式。支援兩種主要資料Volume類型：功能完善和功能完善。ONTAP FlexVol FlexGroup由於資料不多（本文所述的最大大小取決於特定部署）、因此不受支援。FlexVol另一方面、由於支援的資料量可線性擴充至20PB和4000億個檔案、因此單一命名空間可大幅簡化資料管理。FlexGroup因此FlexGroup、對於仰賴大量資料的AI和ML工作負載而言、此功能是最佳選擇。

如果您使用的是少量資料、想要使用FlexVol 不FlexGroup 含「orfvolume」的「orfvolume」、您可以建立使用「ontap-nas」儲存驅動程式而非「ontap-nas flexgroup」儲存驅動程式的「Trident後端」。

```
$ 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 |
+-----+-----+
|          | 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 |
+-----+-----+
|          | STATE | VOLUMES |
+-----+-----+
| ontap-ai-flexgroups-iface2 | ontap-nas-flexgroup | 61814d48-c770-436b-
9cb4-cf7ee661274d | online |      0 |
+-----+-----+
+-----+-----+
$ tridentctl get backend -n trident
+-----+
+-----+-----+

```

UUID	NAME	STATE	VOLUMES	STORAGE DRIVER
+-----+-----+-----+-----+				
b74cbddb-e0b8-40b7-b263-b6da6dec0bdd	ontap-ai-flexgroups-iface1	online	0	ontap-nas-flexgroup
61814d48-c770-436b-9cb4-cf7ee661274d	ontap-ai-flexgroups-iface2	online	0	ontap-nas-flexgroup
+-----+-----+-----+-----+				

2. NetApp也建議您建立一個或多個FlexVol 啟用了功能不全的Trident後端。如果您使用FlexGroup 支援資料集儲存設備的功能來進行測試、您可能會想要使用FlexVol 支援資料集的功能來儲存結果、輸出、偵錯資訊等。如果您想要使用FlexVol 「資料不全」、您必須建立一個或多個FlexVol 啟用「功能不全」的「資料不全」後端。以下的命令範例顯示如何建立FlexVol 使用單一資料LIF的單一啟用了功能不全的Trident後端。

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

Kubernetes StorageClass範例ONTAP、可用於進行AI部署

您必須先建立一或多個Kubernetes StorageClass、才能使用Trident在Kubernetes叢集中動態配置儲存資源。以下範例代表在ONTAP 將NetApp AI Control Plane解決方案部署到AI Pod上時、您可能想要建立的不同類型StorageClass。如需StorageClass的詳細資訊、

請參閱 ["Trident文件"](#)。

1. NetApp建議為您在本節中建立的每個啟用FlexGroup的Trident後端、建立個別的StorageClass ["Trident後端範例、適用於ONTAP AI部署"](#)步驟1。這些精細的StorageClass可讓您新增對應於特定LIF（建立Trident後端時所指定的LIF）的NFS掛載、做為StorageClass規格檔案中指定的特定後端。以下命令範例顯示建立兩個StorageClass的方式、這兩個類別對應於本節中建立的兩個範例後端 ["Trident後端範例、適用於ONTAP AI部署"](#)步驟1。如需StorageClass的詳細資訊、請參閱 ["Trident文件"](#)。

因此當刪除對應的PersistentVolume Claim (PVC) 時、不會刪除持續磁碟區、以下範例使用「回收原則」值「保留」。如需「回收政策」欄位的詳細資訊、請洽相關官員 ["Kubernetes文件"](#)。

```
$ 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也建議您建立與您在本節中建立的具有FlexVol功能的Trident後端相對應的StorageClass ["Trident後端範例、適用於ONTAP AI部署"](#)步驟2：以下命令範例顯示建立FlexVol 單一StorageClass for the餐廳。

在下列範例中、由於只建立一個具有FlexVol功能的Trident後端、因此不會在StorageClass定義檔案中指定特定的後端。當您使用Kubernetes來管理使用此StorageClass的磁碟區時、Trident會嘗試使用任何使用「ONTAP-NAS」驅動程式的可用後端。

```
$ 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也建議建立FlexGroup 通用StorageClass以供參考Volume使用。下列命令範例顯示建立FlexGroup 單一通用StorageClass for the餐廳。

請注意、StorageClass定義檔案中未指定特定的後端。因此、當您使用Kubernetes來管理使用此StorageClass的磁碟區時、Trident會嘗試使用任何使用「ONTAP-NAS-flexgroup」驅動程式的可用後端。

```
$ 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

Kubeflow部署

本節說明在Kubernetes叢集中部署Kubeflow時、必須完成的工作。

先決條件

在您執行本節所述的部署練習之前、我們假設您已經執行下列工作：

1. 您已經有一個正常運作的Kubernetes叢集、而且您正在執行Kubernetes支援的Kubernetes版本。如需支援版本的清單、請參閱 ["官方Kubeflow文件"](#)。
2. 您已在Kubernetes叢集中安裝及設定NetApp Trident、如所述 ["Trident部署與組態"](#)。

設定預設Kubernetes StorageClass

在部署Kubeflow之前、您必須在Kubernetes叢集中指定預設StorageClass。Kubeflow部署程序會嘗試使用預設StorageClass來配置新的持續磁碟區。如果沒有將StorageClass指定為預設StorageClass、則部署將會失敗。若要在叢集內指定預設StorageClass、請從部署跨接主機執行下列工作。如果您已在叢集內指定預設StorageClass、則可以跳過此步驟。

1. 將現有的其中一個StorageClass指定為預設StorageClass。以下命令範例顯示名為「ONTAP-AI - FlexVols - Retain」的StorageClass為預設StorageClass。



「ontap-non-flexgroup」Trident後端類型的最小PVc尺寸相當大。根據預設、Kubeflow會嘗試配置大小只有幾GB的PVCS。因此、您不應將使用「ONTAP-NAAS-Flexgroup」後端類型的StorageClass指定為Kubeflow部署的預設StorageClass。

```
$ 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
```

使用NVIDIA DeepOps部署Kubeflow

NetApp建議使用NVIDIA DeepOps提供的Kubeflow部署工具。若要使用DeepOps部署工具在Kubernetes叢集中部署Kubeflow、請從部署跳接主機執行下列工作。



或者、您也可以依照手動部署Kubeflow ["安裝說明"](#) 在正式的Kubeflow文件中

1. 遵循、在叢集中部署Kubeflow ["Kubeflow部署指示"](#) 在NVIDIA DeepOps GitHub網站上。
2. 記下DeepOps Kubeflow部署工具所輸出的Kubeflow儀表板URL。

```
$ ./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. 確認Kubeflow命名空間內部署的所有Pod均顯示「執行中」的「狀態」、並確認命名空間內未部署任何元件處於錯誤狀態。所有Pod可能需要幾分鐘的時間才能啟動。

```
$ 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-2qzxx			1/1
Running	0	93s	
pod/katib-suggestion-random-77b88b5c79-164j9			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-758b769d67-1lvdr			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-5fdfd9db779-17qkm            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	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-5fdfd9b779			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

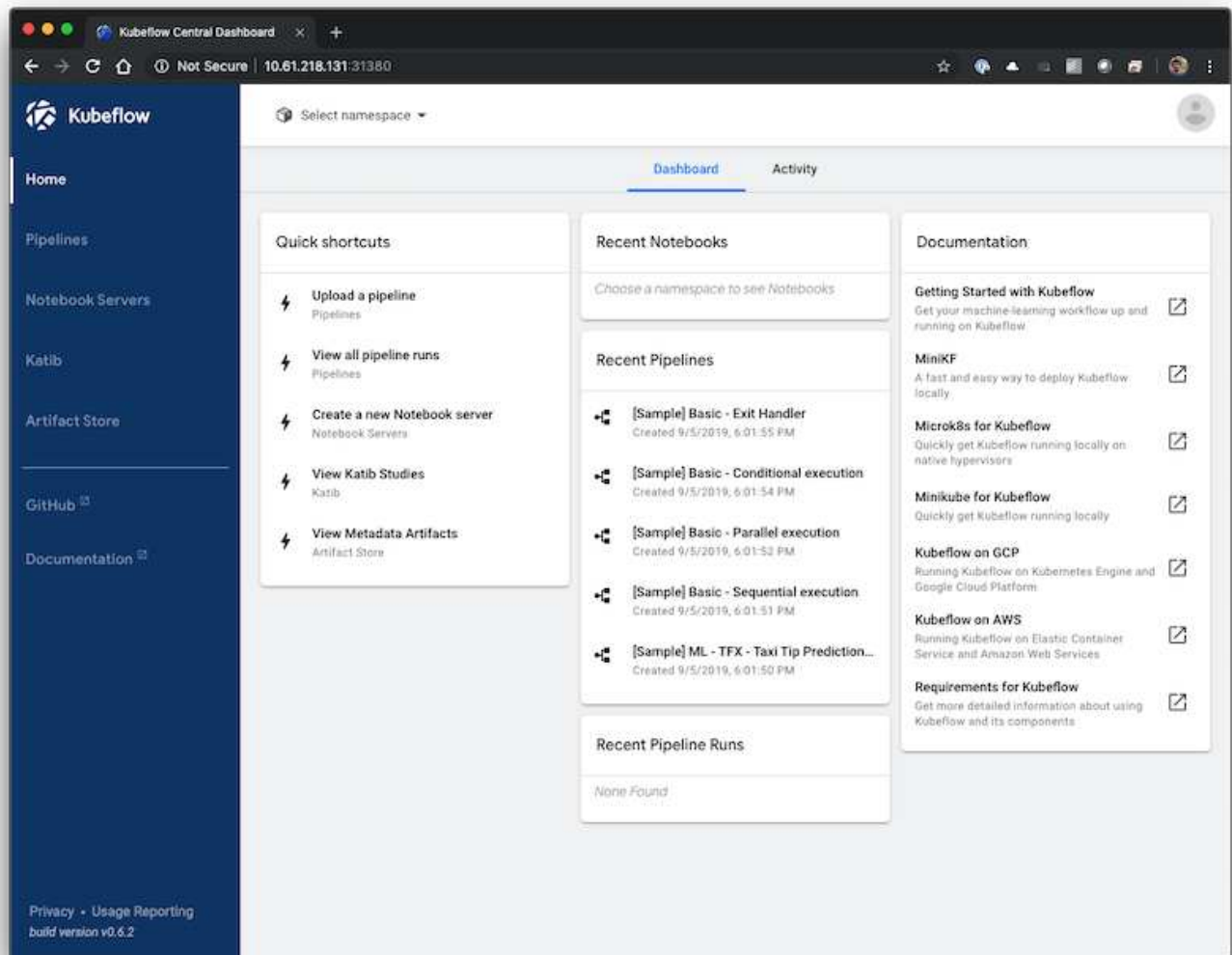
```

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. 在網頁瀏覽器中、瀏覽至您在步驟2中記下的URL、即可存取Kubeflow中央儀表板。

預設使用者名稱為「admin@kubeflow.org」、預設密碼為「12341234」。若要建立其他使用者、請遵循中的指示 "[官方Kubeflow文件](#)"。



Kubeflow作業與工作範例

本節包含使用Kubeflow執行的各種作業和工作範例。

Kubeflow作業與工作範例

本節包含使用Kubeflow執行的各種作業和工作範例。

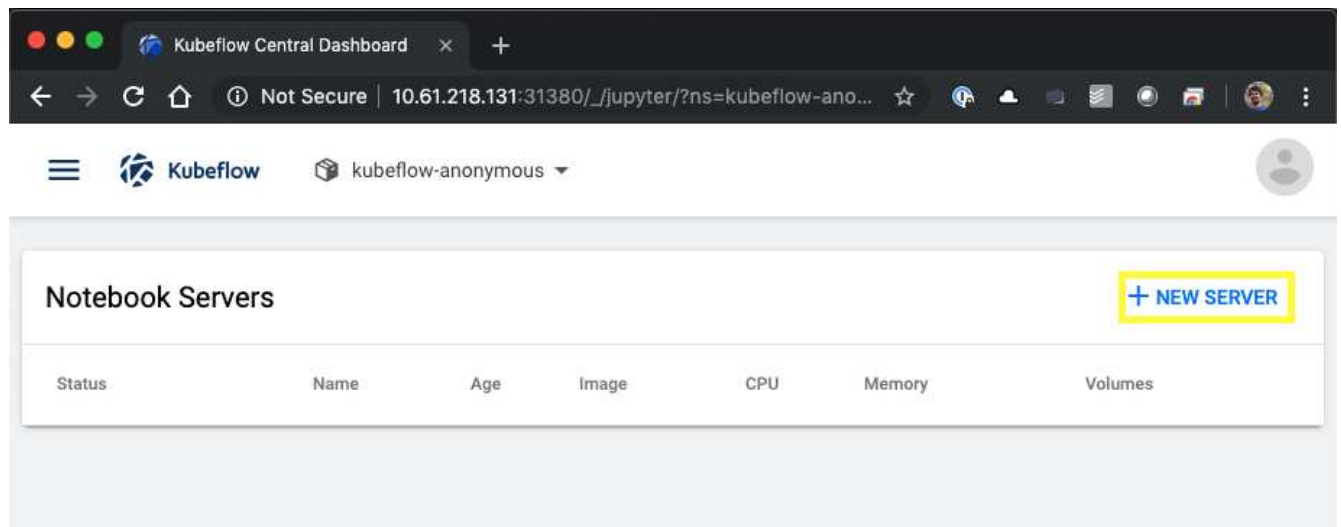
為資料科學家或開發人員提供Jupyter筆記型電腦工作區

Kubeflow能夠快速配置新的Jupyter筆記型電腦伺服器、做為資料科學家工作空間。若要使用Kubeflow來配置新的Jupyter筆記型電腦伺服器、請執行下列工作。如需Kubeflow內容中Jupyter Notebooks的詳細資訊、請參閱 "[官方Kubeflow文件](#)"。

1. 在Kubeflow中央儀表板中、按一下主功能表中的「Notebook Servers（筆記型電腦伺服器）」、瀏覽至Jupyter Notebook server管理頁面。



2. 按一下「New Server（新伺服器）」以配置新的Jupyter Notebook伺服器。



- 命名新伺服器、選擇您要伺服器所依據的Docker映像、然後指定伺服器要保留的CPU和RAM數量。如果「命名空間」欄位為空白、請使用頁面標題中的「選取命名空間」功能表來選擇命名空間。然後、「Namespace（命名空間）」欄位會自動填入所選的命名空間。

在以下範例中、會選擇「kubeflow匿名」命名空間。此外、Docker映像、CPU和RAM的預設值也會被接受。

Name

Specify the name of the Notebook Server and the Namespace it will belong to.

Name: Namespace:

Image

A starter Jupyter Docker Image with a baseline deployment and typical ML packages.

☐ Custom Image

Image:

CPU / RAM

Specify the total amount of CPU and RAM reserved by your Notebook Server. For CPU-intensive workloads, you can choose more than 1 CPU (e.g. 1.5).

CPU: Memory:

- 指定工作區Volume詳細資料。如果您選擇建立新磁碟區、則該磁碟區或永久虛擬磁碟區會使用預設StorageClass進行資源配置。因為使用Trident的StorageClass已在區段中指定為預設StorageClass "[Kubeflow部署](#)"、磁碟區或永久虛擬磁碟區是以Trident來配置。此Volume會自動掛載為Jupyter Notebook Server Container中的預設工作區。使用者在伺服器上建立的任何筆記型電腦、若未儲存至個別的資料Volume、都會自動儲存至此工作區磁碟區。因此、筆記型電腦會在重新開機後持續運作。


Workspace Volume

Configure the Volume to be mounted as your personal Workspace.

☐ Don't use Persistent Storage for User's home

Type: Name: Size: Mode: Mount Point:

- 新增資料磁碟區。下列範例指定名為「PPB -FG All」的現有PVc、並接受預設掛載點。


 **Data Volumes**

Configure the Volumes to be mounted as your Datasets.

[+ ADD VOLUME](#)


Type	Name	Size	Mode	Mount Point
Existing	pb-fg-all	10Gi	ReadWriteOnce	/home/jovyan/data-vol-1

6. *選用：*要求將所需的GPU數量分配給您的筆記型電腦伺服器。在下列範例中、需要一個GPU。

 **Configurations**

Extra layers of configurations that will be applied to the new Notebook. (e.g. Insert credentials as Secrets, set Environment Variables.)

Configurations

 **Extra Resources**

Specify extra resources that might be needed in the Notebook Server.

☒ **Enable Shared Memory**

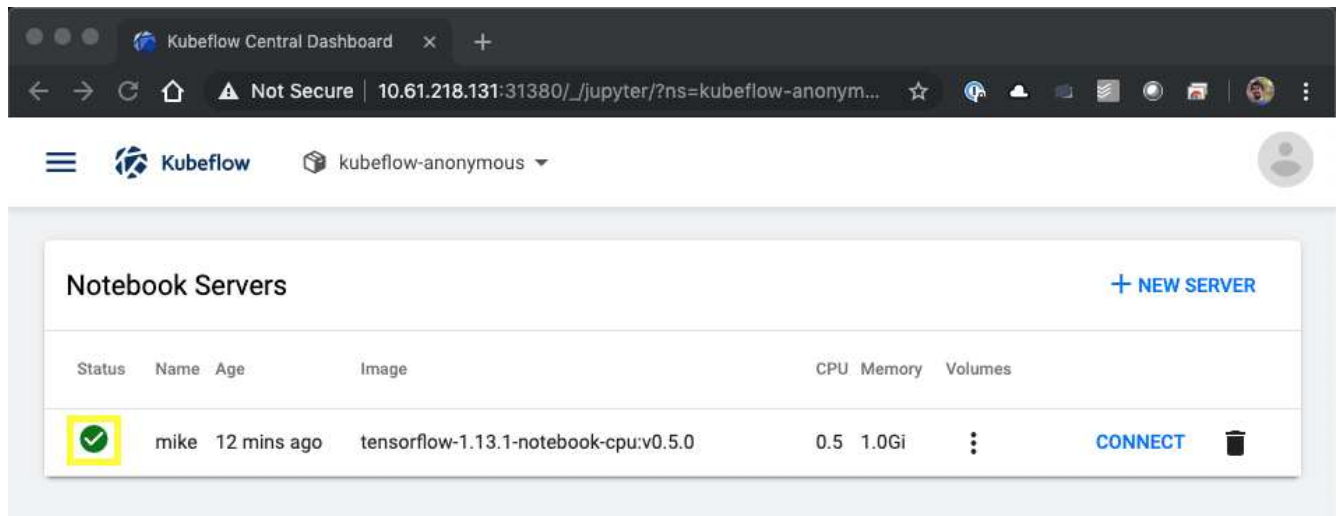
Extra Resources *

`{"nvidia.com/gpu": 1}`

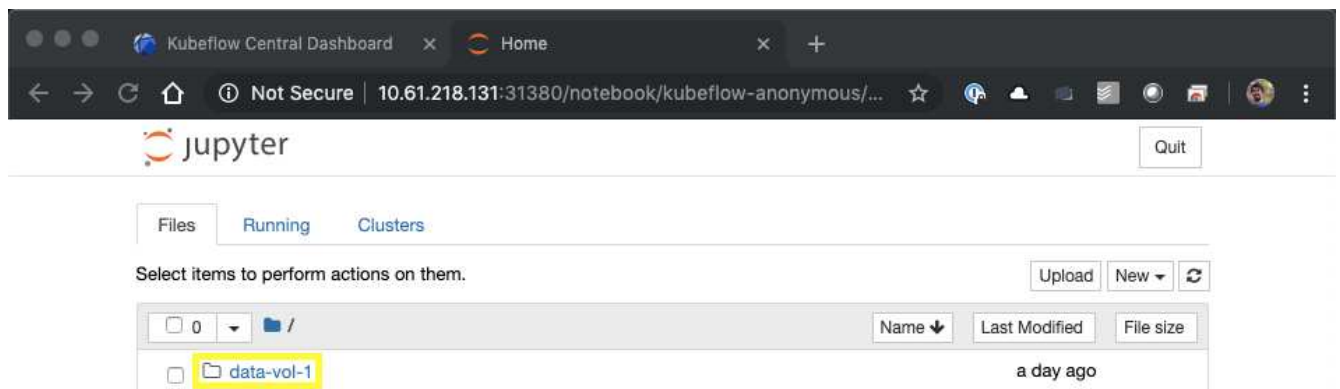
Extra Resources available in the cluster (ex. NVIDIA GPUs)

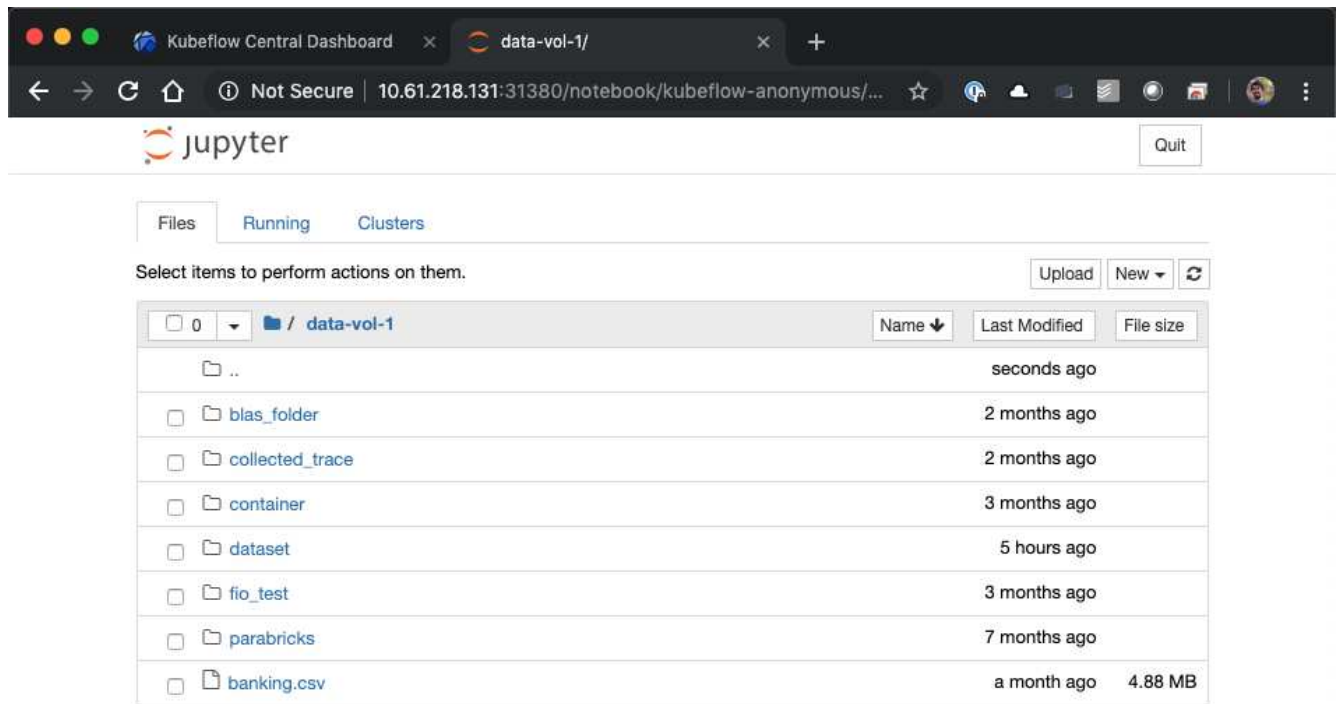
[LAUNCH](#) [CANCEL](#)

7. 按一下「啟動」以配置新的筆記型電腦伺服器。
8. 等待筆記型電腦伺服器完全配置完成。如果您從未使用您指定的Docker映像來配置伺服器、則可能需要幾分鐘的時間、因為該映像需要下載。當伺服器已完全配置完成時、您會在Jupyter筆記型電腦管理頁面的「Status（狀態）」欄中看到綠色勾號。



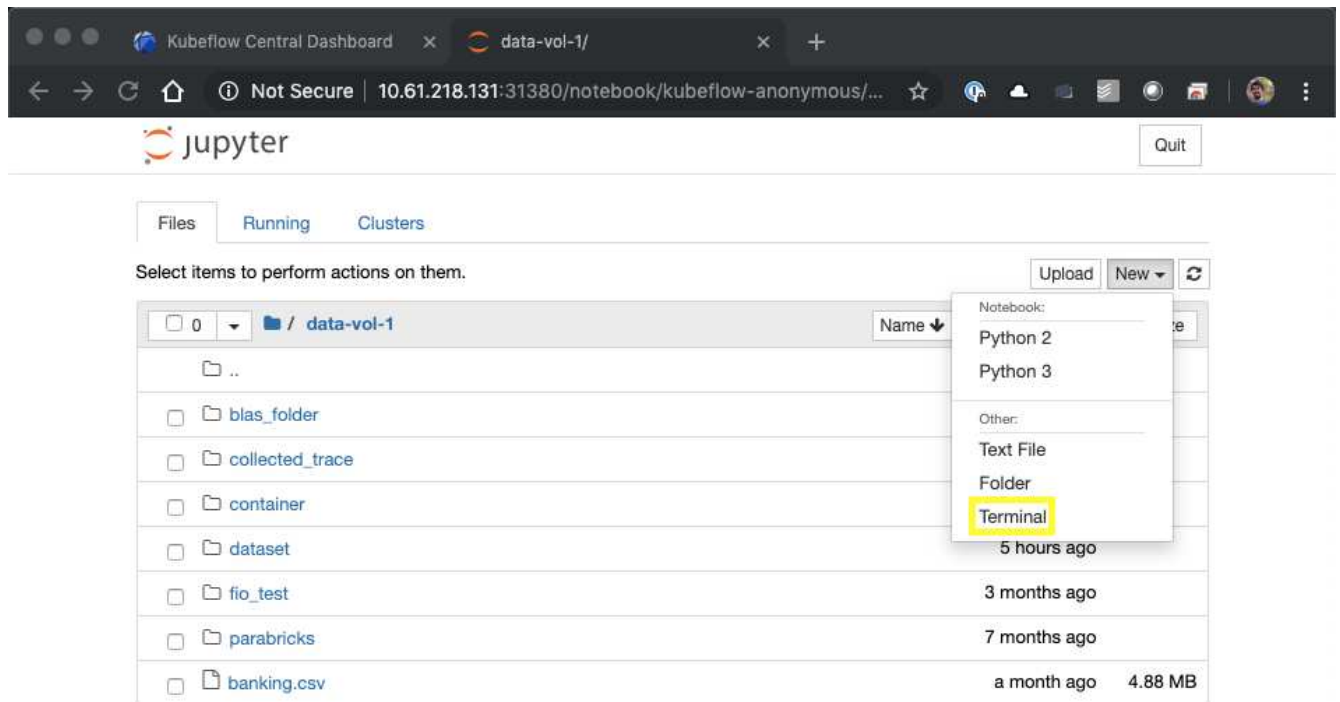
9. 按一下「連線」以連線至新的伺服器Web介面。
10. 確認已在伺服器上掛載步驟6中指定的資料集磁碟區。請注意、此磁碟區預設會掛載於預設工作區內。從使用者的觀點來看、這只是工作區內的另一個資料夾。使用者可能是資料科學家、而非基礎架構專家、因此不需要擁有任何儲存專業知識、就能使用此磁碟區。





11. 開啟終端機、並假設步驟5要求新磁碟區、執行「df -h」、確認新的Trident資源配置持續磁碟區已掛載為預設工作區。

預設工作區目錄是您第一次存取伺服器Web介面時所顯示的基礎目錄。因此、您使用Web介面建立的任何成品都會儲存在此Trident資源配置的持續Volume中。



```
Kubeflow Central Dashboard x data-vol-1/ x 10.61.218.131:31380/notebook/ x +
Not Secure | 10.61.218.131:31380/notebook/kubeflow-anonymous/mike/t...
jupyter

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

12. 使用終端機執行「nvidia-smi」、確認已將正確數量的GPU分配給筆記型電腦伺服器。在下列範例中、已依照步驟7的要求、將一個GPU分配給筆記型電腦伺服器。

```
Kubeflow Central Dashboard x Home x 10.61.218.131:31380/notebook/ x +
Not Secure | 10.61.218.131:31380/notebook/kubeflow-anonymous/mike/t...
jupyter

$ nvidia-smi
Fri Sep 13 13:52:15 2019
+-----+
| NVIDIA-SMI 410.104      Driver Version: 410.104      CUDA Version: N/A      |
+-----+
| GPU   Name               Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+
|    0   Tesla V100-SXM2...    On      | 00000000:86:00:0 Off  |           0%         0 |
| N/A   38C    P0      46W / 300W | 0MiB / 32480MiB |           0%         Default |
+-----+-----+

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

◦ ["適用於Kubernetes的NetApp Data科學工具套件"](#) 可與Kubeflow搭配使用。搭配Kubeflow使用NetApp Data科學工具套件可提供下列效益：

- 資料科學家可直接從Jupyter筆記型電腦執行進階NetApp資料管理作業。
- 進階NetApp資料管理作業可透過Kubeflow Pipes架構整合至自動化工作流程中。

請參閱 ["Kubeflow範例"](#) NetApp Data科學工具套件GitHub儲存庫中的一節、詳細說明如何搭配Kubeflow使用此工具組。

Apache Airflow部署

NetApp建議在Kubernetes上執行Apache Airflow。本節說明在Kubernetes叢集中部署氣流時、必須完成的工作。



您可以在Kubernetes以外的平台上部署氣流。在Kubernetes以外的平台上部署氣流、不在本解決方案的範圍之內。

先決條件

在您執行本節所述的部署練習之前、我們假設您已經執行下列工作：

1. 您已經擁有有效的Kubernetes叢集。
2. 您已經在Kubernetes叢集中安裝並設定NetApp Trident、如「[NetApp Trident部署與組態](#)」一節所述。

安裝Helm

我們使用適用於Kubernetes的常用套件管理程式Helm來部署氣流。部署氣流之前、您必須先在部署跨接主機上安裝Helm。若要在部署跳接主機上安裝Helm、請遵循 ["安裝說明"](#) 在官方Helm文件中。

設定預設Kubernetes StorageClass

在部署氣流之前、您必須在Kubernetes叢集中指定預設StorageClass。氣流部署程序會嘗試使用預設StorageClass來配置新的持續磁碟區。如果沒有將StorageClass指定為預設StorageClass、則部署將會失敗。若要在叢集內指定預設StorageClass、請遵循一節中所述的指示 ["Kubeflow部署"](#)。如果您已在叢集內指定預設StorageClass、則可以跳過此步驟。

使用Helm來部署氣流

若要使用Helm在Kubernetes叢集中部署氣流、請從部署跨接主機執行下列工作：

1. 請遵循、使用Helm來部署氣流 ["部署指示"](#) 以取得雜訊中心的官方氣流圖表。以下命令範例顯示如何使用Helm部署氣流。視您的環境和所需組態而定、視需要修改、新增及/或移除「custom-values.yaml」檔案中的值。

```
$ 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. 確認所有的氣流網墊都已啟動且正常運作。所有Pod可能需要幾分鐘的時間才能啟動。

```

$ 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-9d95fcd9-clf4b	2/2	Running	2	2h
airflow-web-59c94db9c5-z7rg4	1/1	Running	0	2h
airflow-worker-0	2/2	Running	2	2h

3. 請依照步驟1中使用Helm部署氣流時列印至主控台的指示、取得氣流Web服務URL。

```

$ 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. 確認您可以存取氣流Web服務。

	DAG	Schedule	Owner	Recent Tasks	Last Run	DAG Runs	Links
	ai_training_run	None	NetApp				
	create_data_scientist_workspace	None	NetApp				
	example_bash_operator	@ 0 * * * *	Airflow				
	example_branch_dop_operator_v3	* / * * * *	Airflow				
	example_branch_operator	@ daily	Airflow				
	example_complex	None	airflow				
	example_external_task_marker_child	None	airflow				
	example_external_task_marker_parent	None	airflow				
	example_http_operator	1 day, 0:00:00	Airflow				
	example_kubernetes_executor_config	None	Airflow				
	example_nested_branch_dag	@ daily	airflow				
	example_passing_params_via_test_command	* / * * * *	airflow				
	example_pig_operator	None	Airflow				
	example_python_operator	None	Airflow				
	example_short_circuit_operator	1 day, 0:00:00	Airflow				
	example_skip_dag	1 day, 0:00:00	Airflow				

Apache氣流工作流程範例

。"適用於Kubernetes的NetApp Data科學工具套件" 可搭配氣流一起使用。運用NetApp Data科學工具套件搭配氣流、您可以將NetApp資料管理作業整合到由氣流協調的自動化工作流程中。

請參閱 "氣流範例" NetApp Data科學工具套件GitHub儲存庫中的一節、詳細瞭解如何搭配氣流使用此工具組。

Trident作業範例

本節包含您可能想要使用Trident執行的各種作業範例。

匯入現有Volume

如果您的NetApp儲存系統/平台上有您要掛載到Kubernetes叢集內的容器上的現有磁碟區、但這些磁碟區並未繫結到叢集中的PVCS、則您必須匯入這些磁碟區。您可以使用Trident Volume匯入功能匯入這些Volume。

以下的命令範例顯示兩次匯入同一個磁碟區、名稱為「PB_FP_ALL」、一次匯入在本節範例中建立的每個Trident後端 "Trident後端範例、適用於ONTAP AI部署" 步驟1.以這種方式匯入相同的磁碟區兩次、可讓您

在FlexGroup 不同的生命週期內多次掛載該磁碟區（現有的一套）、如一節所述 ["Trident後端範例、適用於ONTAP AI部署"](#)步驟1.如需PVCS的詳細資訊、請參閱 ["Kubernetes官方文件"](#)。如需Volume匯入功能的詳細資訊、請參閱 ["Trident文件"](#)。

在範例的PVC規格檔案中、會指定「存取模式」值「ReadOnlyMany」。如需「存取模式」欄位的詳細資訊、請參閱 ["Kubernetes官方文件"](#)。



下列匯入命令範例中所指定的後端名稱、會對應至本節範例中所建立的後端 ["Trident後端範例、適用於ONTAP AI部署"](#)步驟1.下列範例PVC定義檔中所指定的StorageClass名稱、會對應至本節範例中所建立的StorageClass ["Kubernetes StorageClass範例ONTAP、可用於進行AI部署"](#)步驟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

```


配置新Volume

您可以使用Trident在NetApp儲存系統或平台上配置新磁碟區。下列命令範例顯示新FlexVol 的供應功能。在此範例中、磁碟區是使用本節範例中所建立的StorageClass進行資源配置 ["Kubernetes StorageClass範例ONTAP、可用於進行AI部署"步驟2](#)：

以下範例的PVC定義檔中指定「存取模式」值「ReadWriteMany」。如需「存取模式」欄位的詳細資訊、請參閱 ["Kubernetes官方文件"](#)。

```
$ 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
10995116277760                      Bound    default-pb-fg-all-iface2-85aee          10995116277760    ROX            ontap-ai-flexgroups-retain-iface2    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
```

適用於AI部署的高效能工作範例ONTAP

本節包括在ONTAP 將Kubernetes部署在AI Pod上時、可執行的各種高效能工作範例。

適用於AI部署的高效能工作範例ONTAP

本節包括在ONTAP 將Kubernetes部署在AI Pod上時、可執行的各種高效能工作範例。

執行單節點AI工作負載

若要在Kubernetes叢集中執行單節點AI和ML工作、請從部署跳接主機執行下列工作。有

了Trident、您就能快速輕鬆地建立資料磁碟區、讓Kubernetes工作負載能夠存取可能含有PB資料的資料。若要從Kubernetes Pod中存取此類資料磁碟區、只需在Pod定義中指定一個PVC即可。此步驟為Kubernetes原生作業、不需要NetApp專業人員。



本節假設您已將您嘗試在Kubernetes叢集中執行的特定AI和ML工作負載（採用Docker容器格式）容器化。

1. 下列命令範例顯示使用ImageNet資料集的TensorFlow基準測試工作負載建立Kubernetes工作。如需ImageNet資料集的詳細資訊、請參閱 ["ImageNet網站"](#)。

此範例工作要求八個GPU、因此可在單一GPU工作節點上執行、該工作節點具備八個或更多GPU。此範例工作可在叢集中提交、而具有八個以上GPU的工作節點不存在、或目前正與其他工作負載一起使用。如果是、則工作會維持在擱置狀態、直到該工作者節點可供使用為止。

此外、為了將儲存頻寬最大化、包含所需訓練資料的磁碟區會在本工作所建立的Pod內掛載兩次。另外一個Volume也會掛載在Pod中。第二個磁碟區將用於儲存結果和指標。這些磁碟區會使用PVCS名稱在工作定義中參考。如需Kubernetes工作的詳細資訊、請參閱 ["Kubernetes官方文件"](#)。

此範例所建立的Pod中、會將「medium」值為「memory」的「emptyDir」磁碟區掛載到「開發/shm」。Docker Container執行時間所自動建立的「/dev/shm」虛擬磁碟區的預設大小、有時可能不足以滿足TensorFlow的需求。如以下範例所示、掛載「emptyDir」磁碟區可提供足夠大的「/dev/shm」虛擬磁碟區。如需有關「emptyDir」Volume的詳細資訊、請參閱 ["Kubernetes官方文件"](#)。

在此範例工作定義中所指定的單一容器、其「優先」值為「true」。此值表示容器有效擁有主機root存取權。在這種情況下會使用此註釋、因為執行的特定工作負載需要root存取權。具體而言、工作負載執行的清除快取作業需要root存取權。是否需要這種「特殊權限：真」註解、取決於您執行的特定工作負載需求。

```
$ 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. 確認您在步驟1中建立的工作正在正確執行。下列範例命令可確認已為工作建立單一Pod（如工作定義所指定）、而且此Pod目前正在其中一個GPU工作節點上執行。

```

$ 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. 確認您在步驟1中建立的工作已成功完成。下列命令範例可確認工作已成功完成。

```

$ 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. *選用：*清除工作成品。下列命令範例顯示刪除在步驟1中建立的工作物件。

刪除工作物件時、Kubernetes會自動刪除任何相關的Pod。

```

$ 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.

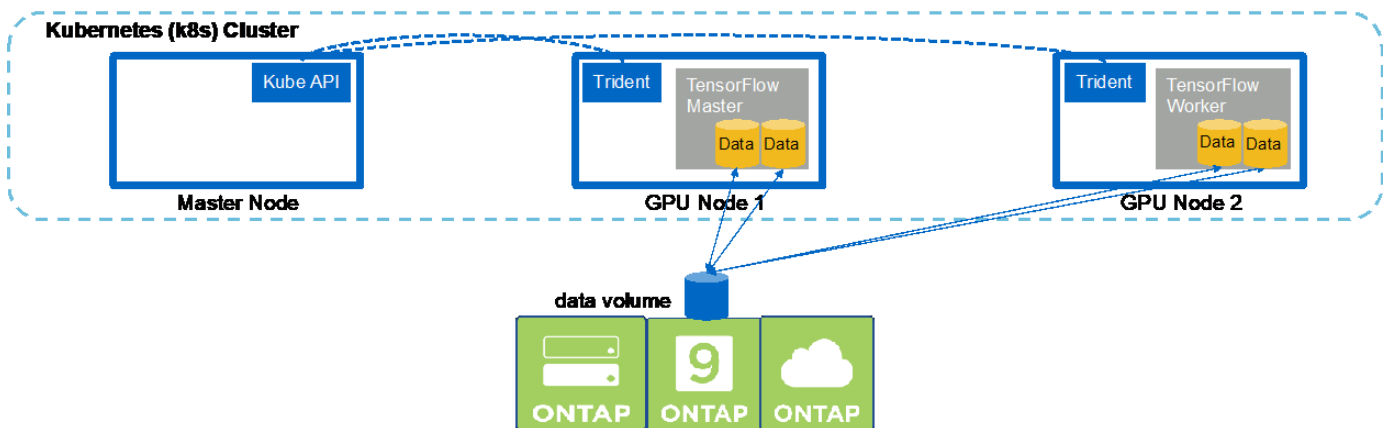
```

執行同步分散式AI工作負載

若要在Kubernetes叢集中執行同步多節點AI和ML工作、請在部署跨接主機上執行下列工作。此程序可讓您利用儲存在NetApp磁碟區上的資料、並使用比單一工作節點更多的GPU。如需同步分散式AI工作的說明、請參閱下圖。



相較於非同步分散式工作、同步分散式工作有助於提升效能和訓練準確度。關於同步工作與非同步工作的優缺點的討論、不在本文的討論範圍之內。



1. 下列命令範例顯示建立一個工作者、以參與本節範例中單一節點上執行的相同TensorFlow基準測試工作之同步分散式執行 "[執行單節點AI工作負載](#)"。在此特定範例中、只會部署一名員工、因為該工作會在兩個工作節點之間執行。

此範例的工作者部署要求八個GPU、因此可在單一GPU工作者節點上執行、該節點具備八個以上的GPU。如果GPU工作節點的GPU功能超過八個GPU、為了發揮最大效能、您可能想要增加此數目、使其等於工作節點所使用的GPU數量。如需Kubernetes部署的詳細資訊、請參閱 "[Kubernetes官方文件](#)"。

在此範例中會建立Kubernetes部署、因為這個特定的容器化工作者永遠不會自行完成。因此、使用Kubernetes工作架構來部署IT並不合理。如果您的員工是自行設計或撰寫完成、則使用工作架構來部署您的員工可能是合理的做法。

本範例部署規格中所指定的Pod、其「hostNetwork」值為「true」。此值表示Pod使用主機工作節點的網路堆疊、而非Kubernetes通常為每個Pod建立的虛擬網路堆疊。此註釋用於此案例、因為特定工作負載仰賴Open MPI、NCCL和Horovod以同步分散的方式執行工作負載。因此、它需要存取主機網路堆疊。關於Open MPI、NCCL和Horovod的討論不在本文的討論範圍之內。是否需要此「hostNetwork: true」註釋、取決於您執行的特定工作負載需求。如需有關「hostNetwork」欄位的詳細資訊、請參閱 ["Kubernetes官方文件"](#)。

```
$ 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. 確認您在步驟1中建立的工作者部署已成功啟動。下列命令範例可確認已針對部署建立單一工作者Pod、如部署定義所示、而且此Pod目前正在其中一個GPU工作者節點上執行。

```

$ 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. 為啟動、參與及追蹤同步多節點工作執行的主節點建立Kubernetes工作。下列命令範例可建立一個主磁片、用於啟動、參與及追蹤同一個TensorFlow基準測試工作的同步分散式執行、該工作是在本節範例的單一節點上執行 ["執行單節點AI工作負載"](#)。

此範例主要工作要求八個GPU、因此可在具有八個以上GPU的單一GPU工作節點上執行。如果GPU工作節點的GPU功能超過八個GPU、為了發揮最大效能、您可能想要增加此數目、使其等於工作節點所使用的GPU數量。

本範例工作定義中所指定的主Pod、其「主機網路」值為「真」、就如同在步驟1中給工作群組「主機網路」值「真」一樣。請參閱步驟1、瞭解為何需要此值的詳細資訊。

```

$ 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. 確認您在步驟3中建立的主要工作正在正確執行。下列範例命令可確認已為工作建立單一主Pod、如工作定義所示、而且此Pod目前正在其中一個GPU工作節點上執行。您也應該看到、您在步驟1中看到的工作者Pod仍在執行中、而且主要和工作者Pod正在不同的節點上執行。

```
$ kubectl get pods -o wide
```

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

5. 確認您在步驟3中建立的主要工作已成功完成。下列命令範例可確認工作已成功完成。

```
$ kubectl get jobs
```

NAME	COMPLETIONS	DURATION	AGE
netapp-tensorflow-multi-imagenet-master	1/1	5m50s	9m18s

```
$ kubectl get pods
```

NAME	STATUS	RESTARTS	AGE	READY
netapp-tensorflow-multi-imagenet-master-ppwwj	Completed	0	9m38s	0/1
netapp-tensorflow-multi-imagenet-worker-654fc7f486-v6725	Running	0	35m	1/1

```
$ 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 obl -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_benchmarks.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_imagenet_nodistort_fp16_r10_m2_nockpt.txt 2>&1

```

6. 當您不再需要部署時、請刪除該員工部署。下列命令範例顯示刪除在步驟1中建立的工作者部署物件。

當您刪除工作者部署物件時、Kubernetes會自動刪除任何關聯的工作者Pod。

```

$ 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. *選用：*清除主要工作成品。下列命令範例顯示刪除在步驟3中建立的主要工作物件。

刪除主工作物件時、Kubernetes會自動刪除任何相關的主Pod。

```

$ 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.

```

效能測試

我們在建立此解決方案時、進行了簡單的效能比較。我們使用Kubernetes執行數項標準NetApp AI基準測試工作、並將基準測試結果與使用簡易Docker RUN命令執行的執行結果進行比較。我們並未發現效能有任何顯著差異。因此、我們的結論是、使用Kubernetes來協調容器化AI訓練工作、並不會對效能造成負面影響。請參閱下表以取得效能比較結果。

基準測試	資料集	Docker Run (影像/秒)	Kubernetes (影像/秒)
單節點TensorFlow	綜合資料	6、667.2475	6、661.93125
單節點TensorFlow	ImageNet	6,570.2025	6、530.59125
同步分散式雙節點TensorFlow	綜合資料	13、213.70625	13、218.288125
同步分散式雙節點TensorFlow	ImageNet	12、941.69125	12、881.33875

結論

各種規模的公司和組織、以及所有產業、紛紛轉向人工智慧（AI）、機器學習（ML）和深度學習（DL）、以解決實際問題、提供創新產品和服務、並在競爭日益激烈的市場中獲得優勢。隨著企業組織增加AI、ML和DL的使用率、他們面臨許多挑戰、包括工作負載擴充性和資料可用度。這些挑戰可透過使用NetApp AI Control Plane解決方案來解決。

此解決方案可讓您快速複製資料命名空間。此外、它還可讓您定義及實作AI、ML及DL訓練工作流程、並整合近乎即時的資料建立與模型基準、以利追蹤及版本管理。有了這套解決方案、您可以追蹤每個模型訓練、並將其重新傳回模型訓練及/或驗證的確切資料集。最後、此解決方案可讓您迅速配置Jupyter Notebook工作區、並存取大量的資料集。

由於本解決方案的目標對象是資料科學家和資料工程師、ONTAP 因此需要最少的NetApp或NetApp的支援專業知識。有了這套解決方案、您就能使用簡單且熟悉的工具和介面來執行資料管理功能。此外、此解決方案還充分運用開放原始碼和免費元件。因此、如果您的環境中已經有NetApp儲存設備、現在就可以實作此解決方案。如果您想試用此解決方案、但尚未擁有NetApp儲存設備、請造訪 ["cloud.netapp.com"](https://cloud.netapp.com)而且您可以立即使用雲端型NetApp儲存解決方案來啟動和執行。

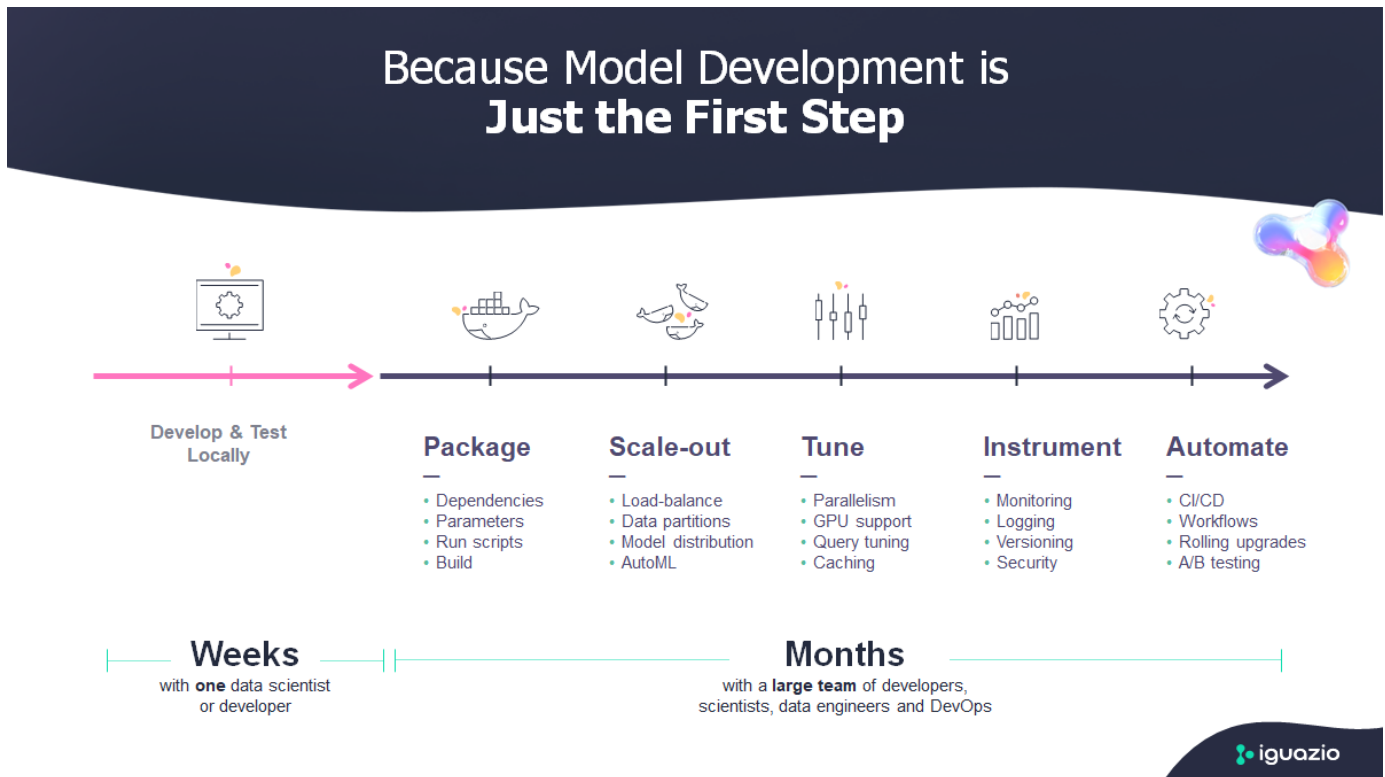
採用Iguazio的MLRun Pipeline

TR-4834：適用於MLRun Pipeline的NetApp和Iguazio

Rick Huang、David Arette、NetApp Marcelo Lettovsky、Iguazio

本文件涵蓋採用NetApp ONTAP AI、NetApp AI Control Plane、NetApp Cloud Volumes軟體及Iguazio Data Science Platform的MLRun管線詳細資料。我們使用Nuclio無伺服器功能、Kubernetes持續磁碟區、NetApp Cloud Volumes、NetApp Snapshot複本、Grafana儀表板、以及Iguazio平台上的其他服務、以建置端點對端點資料傳輸途徑來模擬網路故障偵測。我們整合了Iguazio和NetApp技術、可在內部部署和雲端上實現快速的模型部署、資料複寫和正式作業監控功能。

資料科學家的工作重點應放在機器學習（ML）和人工智慧（AI）模式的訓練和調校。不過、根據Google的研究結果、資料科學家將約80%的時間花在研究如何讓他們的模型與企業應用程式搭配運作、並以大規模方式執行、如以下影像所示、說明AI/ML工作流程中的模型開發。



若要管理端點對端點的AI/ML專案、需要更深入瞭解企業元件。雖然DevOps已接管這些類型元件的定義、整合及部署、但機器學習作業的目標是類似流程、包括AI/ML專案。若要瞭解企業中端點對端點的AI/ML傳輸途徑、請參閱下列必要元件清單：

- 儲存設備
- 網路
- 資料庫
- 檔案系統
- 容器
- 持續整合與持續部署（CI/CD）管道
- 開發整合式開發環境（IDE）
- 安全性
- 資料存取原則
- 硬體
- 雲端
- 虛擬化
- 資料科學工具集與程式庫

在本白皮書中、我們將示範NetApp與Iguazio之間的合作關係如何大幅簡化端點對端點AI/ML管線的開發。這項

簡化可加速所有AI / ML應用程式的上市時間。

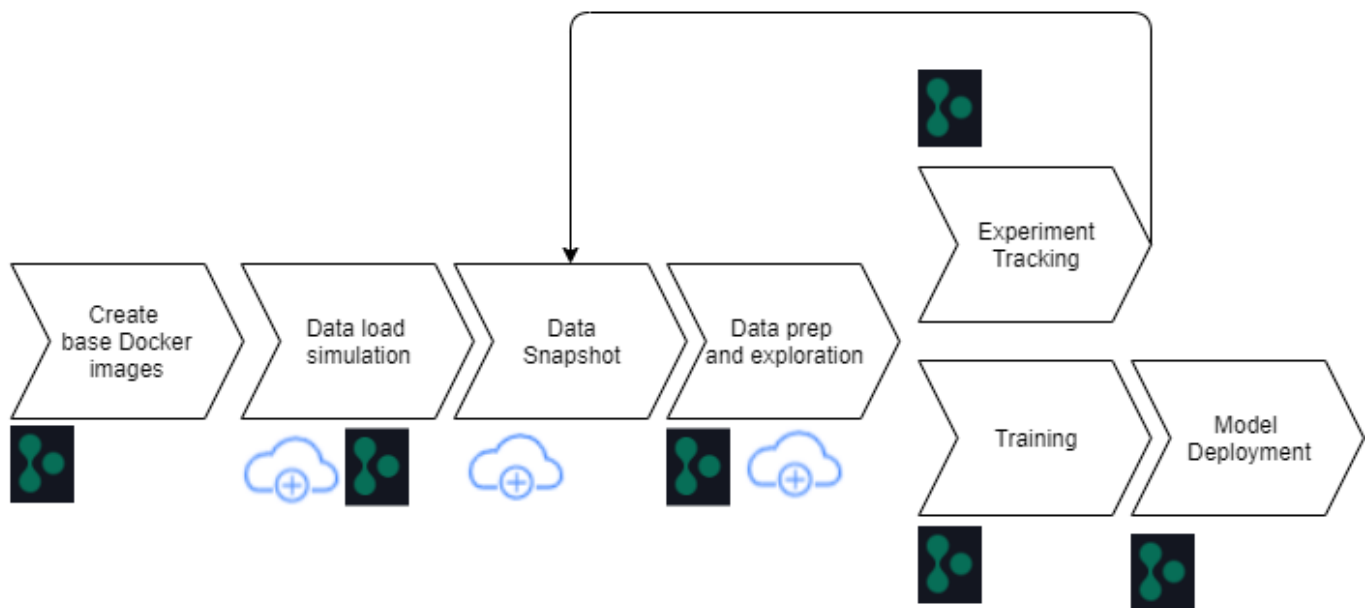
目標對象

資料科學的世界涉及資訊技術和商業的多個領域。

- 資料科學家需要靈活運用所選的工具和程式庫。
- 資料工程師需要知道資料的流通方式及存放位置。
- DevOps工程師需要工具、將新的AI/ML應用程式整合至其CI/CD管道。
- 企業使用者想要存取AI / ML應用程式。我們說明NetApp和Iguazio如何協助這些角色、讓我們的平台為企業帶來價值。

解決方案總覽

此解決方案遵循AI / ML應用程式的生命週期。我們從資料科學家的工作開始著手、定義準備資料、以及訓練和部署模型所需的不同步驟。我們會依照所需的工作來建立完整的管道、以便追蹤成品、實驗執行、以及部署至Kubeflow。為了完成整個週期、我們將管線與NetApp Cloud Volumes整合、以啟用資料版本管理功能、如下圖所示。



技術總覽

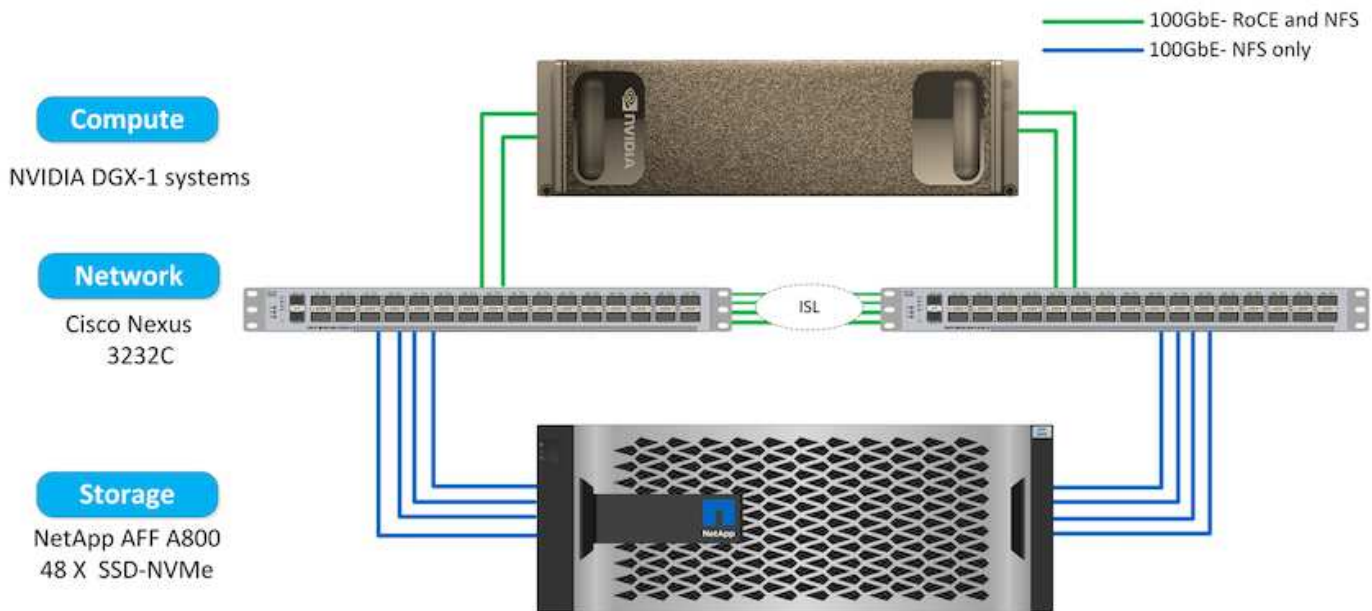
NetApp總覽

NetApp是混合雲的資料權威。NetApp提供全系列的混合雲資料服務、可簡化跨雲端和內部部署環境的應用程式與資料管理、加速數位轉型。NetApp與我們的合作夥伴一起、讓全球組織能夠充分發揮資料的完整潛能、以擴大客戶接觸點、促進更多創新、並最佳化營運。

NetApp ONTAP AI

NetApp ONTAP 支援NVIDIA DGX系統與NetApp雲端連線的All Flash儲存設備、能夠可靠地簡化資料流、並利用從邊緣到核心到雲端的資料架構、加速分析、訓練和推斷。它為IT組織提供以下優點的架構：

- 消除設計複雜性
- 可獨立擴充運算與儲存設備
- 讓客戶從小規模開始、並無縫擴充
- 提供多種儲存選項、適用於各種效能與成本點NetApp ONTAP 支援融合式基礎架構堆疊、結合NVIDIA DGX-1、Petaflop等級AI系統、以及NVIDIA Mellanox高效能乙太網路交換器、可統一處理AI工作負載、簡化部署並加速ROI。我們在ONTAP 本技術報告中運用包含一個DGX-1和NetApp AFF 的NetApp供應系統的AI技術。下圖顯示ONTAP 本驗證所使用之DGX-1系統的支援範本。



NetApp AI Control Plane

NetApp AI Control Plane提供極佳的擴充性、簡化的部署、以及不中斷的資料可用度、讓您充分發揮AI和ML的威力。AI Control Plane解決方案將Kubernetes和Kubeflow與NetApp支援的資料架構整合。Kubernetes是適用於雲端原生部署的業界標準容器協調平台、可提供工作負載擴充性與可攜性。Kubeflow是開放原始碼的機器學習平台、可簡化管理與部署作業、讓開發人員能以更短的時間完成更多的資料科學工作。採用NetApp技術的資料架構可提供毫不妥協的資料可用度與可攜性、確保您的資料可在整個通路中存取、從邊緣到核心到雲端。本技術報告使用MLRun管道中的NetApp AI Control Plane。下圖顯示Kubernetes叢集管理頁面、您可以在其中為每個叢集設定不同的端點。我們將NFS持續磁碟區連線至Kubernetes叢集、下列映像顯示連接至叢集的持續磁碟區、位於何處 "NetApp Trident" 提供持續的儲存支援與資料管理功能。

4 Kubernetes Clusters



kubernetes



https://3.20.111.39:6443
Cluster Endpoint



v1.15.5
Cluster Version



19.07.1
Trident Version



0
Working Environments



kubernetes



https://172.31.14.31:6443
Cluster Endpoint



v1.15.5
Cluster Version



19.07.1
Trident Version



1
Working Environments

Persistent Volumes for Kubernetes

Connected with Kubernetes Cluster

Cloud Volumes ONTAP is connected to 1 Kubernetes cluster. [View Cluster](#)

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

Kubernetes Cluster

Custom Export Policy *(Optional)*

Select Kubernetes Cluster

kubernetes

Custom Export Policy

172.31.0.0/16


☒ Set as default storage class☒ NFS ☐ iSCSI

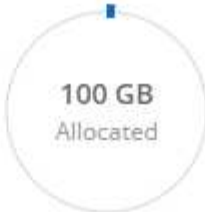
Connect

Cancel

Volumes

4 Volumes | 300 GB Allocated | 1.43 GB Total Used

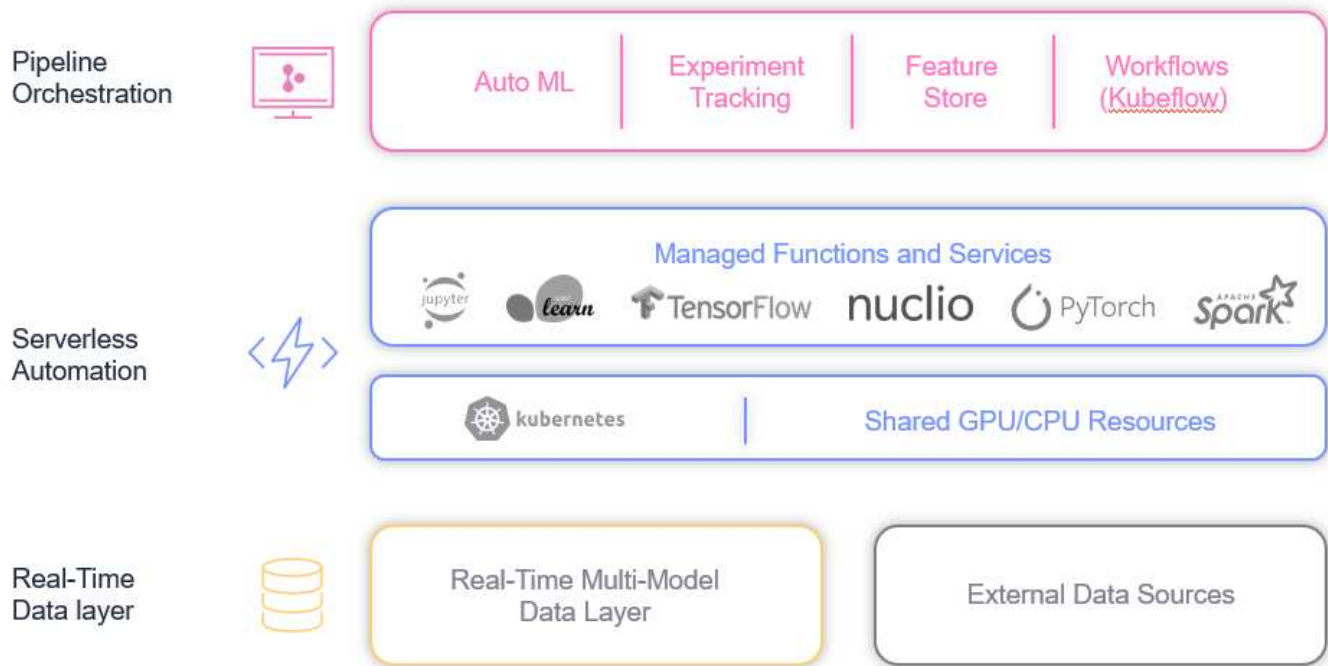

kubernetes_trident_pvc_551720fa_3758_461...
ONLINE

INFO		CAPACITY	
Disk Type	GP2	 <div>1.25 GB EBS Used</div>	
Tiering Policy	None		
Backup	OFF		

Iguazio概述

Iguazio Data科學平台是完全整合且安全的資料科學平台即服務（PaaS）、可簡化開發、加速效能、促進協同作業、並解決營運挑戰。此平台整合了下列元件、Iguazio Data科學平台以下列影像呈現：

- 資料科學工作平台、包括Jupyter筆記型電腦、整合式分析引擎和Python套件
- 利用實驗追蹤和自動化管道功能來建立模型管理
- 透過可擴充的Kubernetes叢集來管理資料和ML服務
- Nuclio是即時無伺服器功能架構
- 極為快速且安全的資料層、支援SQL、NoSQL、時間序列資料庫、檔案（簡單物件）和串流
- 與第三方資料來源整合、例如NetApp、Amazon S3、HDFS、SQL資料庫、串流或訊息傳輸協定
- 以Grafana為基礎的即時儀表板



軟體與硬體需求

網路組態

以下是在雲端設定的網路組態需求：

- Iguazio叢集和NetApp Cloud Volumes必須位於相同的虛擬私有雲中。
- 雲端管理員必須能夠存取Iguazio應用程式節點上的連接埠6443。
- 我們在本技術報告中使用Amazon Web Services。不過、使用者可以選擇在任何雲端供應商中部署解決方案。為了在ONTAP 採用NVIDIA DGX-1的AI中進行內部部署測試、我們使用Iguazio代管DNS服務、以方便使用。

用戶端必須能夠存取動態建立的DNS網域。客戶可視需要使用自己的DNS。

硬體需求

您可以在自己的叢集內建安裝Iguazio。我們已使用ONTAP NVIDIA DGX-1系統、驗證NetApp的解決方案。下表列出用於測試此解決方案的硬體。

硬體	數量
DGX-1系統	1.
NetApp AFF 產品系列A800系統	1個高可用度（HA）配對、包括2個控制器和48個NVMe SSD（3.8TB或以上）
Cisco Nexus 3232C網路交換器	2.

下表列出內部部署測試所需的軟體元件：

軟體	版本或其他資訊
NetApp ONTAP 資料管理軟體	9.7
Cisco NX-OS交換器韌體	7.0 (3) I6 (1)
NVIDIA DGX OS	4.4 - Ubuntu 18.04 LTS
Docker Container平台	19.03.5
Container版本	20.01-tf1-py2.
機器學習架構	TensorFlow 1.15.0
Iguazio	版本2.8+
ESX Server	6.5

此解決方案已通過Iguazio 2.5版和NetApp Cloud Volumes ONTAP 的AWS版完整測試。Iguazio叢集和NetApp軟體都在AWS上執行。

軟體	版本或類型
Iguazio	版本2.8+
應用程式節點	M5.4xLarge
資料節點	I3.4xLarge

網路裝置故障預測使用案例摘要

此使用案例是以亞洲電信領域的Iguazio客戶為基礎。由於每年有10萬家企業客戶和125萬次網路中斷事件、因此必須預測並採取主動行動、避免網路故障影響客戶。此解決方案提供給他們下列效益：

- 預測性網路故障分析
- 與票務系統整合
- 採取主動行動預防網路故障因為這項Iguazio實作的結果、60%的故障都是主動預防的。

設定總覽

Iguazio可安裝在內部部署或雲端供應商上。

Iguazio安裝

資源配置可做為服務、並由Iguazio或客戶進行管理。在這兩種情況下、Iguazio都提供部署應用程式（Provazio）來部署及管理叢集。

如需內部部署安裝、請參閱 "[NVA-1121](#)." 適用於運算、網路和儲存設備設定。Iguazio提供內部部署Iguazio、客戶無需支付額外成本。請參閱 "[本頁](#)" 適用於DNS和SMTP伺服器組態。Provazio安裝頁面如下所示。

×

New System (dev)

Installation Scenario

General

Clusters

Cloud

Bare metal / virtual machines

Installs the system on bare-metal or virtual-machine instances, pre-provisioned with prerequ...

AWS

Creates applicable compute/networking resources in AWS and installs the system on the in...

Azure

Creates applicable compute/networking resources in Azure and installs the system on the i...

AWS (pre-provisioned)

Installs the system on Amazon Web Services instances, manually provisioned beforehand

Azure (pre-provisioned)

Installs the system on Microsoft Azure instances, manually provisioned beforehand

Advanced

Show advanced options in the next steps

BACK

NEXT

設定Kubernetes叢集



本節分為兩部分、分別用於雲端和內部部署。


雲端部署Kubernetes組態

透過NetApp Cloud Manager、您可以定義與Iguazio Kubernetes叢集的連線。Trident需要存取叢集中的多個資源、才能使磁碟區可用。

1. 若要啟用存取、請從其中一個Iguazio節點取得Kubernetes組態檔案。檔案位於「`/home/iguazio/.kue/config`」下 將此檔案下載至桌面。
2. 前往探索叢集進行設定。

4 Kubernetes Clusters

 kubernetes			
 https://3.20.111.39:6443 Cluster Endpoint	 v1.15.5 Cluster Version	 19.07.1 Trident Version	 0 Working Environments

 kubernetes			
 https://172.31.14.31:6443 Cluster Endpoint	 v1.15.5 Cluster Version	 19.07.1 Trident Version	 1 Working Environments

3. 上傳Kubernetes組態檔案。請參閱下圖。

Upload Kubernetes Configuration File

Upload the Kubernetes configuration file (kubeconfig) so Cloud Manager can install Trident on the Kubernetes cluster.

Connecting Cloud Volumes ONTAP with a Kubernetes cluster enables users to request and manage persistent volumes using native Kubernetes interfaces and constructs. Users can take advantage of ONTAP's advanced data management features without having to know anything about it. Storage provisioning is enabled by using NetApp Trident.

Learn more about [Trident for Kubernetes](#).

Upload File

4. 部署Trident、並將磁碟區與叢集建立關聯。請參閱下列影像、瞭解如何定義及指派持續Volume給Iguazio叢集。此程序可在Iguazio的Kubernetes叢集中建立持續Volume（PV）。在使用之前、您必須先定義持續磁碟區宣告（PVC）。

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

內部部署Kubernetes組態

如需NetApp Trident的內部部署安裝、請參閱 ["TR-4798"](#) 以取得詳細資料。在設定Kubernetes叢集並安裝NetApp Trident之後、您可以將Trident連線至Iguazio叢集、以啟用NetApp資料管理功能、例如建立資料與模型的Snapshot複本。

定義持續Volume宣告

1. 將下列Yaml儲存至檔案、以建立Basic類型的PVC。

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

2. 將Yaml檔案套用至您的Iguazio Kubernetes叢集。

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

將**NetApp Volume**附加至**Jupyter**筆記型電腦

Iguazio提供多種託管服務、可為資料科學家提供完整的端點對端堆疊、以供開發及部署AI / ML應用程式。如需這些元件的詳細資訊、請參閱 "[Iguazio應用程式服務與工具總覽](#)"。

其中一項託管服務是Jupyter Notebook。每個開發人員都能自行部署筆記型電腦容器、並提供開發所需的資源。若要讓他們存取NetApp Cloud Volume、您可以將Volume指派給他們的容器、並在下列映像中顯示持續Volume Claims的資源配置、執行中的使用者及環境變數設定。

如需內部部署組態、請參閱 "[TR-4798](#)" 在Trident設定中啟用NetApp ONTAP 的支援資料管理功能、例如為資料或模型製作Snapshot複本以進行版本控制。在Trident後端組態檔中新增下列行、以顯示Snapshot目錄：

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

您必須以Json格式建立Trident後端組態檔、然後執行下列命令 "[Trident命令](#)" 若要參考：

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

Enabled

Inactivity window: 5m, 10m, 1h, 2h, 4h

Resources

For more information about the resource parameters, see [Kubernetes documentation](#).

The memory and CPU configurations are applied to each replica.

Memory: Request: GB, Limit: GB

CPU: Request: Example: 1500 millicpu, Limit: Example: 1500 millicpu

Running User: admin

Flavor: Full stack without GPU

Spark: spark

Environment Variables

Create a new environment variable

Persistent Volume Claims (PVCs)

Name	Mount Path
basic	/nctapp

Add PVC

部署應用程式

下列各節將說明如何安裝及部署應用程式。

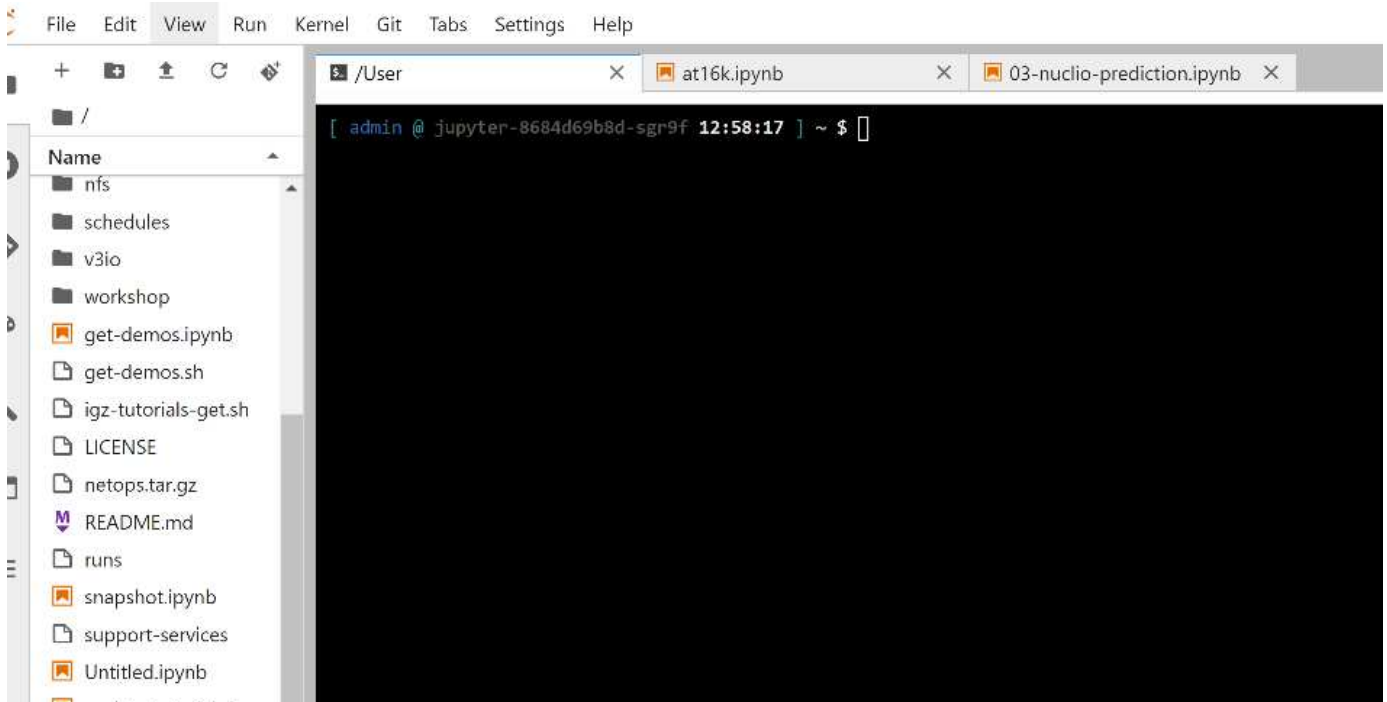
從[GitHub](#)取得程式碼

現在NetApp Cloud Volume或NetApp Trident Volume可用於Iguazio叢集和開發人員環境、

您就可以開始檢閱應用程式。

使用者擁有自己的工作區（目錄）。在每個筆記型電腦上、使用者目錄的路徑都是「/使用者」。Iguazio平台負責管理目錄。如果您依照上述指示操作、NetApp Cloud Volume可在「/NetApp」目錄中找到。

使用Jupyter終端從GitHub取得程式碼。



在Jupyter終端機提示字元下、複製專案。

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

現在您應該可以在Jupyter工作區的檔案樹狀結構中看到「netops-netapp」資料夾。

設定工作環境

將「Notebook」（筆記型電腦）的「set_env-example.ipynb」複製為「set_env.ipynb」。開啟並編輯「set_env.ipynb」。此筆記型電腦可設定認證資料、檔案位置及執行驅動程式的變數。

如果您依照上述指示進行、下列步驟是您唯一要做的變更：

1. 從Iguazio服務儀表板取得此值：「Docker_registry」

範例：「Docker-registry.default-tenant.app.clusterq.iguaziodev.com:80」

2. 將「admin」變更為您的Iguazio使用者名稱：

「IGZ_container路徑='/user/admin'」

以下ONTAP 是有關「系統連線」的詳細資料。包括安裝Trident時所產生的Volume名稱。下列設定適用於內部部署ONTAP 的內部部署的叢集：

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

下列設定適用於Cloud Volumes ONTAP 下列項目：

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

建立基礎Docker映像檔

Iguazio平台包含建置ML管線所需的一切。開發人員可以定義執行管線所需的Docker映像規格、以及從Jupyter Notebook建立映像。開啟筆記型電腦「create- images.ipynb」、然後執行「All」儲存格。

這款筆記型電腦會建立兩個我們在銷售管道中使用的映像。

- 《iguazio/NetApp》 用於處理ML工作。

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」。包含處理NetApp Snapshot複本的公用程式。

Create image for Ontap utilites

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

檢閱個別的Jupyter筆記型電腦

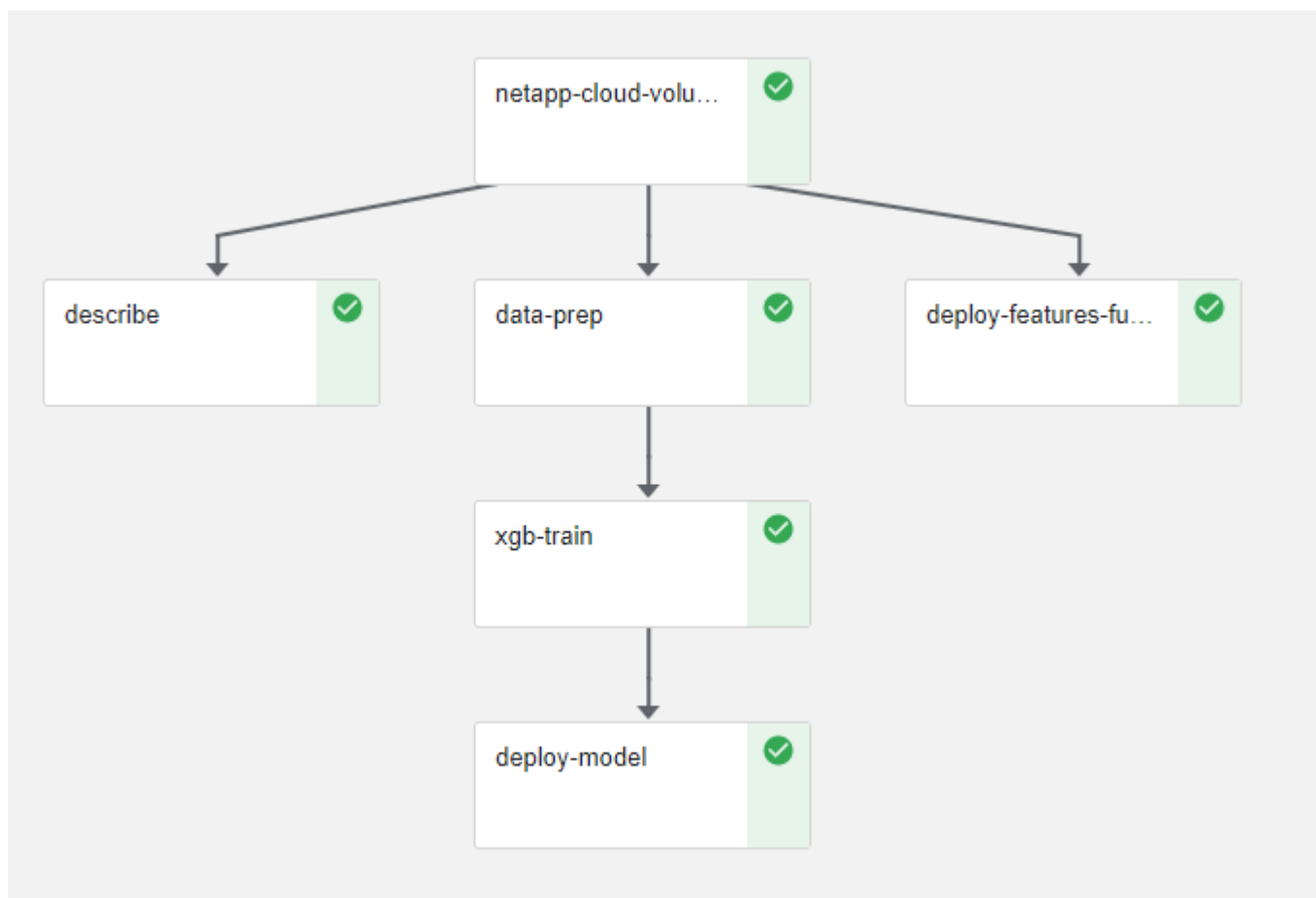
下表列出我們用來建置此工作的程式庫和架構。所有這些元件都已與Iguazio的角色型存取與安全控制功能完全整合。

程式庫/架構	說明
MLRun	由Iguazio管理、可讓您組裝、執行及監控ML/AI傳輸途徑。

程式庫/架構	說明
Nuclio	與Iguazio整合的無伺服器功能架構。也可作為由Iguazio管理的開放原始碼專案。
Kubeflow	以Kubernetes為基礎的架構、用於部署管線。這也是Iguazio所貢獻的開放原始碼專案。它與Iguazio整合、可提升安全性、並與其他基礎架構整合。
Docker	Docker登錄是以服務形式在Iguazio平台上執行。您也可以變更此設定以連線至登錄。
NetApp Cloud Volumes	在AWS上執行的Cloud Volumes可讓我們存取大量資料、並可將Snapshot複本複製到訓練所用的資料集版本。
Trident	Trident是由NetApp管理的開放原始碼專案。它有助於整合Kubernetes的儲存與運算資源。

我們使用數個筆記型電腦來建構ML管線。每一部筆記型電腦都能在整合到銷售管道之前進行個別測試。我們會依照此示範應用程式的部署流程、分別說明每一部筆記型電腦。

所需的結果是一條管道、會根據資料的Snapshot複本來訓練模型、並部署模型以供參考。下圖顯示完整MLRun管線的區塊圖。



部署資料產生功能

本節說明我們如何使用Nuclio無伺服器功能來產生網路裝置資料。使用案例可從部署管線的Iguazio用戶端進行調

整、並使用Iguazio服務來監控和預測網路裝置故障。

我們模擬來自網路裝置的資料。執行Jupyter筆記型電腦「data-gener.ipynb」可建立無伺服器功能、每10分鐘執行一次、並產生含有新資料的Parquet檔案。若要部署此功能、請執行此筆記型電腦中的所有儲存格。請參閱["Nuclio網站"](#) 檢閱此筆記型電腦中任何不熟悉的元件。

產生函數時會忽略具有下列註解的儲存格。筆記型電腦中的每個儲存格都會被視為功能的一部分。匯入Nuclio模組以啟用「%nuclio fic」。

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

在該函數的規格中、我們定義了執行該函數的環境、觸發方式、以及它所耗用的資源。

```
spec = nuclio.ConfigSpec(config={"spec.triggers.inference.kind":"cron",
                                "spec.triggers.inference.attributes.interval" : "10m",
                                "spec.readinessTimeoutSeconds" : 60,
                                "spec.minReplicas" : 1},.....
```

初始化功能時、Nuclio架構會叫用「init_context」功能。

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

當函數初始化時、會叫用任何不在函數中的程式碼。當您叫用它時、會執行處理常式功能。您可以變更處理常式的名稱、並在函數規格中加以指定。

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

您可以在部署之前、先從筆記型電腦測試功能。

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

此功能可從筆記型電腦部署、也可從CI/CD管道部署（修改此程式碼）。

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

管道筆記型電腦

這些筆記型電腦不應個別執行此設定。這只是對每個筆記型電腦的審查。我們將它們視為管道的一部分。若要個別執行、請檢閱MLRun文件、以Kubernetes工作的形式執行。

Snap_CV.ipynb

此筆記型電腦會在管線開始時處理Cloud Volume Snapshot複本。它會將磁碟區名稱傳遞給管線內容。此筆記型電腦會叫用Shell指令碼來處理Snapshot複本。在管線中執行時、執行內容會包含變數、以協助找出執行所需的所有檔案。撰寫此程式碼時、開發人員不必擔心執行程式碼的容器中的檔案位置。如稍後所述、此應用程式會隨其所有相依性一起部署、而且是提供執行內容的管線參數定義。

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

建立的Snapshot複本位置會放置在MLRun內容中、供管線中的步驟使用。

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

接下來的三部筆記型電腦會平行執行。

資料準備：ipynb

原始指標必須轉變為功能、才能進行模型訓練。此筆記型電腦會從Snapshot目錄讀取原始指標、並將模型訓練功能寫入NetApp Volume。

在管線內容中執行時、輸入「DAAT_DIR」會包含Snapshot複本位置。

```
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')))
```

描述.ipynb

為了視覺化傳入的度量、我們部署了一個管線步驟、提供可透過Kubeflow和MLRun UI取得的繪圖和圖表。每次執行都有其專屬版本的視覺化工具。

```
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功能.ipynb

我們持續監控指標、以尋找異常狀況。這款筆記型電腦會建立一個無伺服器功能、產生在傳入度量上執行預測所需的功能。此筆記型電腦會啟動功能的建立。功能代碼位於筆記型電腦「data- prep.ipynb」中。請注意、我們使用同一部筆記型電腦做為此目的的管道步驟。

訓練.ipynb

建立這些功能之後、我們便開始進行模型訓練。此步驟的輸出是用於推斷的模型。我們也會收集統計資料、以追蹤每次執行（實驗）。

例如、下列命令會在該實驗的內容中輸入準確度分數。此值可在Kubeflow和MLRun中看到。

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

deploy推論函數.ipynb

管道的最後一步是將模型部署為無伺服器功能、以便持續推斷。此筆記型電腦會啟動建立在「nuclio-inertere-fuite.ipynb」中定義的無伺服器功能。

審查及建置管道

在管線中執行所有的筆記型電腦、可持續執行實驗、根據新的指標來重新評估模型的準確度。首先、開啟「pipe.ipynb」筆記型電腦。我們將帶您詳細瞭解NetApp與Iguazio如何簡化這項ML管線的部署。

我們使用MLRun為管線的每個步驟提供背景資料並處理資源分配。MLRun API服務在Iguazio平台上執行、是與Kubernetes資源互動的點。每個開發人員都無法直接要求資源；API會處理要求並啟用存取控制。

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

該管道可與NetApp Cloud Volumes和內部部署Volume搭配使用。我們打造此示範影片來使用Cloud Volumes、但您可以在程式碼中看到可在內部部署執行的選項。

```
# Initialize the NetApp snap function 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.ipyn
b").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]
```

將Jupyter筆記型電腦轉變成Kubeflow步驟所需的第一個行動、就是將程式碼變成功能。某項功能具備執行該筆記型電腦所需的所有規格。當您向下捲動筆記本時、您會看到我們為管道中的每個步驟定義了功能。

筆記型電腦的一部分	說明
<code_to功能> (MLRun模組的一部分)	功能名稱：專案名稱。用於組織所有專案成品。這可在MLRun UI中看到。種類。在此案例中、Kubernetes工作。這可能是dask、MPI、走勢8等等。如需詳細資訊、請參閱MLRun文件。檔案：筆記型電腦的名稱。這也可以是Git (HTTP) 中的位置。
映像	我們在這個步驟中使用的Docker映像檔名稱。我們先前使用cree-image.ipynb系列筆記型電腦來建立這個應用程式。
Volume_掛 載與磁碟區	執行時掛載NetApp Cloud Volume的詳細資料。

我們也定義步驟的參數。

```

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
}

```

為所有步驟定義功能之後、您就可以建構管線。我們使用「kfp」模組來定義這個定義。使用MLRun與自行建置的差異在於編碼的簡化與縮短。

我們定義的功能會使用MLRun的「AS步驟」功能、變成步驟元件。

Snapshot步驟定義

啟動Snapshot功能、輸出及掛載v3io作為來源：

```

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

```

參數	詳細資料
新工作	newtask是函數執行的定義。
(MLRun模組)	處理常式：要叫用的Python函數名稱。我們在筆記型電腦中使用名稱處理常式、但這不是必要的。參數。傳遞給執行的參數。在程式碼中、我們使用context.Get_param（「參數」）來取得值。
AS步驟	名稱。Kubeflow管道步驟名稱。輸出。這些是步驟在完成時新增至字典的值。請參閱Snap_CV.ipynb系列筆記型電腦。mount_v3io()。這會設定執行管線之使用者要掛載/User的步驟。

```

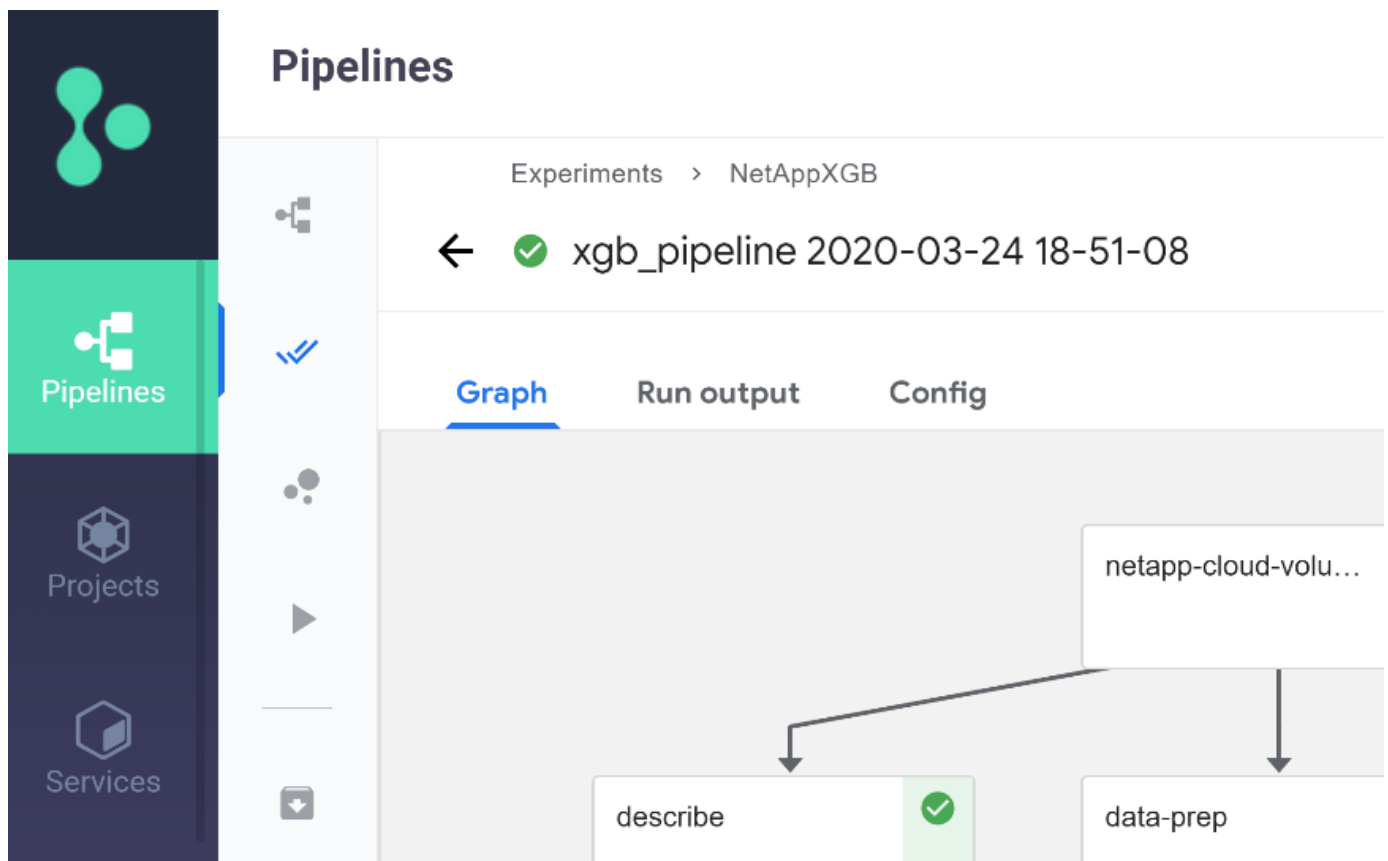
prep = data_prep.as_step(name='data-prep',
handler='handler',params=params,
                        inputs = {'DATA_DIR':
snap.outputs['snapVolumeDetails']} ,

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

```

參數	詳細資料
輸入	您可以將前一個步驟的輸出傳送到一個步驟。在這種情況下、snap.outputs [snapVolume Details]是我們 在Snapshot步驟上建立的Snapshot複本名稱。
Out_path	放置使用MLRun模組log_act件 產生成品的位置。

您可以從上到下執行「pipele.ipynb」。然後、您可以從Iguazio儀表板前往「Pipelines」（管路）索引標籤、以監控Iguazio儀表板「Pipelines」（管路）索引標籤中顯示的進度。



由於我們在每次路跑中都記錄訓練步驟的準確度、因此每次實驗都有精準度的記錄、如訓練準確度記錄所示。

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

如果您選取Snapshot步驟、就會看到用於執行此實驗的Snapshot複本名稱。

×
netops-trainign-pipeline-with-netapp-volume-cloning-rtxdl-2910983943

Artifacts
Input/Output
Volumes
Manifest
Logs

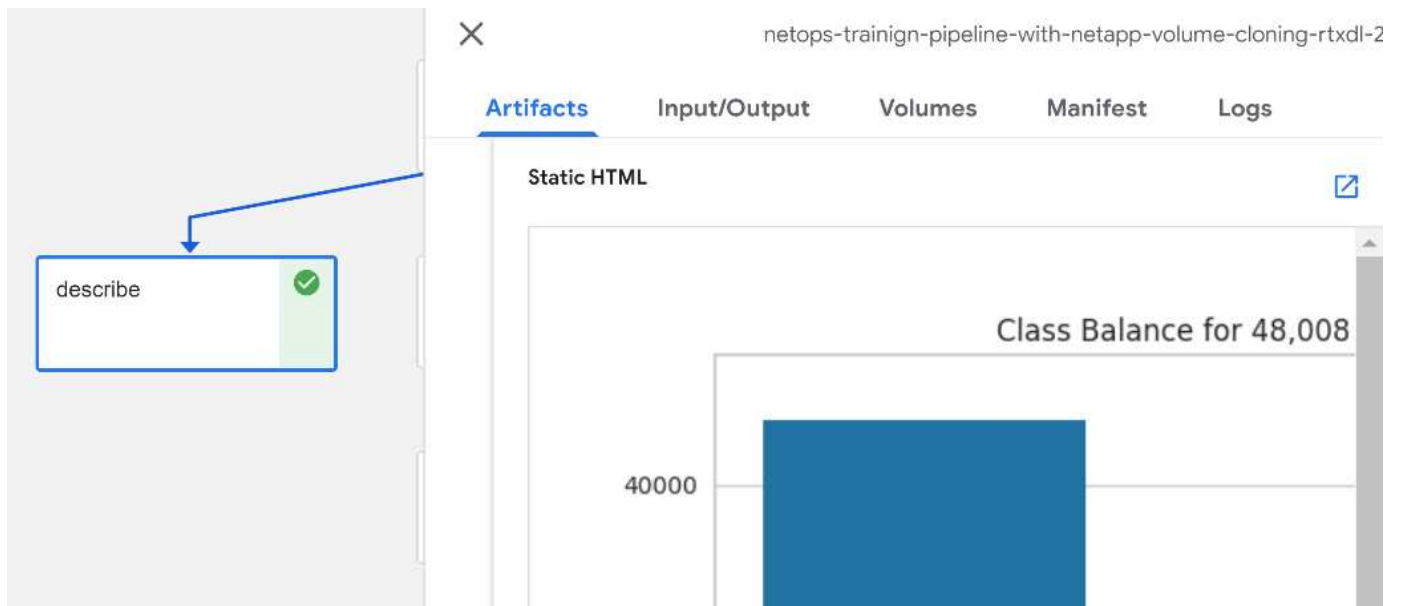
input artifacts

Output parameters

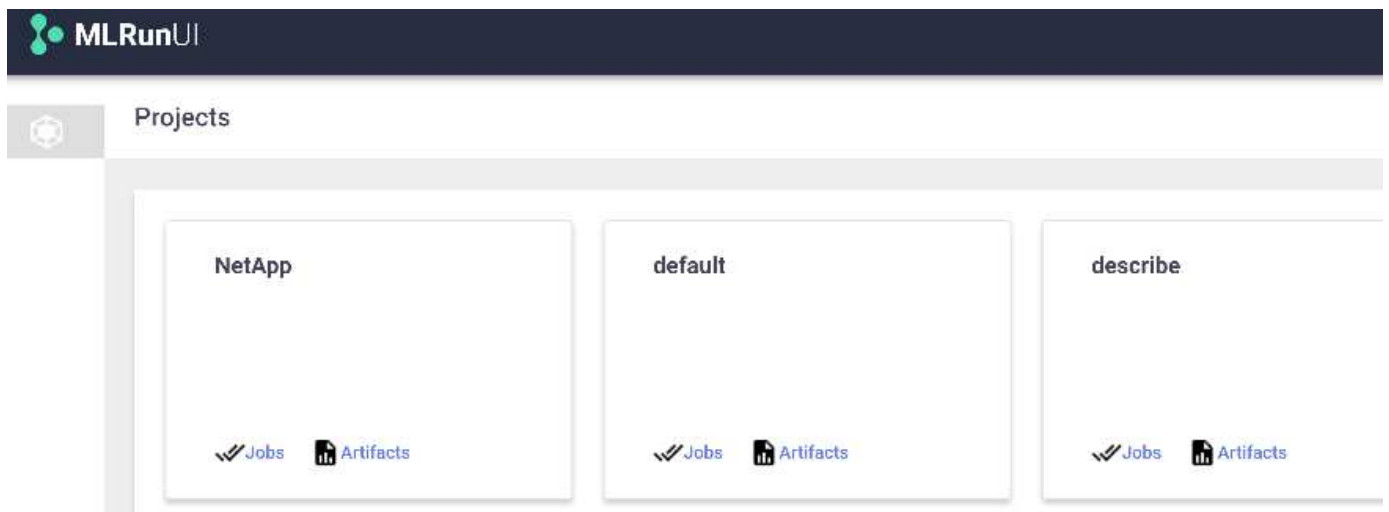
netapp-cloud-volume-snapshot-snapVolumeDetails	/netapp/.snapshot/kfp_20200324_185122
netapp-cloud-volume-snapshot-training_parquet_file	/netapp/.snapshot/kfp_20200324_18512...

Output artifacts

所述步驟具有視覺成品、可用來探索我們使用的指標。您可以展開以檢視完整繪圖、如下圖所示。



MLRun API資料庫也會追蹤各專案所組織之每個執行的輸入、輸出和成品。下列影像提供每個掃描的輸入、輸出和成品範例。



我們會針對每項工作儲存其他詳細資料。

Name

deploy-model

24 Mar, 14:56:03 ...bcbe38e

xgb_train

24 Mar, 14:53:18 ...5c85949

data-prep

24 Mar, 14:52:46 ...126dc73

describe

24 Mar, 14:52:45 ...c2a460e

deploy-features-function

24 Mar, 14:52:43 ...50d8b83

NetApp_Cloud_Volume_Sna

24 Mar, 14:51:22 ...3108eb2

describe

24 Mar, 14:52:45

Info Inputs Artifacts Results Logs

UID

66ef22187efb4ad89e8da8433c2a460e

Start time

24 Mar, 14:52:45

Parameters

Completed

Results

class_label...

key: summary

label_colu...

關於MLRun的資訊比本文件所涵蓋的資訊更多。所有成品（包括步驟和功能的定義）都可儲存至API資料庫、版本控制、個別或完整的專案來叫用。專案也可儲存並推送至Git供日後使用。我們鼓勵您在上深入瞭解 "[MLRun GitHub網站](#)"。








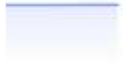











部署Grafana儀表板

部署所有項目之後、我們會針對新資料執行推斷。模型預測網路裝置設備故障。預測結果會儲存在Iguazio Timezio表格中。您可以在整合了Iguazio安全性與資料存取原則的平台上、以Grafana視覺化的方式呈現結果。

您可以將所提供的Json檔案匯入叢集中的Grafana介面、以部署儀表板。

- 若要驗證Grafana服務是否正在執行、請查看「服務」下的。

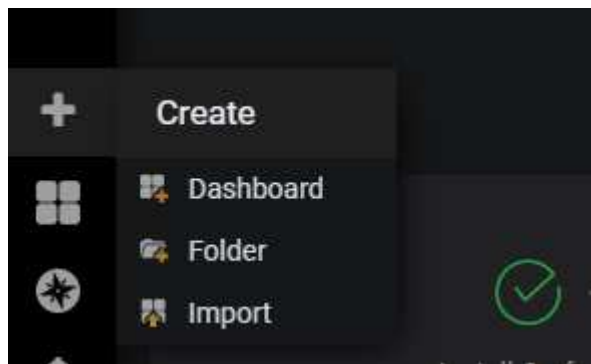
Services

<input type="checkbox"/>	Name ↑	Running User	Version ↕	CPU (cores)	Memory	AF
<input type="checkbox"/>	 docker-registry Type: Docker Regi		2.7.1	96μ 	1.67 GB 	H
<input type="checkbox"/>	 framesd Type: V3IO Frame		0.6.10	369μ 	795.19 MB 	H
<input type="checkbox"/>	 grafana Type: Grafana		6.6.0	1m 	38.39 MB 	
<input type="checkbox"/>	 jupyter Type: Jupyter Note	admin	1.0.2	81m 	3.27 GB 	
<input type="checkbox"/>	 log-forwarder Type: Log forward		6.7.2	0 	0 bytes 	

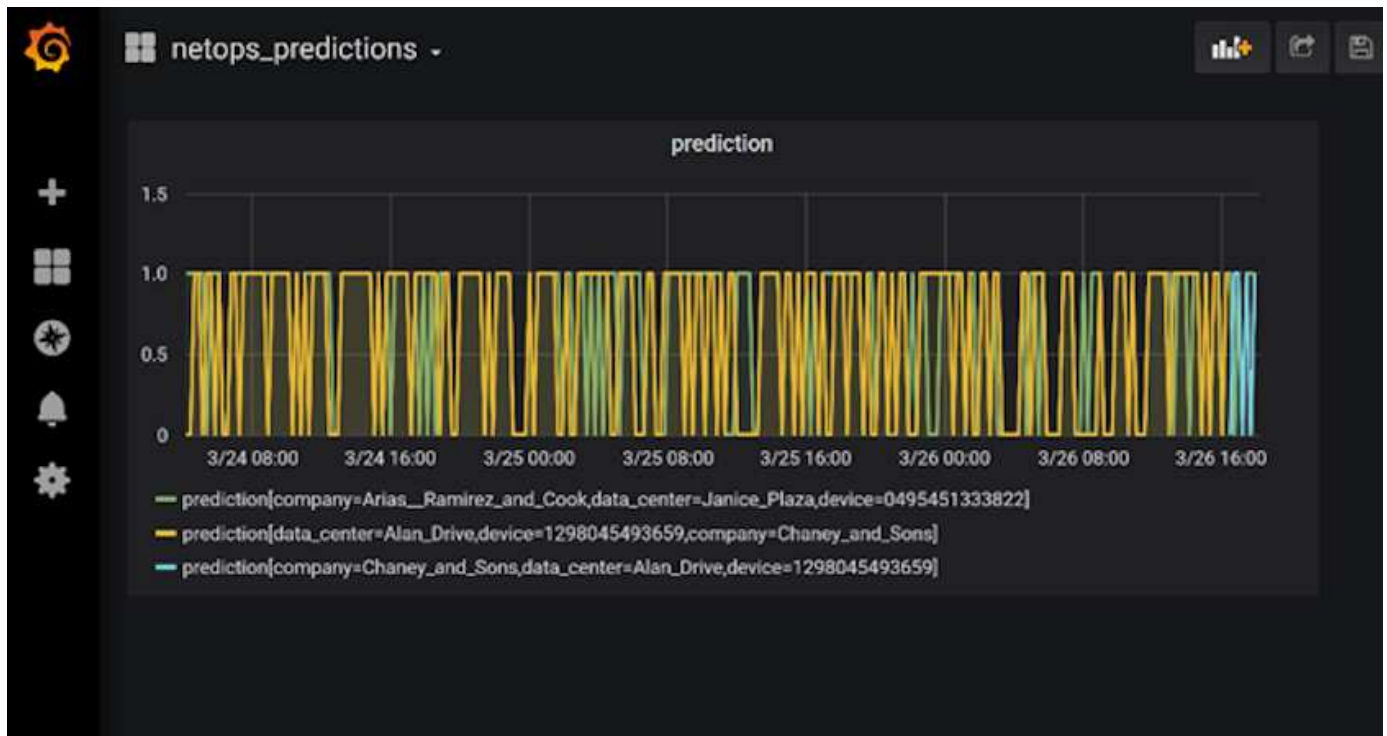
2. 如果不存在、請從「服務」區段部署執行個體：
 - a. 按一下「New Service（新服務）」
 - b. 從清單中選取Grafana。
 - c. 接受預設值。
 - d. 按「Next Step（下一步）」
 - e. 輸入您的使用者ID。
 - f. 按一下儲存服務。
 - g. 按一下頂端的套用變更。
3. 若要部署儀表板、請透過Jupyter介面下載「NetopsPredictions-Dashboard・json」檔案。



4. 從「服務」區段開啟Grafana、然後匯入儀表板。



5. 按一下「上傳」 「*.json」檔案、然後選取您先前下載的檔案（「NetopsPredictions-Dashboard.json」）。儀表板會在上傳完成後顯示。



部署清理功能

當您產生大量資料時、務必保持一切乾淨且有組織。若要這麼做、請使用「cleanup.ipynb」筆記型電腦來部署清理功能。

效益

NetApp和Iguazio藉由在Kubeflow、Apache Spark和TensorFlow等重要架構中建置、以及Docker和Kubernetes等協調工具、來加速和簡化AI和ML應用程式的部署。NetApp與Iguazio藉由統一化端點對端點資料傳輸途徑、有效縮短許多進階運算工作負載固有的延遲與複雜度、有效縮短開發與營運之間的落差。資料科學家可以在大型資料集上執行查詢、並在訓練階段與授權使用者安全地共用資料與演算法模型。在容器化模型準備就緒可供正式作業之後、您可以輕鬆地將其從開發環境移至作業環境。

結論

在建置自己的AI / ML管道時、設定架構中元件的整合、管理、安全性及存取能力、是一項艱鉅的任務。讓開發人員存取及控制其環境是另一組挑戰。

NetApp與Iguazio的結合、將這些技術整合為託管服務、可加速技術採用、並縮短新的AI/ML應用程式上市時間。

TR-4915：E系列和BeeGFS的資料搬移功能、適用於AI和分析工作流程

CODY Harrman和Ryan Rovine、NetApp

TR-4915說明如何將資料從任何資料儲存庫移至以NetApp E系列SAN儲存設備為後盾的BeeGFS檔案系統。對於人工智慧（AI）和機器學習（ML）應用程式、客戶可能經常需要將超過數PB資料的大型資料集移至BeeGFS叢集、以便進行模型開發。本文件探討如何

使用 NetApp XCP 和 NetApp BlueXP 複製與同步工具來達成此目標。

["TR-4915：E系列和BeeGFS的資料搬移功能、適用於AI和分析工作流程"](#)

版權資訊

Copyright © 2024 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

商標資訊

NETAPP、NETAPP 標誌及 <http://www.netapp.com/TM> 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。