



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

NetApp  
April 12, 2024

# 目錄

適用於 MLOps 的 AWS FSx for NetApp ONTAP (FSxN) .....	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 .....	1
第 2 部分：運用 AWS FSx for NetApp ONTAP (FSxN) 做為 SageMaker 模型訓練的資料來源 .....	15
第 3 部分：建立簡化的 MLOps 管道 (CI/CT/CD) .....	24

# 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。

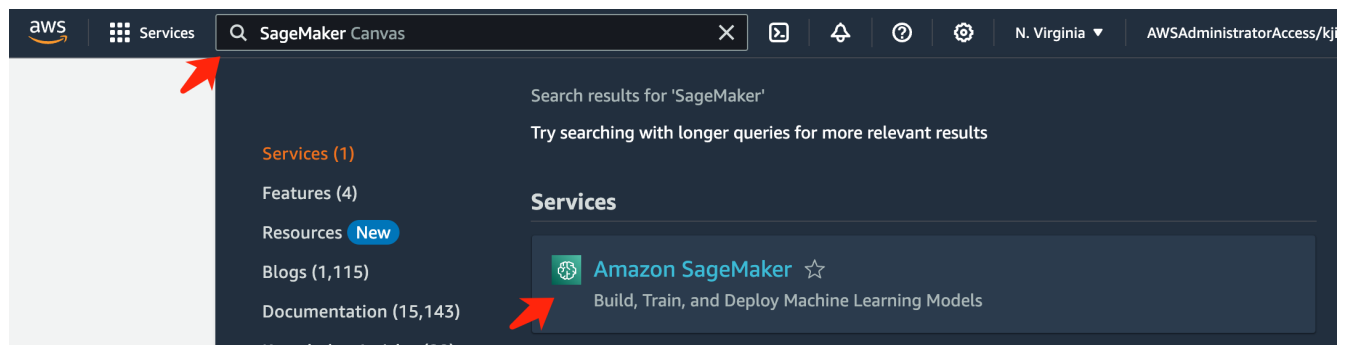
如需 FSxN 的詳細資訊、請參閱本簡報 (["視訊連結"](#))

### 使用者指南

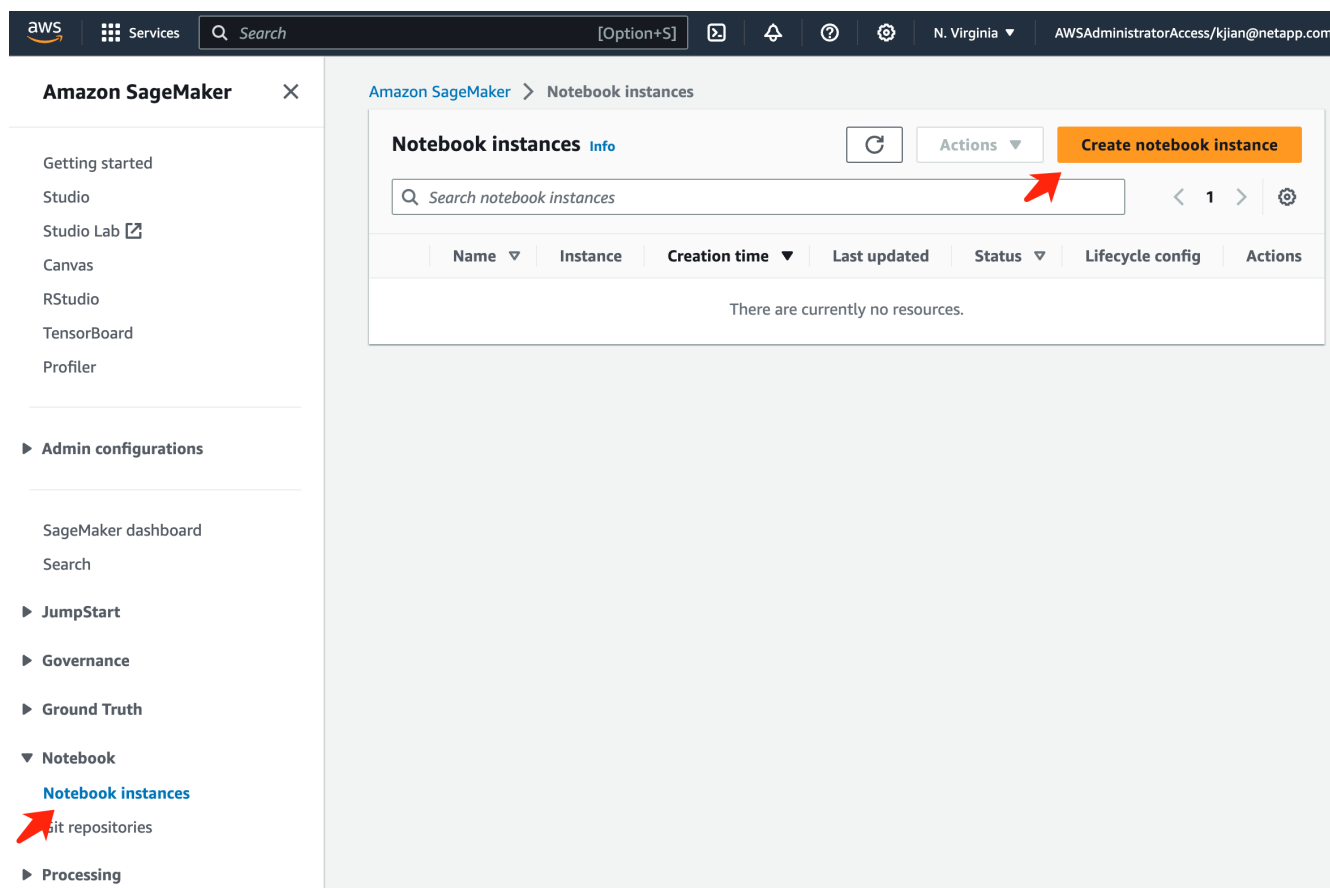
#### 伺服器建立

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

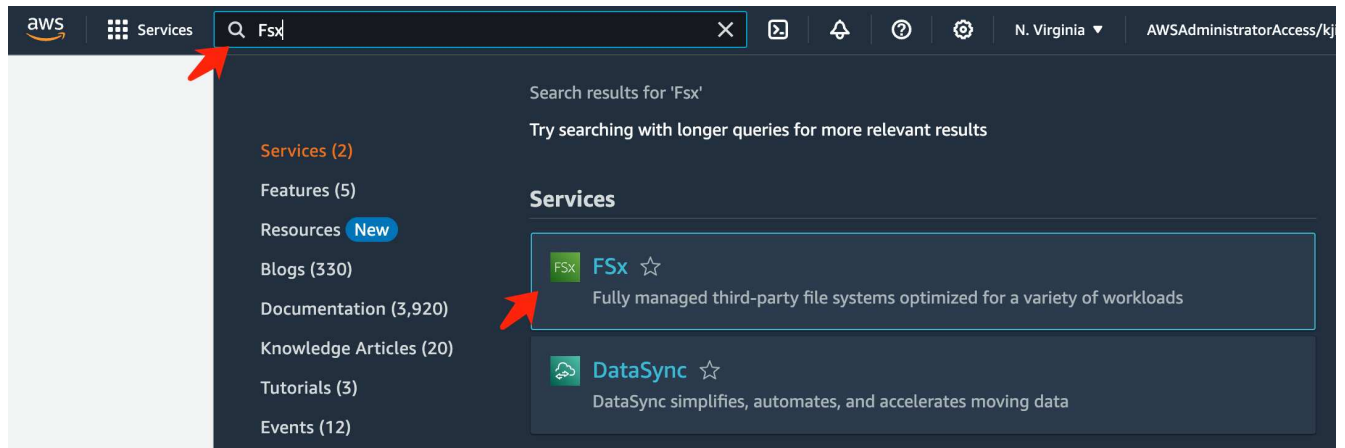
► Tags - optional

Cancel

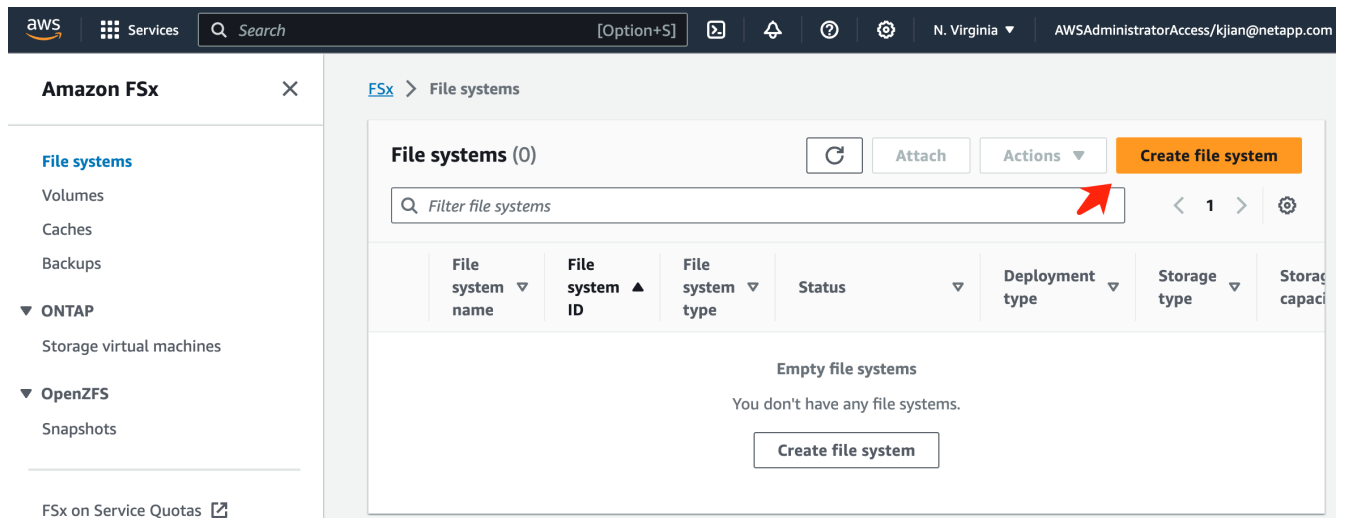
Create notebook instance

## 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]

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]

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

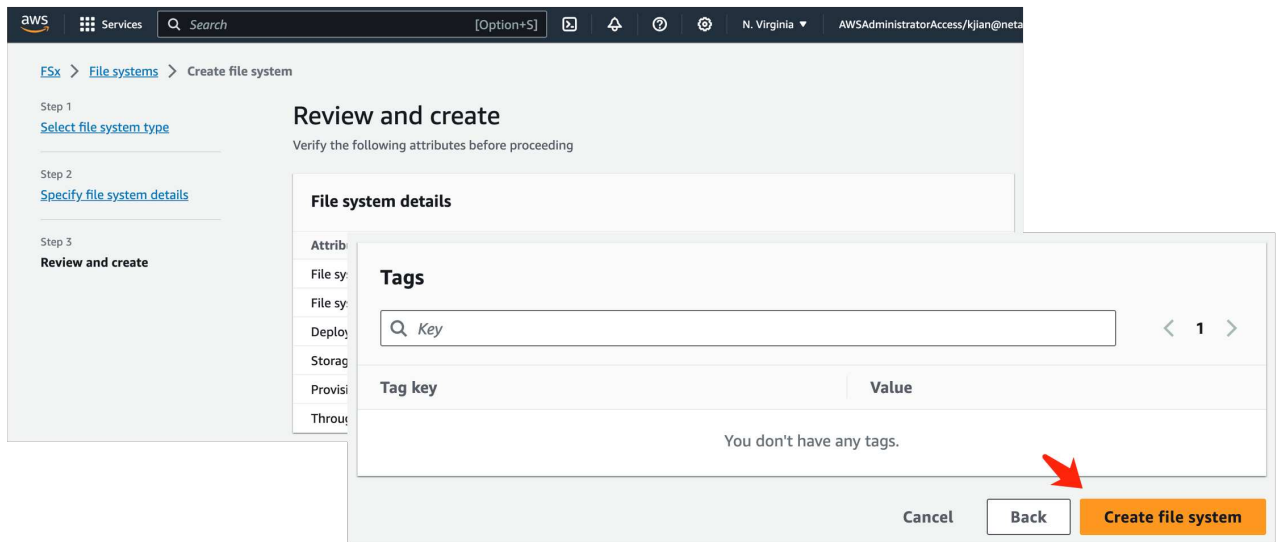
Cancel

Back

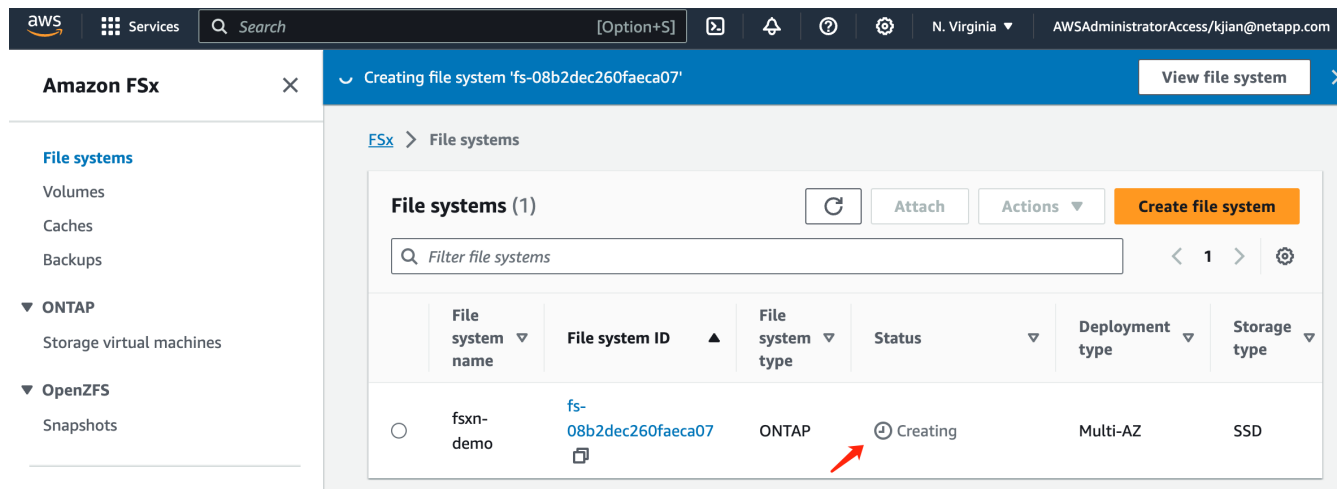
Next

f. 按一下檢閱頁面右下角的橘色按鈕 \* 建立檔案系統 \* 。

8



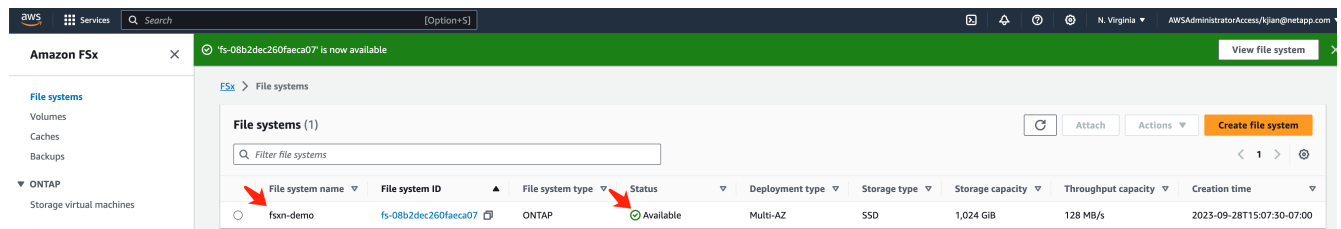
5. 增加 FSX 檔案系統可能需要大約 \*20-40 分鐘\* 。



## 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** ×

File systems  
Volumes  
Caches  
Backups

▼ **ONTAP**  
Storage virtual machines

▼ **OpenZFS**  
Snapshots

FSx on Service Quotas [🔗](#)

[FSx](#) > [File systems](#) > fs-08b2dec260faeca07

## fsxn-demo (fs-08b2dec260faeca07) Attach Actions ▼

▼ **Summary**

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

< [Network & security](#) | [Monitoring & performance](#) | **[Administration](#)** | [Storage virtual machines](#) >

### ONTAP administration

Management endpoint - DNS name management.fs-08b2dec260faeca07.fsx.us-east-1.amazonaws.com <a href="#">🔗</a>	Management endpoint - IP address 172.31.255.250 <a href="#">🔗</a>	ONTAP administrator username fsxadmin <a href="#">🔗</a>
Inter-cluster endpoint - DNS name intercluster.fs-08b2dec260faeca07.fsx.us-east-1.amazonaws.com <a href="#">🔗</a>	Inter-cluster endpoint - IP address 172.31.31.157 <a href="#">🔗</a> 172.31.32.38 <a href="#">🔗</a>	ONTAP administrator password <span>Update</span>

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

**Amazon SageMaker** ×

Getting started  
Studio  
Studio Lab [🔗](#)  
Canvas  
RStudio  
TensorBoard

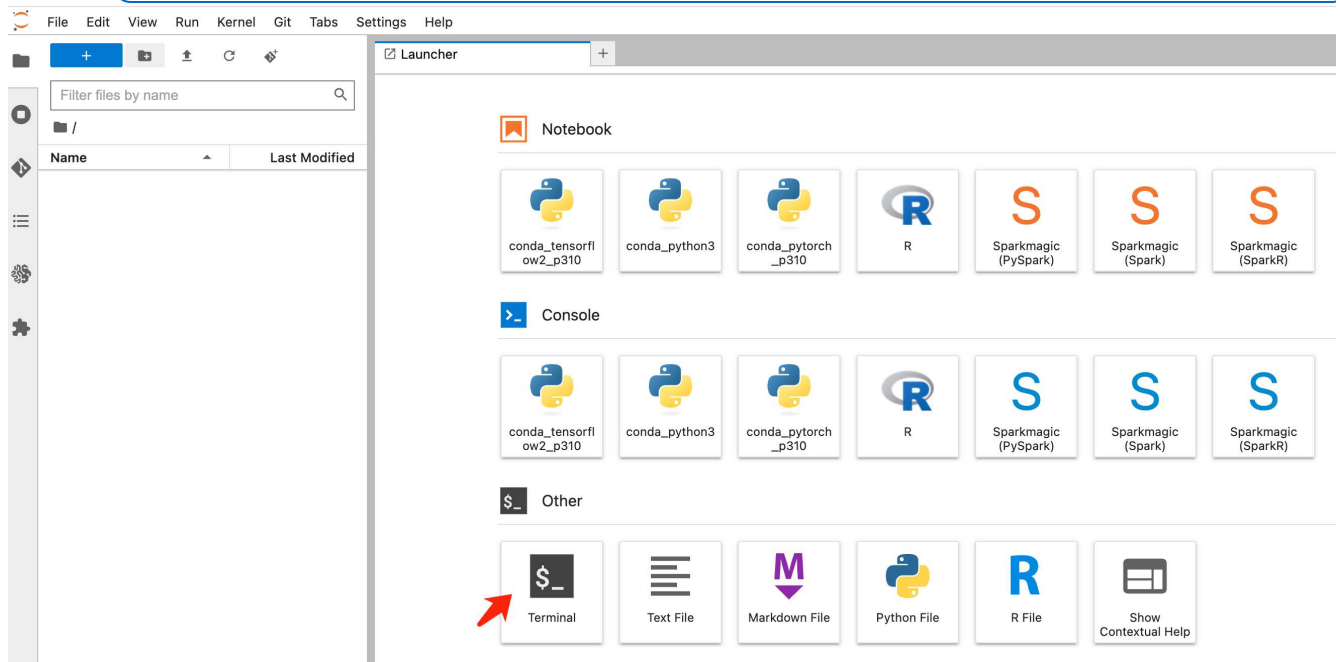
Amazon SageMaker > Notebook instances

### Notebook instances [Info](#)

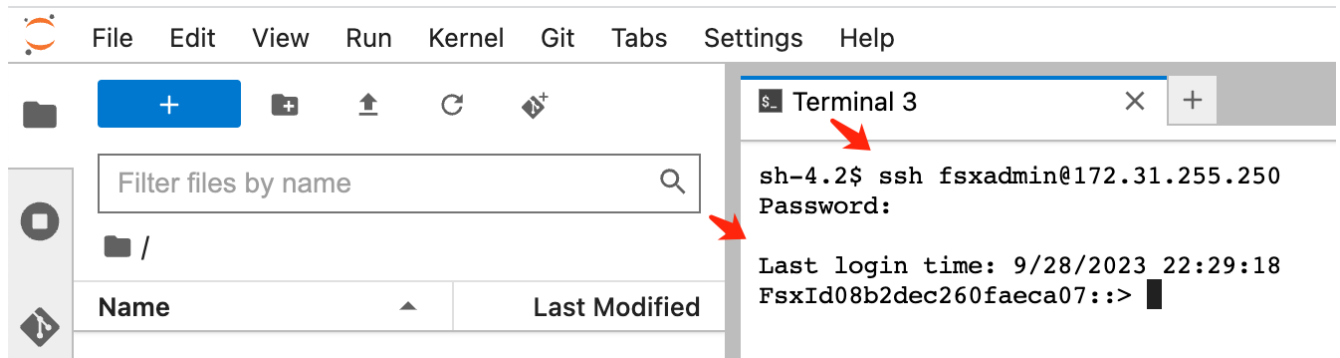
🔄 Actions ▼ Create notebook instance

<input type="radio"/>	Name ▼	Instance	Creation time ▼	Last updated	Status ▼	Lifecycle config	Actions
<input type="radio"/>	fsxn-demo	ml.t3.medium	9/28/2023, 1:47:27 PM	9/28/2023, 1:50:28 PM	InService		<a href="#">Open Jupyter</a>   <a href="#">Open JupyterLab</a>

4. 在 Jupyter Lab 頁面中，打開一個新的 \* 終端 \* 。



- 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。



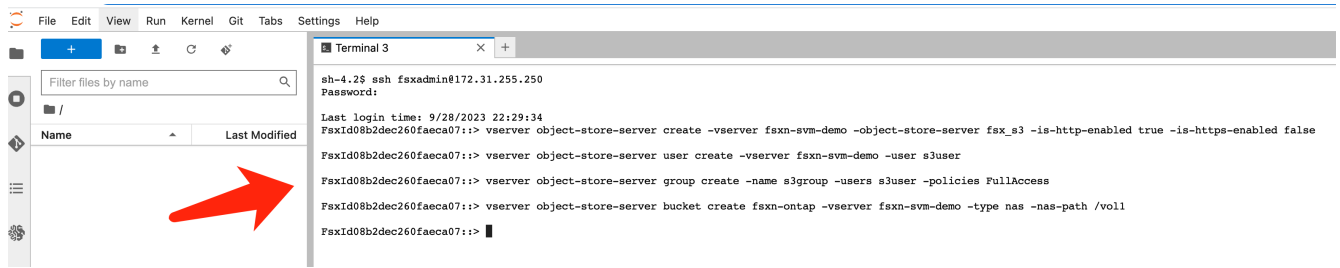
- 依照下列順序執行命令。  
我們使用 **fsxn-ONTAP** 作為 \*FSxN Private S3 儲存區名稱\* 的名稱。  
請將 \* 儲存虛擬機器名稱 \* 用於 \* 虛擬伺服器 \* 引數。

```
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。

```

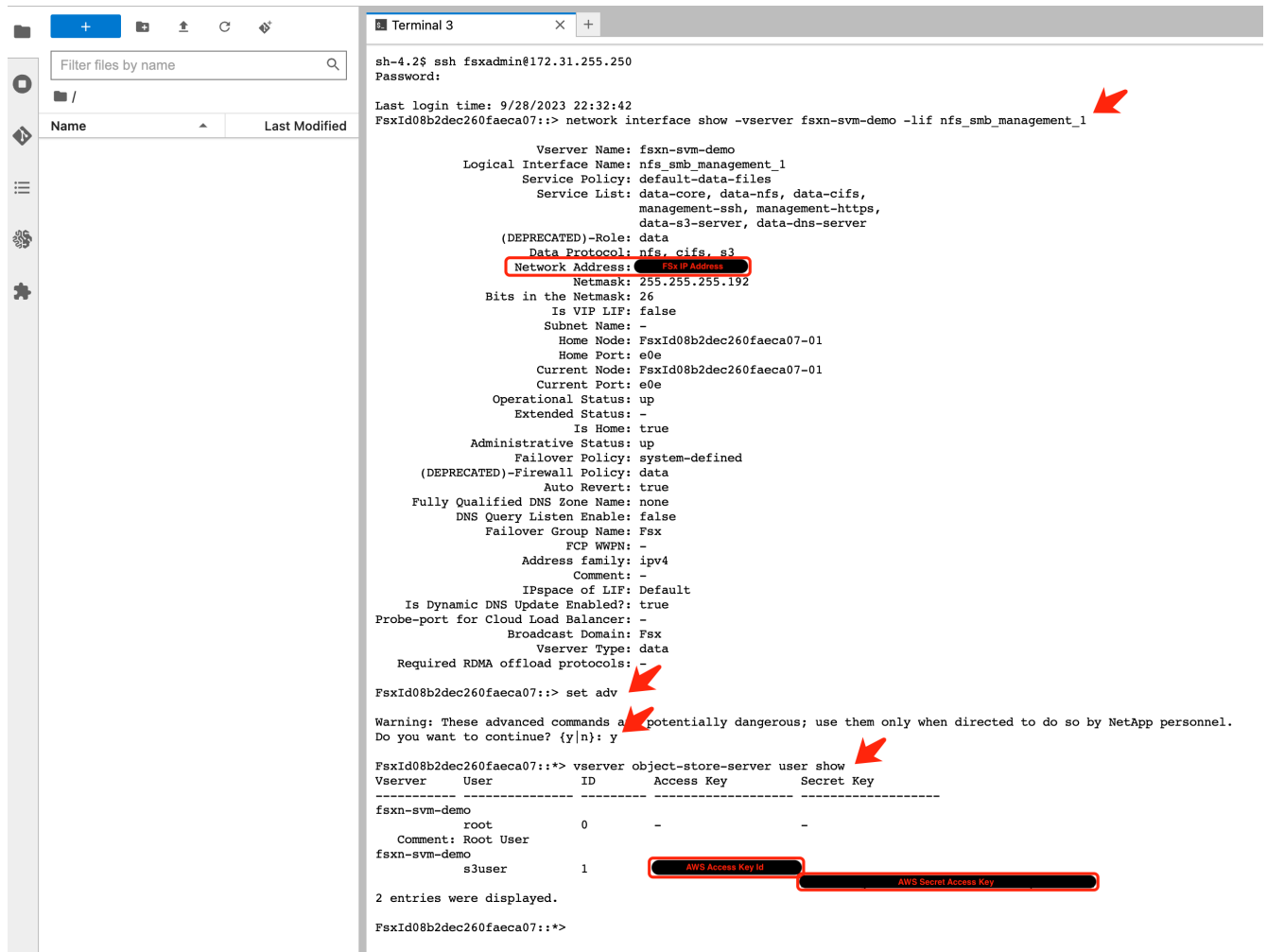
network interface show -vserver fsxn-svm-demo -lif nfs_smb_management_1

set adv

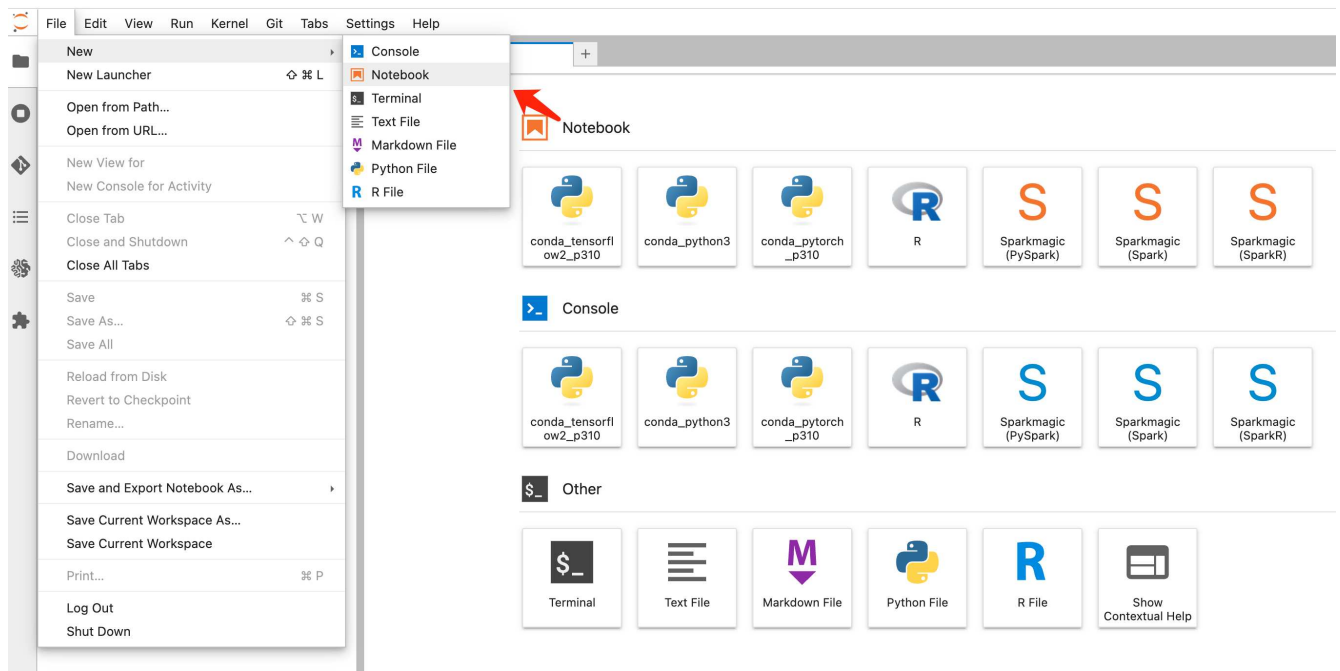
vserver object-store-server user show

```

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 Impleplate\*」（\* 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

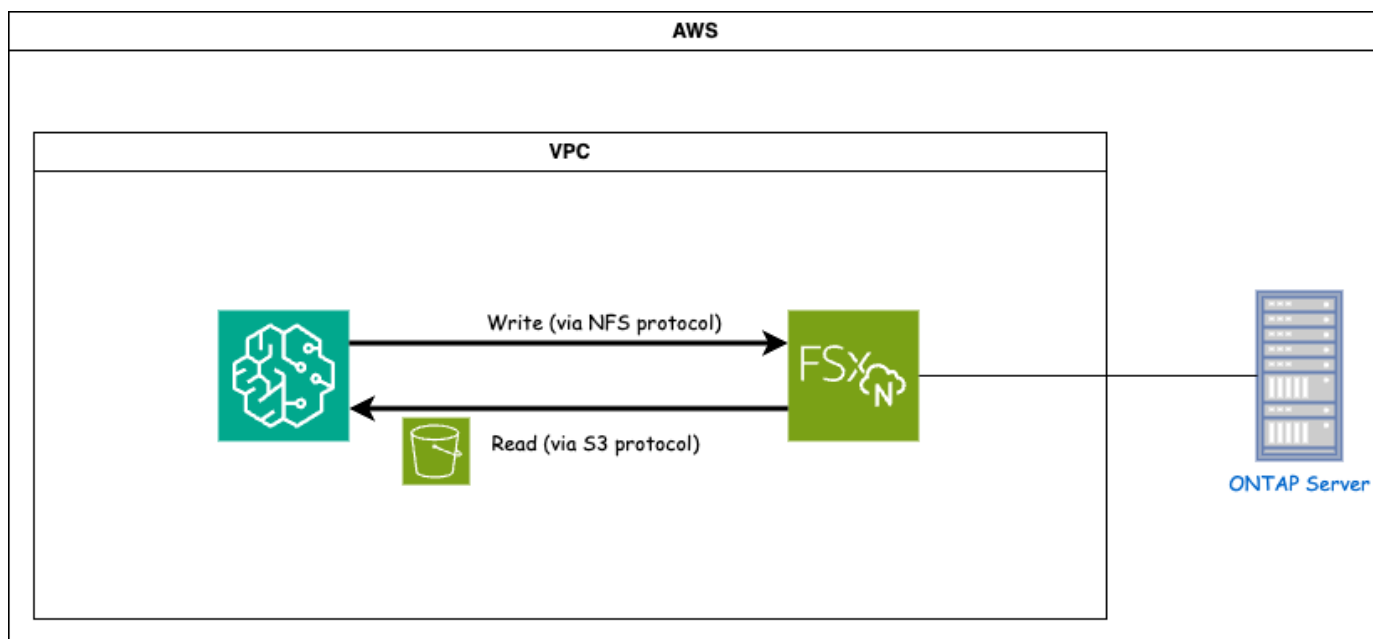
本教學課程提供電腦視覺分類專案的實用範例、提供建置 ML 模型的實際經驗、這些模型將 FSxN 作為 SageMaker 環境中的資料來源。此專案著重於使用深層學習架構 PyTorch、根據輪胎影像來分類輪胎品質。它強調使用 FSxN 做為 Amazon SageMaker 資料來源的機器學習模型開發。

### 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



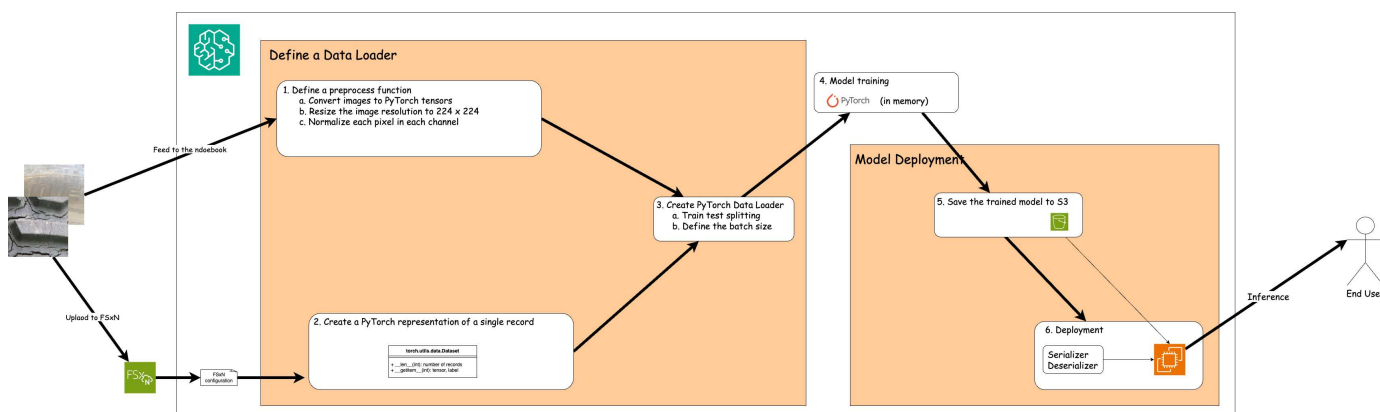
FSxN（Amazon FSx for NetApp ONTAP）是 AWS 儲存服務。其中包括在 NetApp ONTAP 系統上執行的檔案系統、以及連接到它的 AWS 託管系統虛擬機器（SVM）。在所提供的圖表中、AWS 管理的 NetApp ONTAP 伺服器位於 VPC 之外。SVM 是 SageMaker 和 NetApp ONTAP 系統之間的中介、可接收來自 SageMaker 的作業要求、並將其轉送至基礎儲存設備。若要存取 FSxN、SageMaker 必須與 FSxN 部署位於同一個 VPC 內。此組態可確保 SageMaker 和 FSxN 之間的通訊和資料存取。

## 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 24 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。

```

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

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。

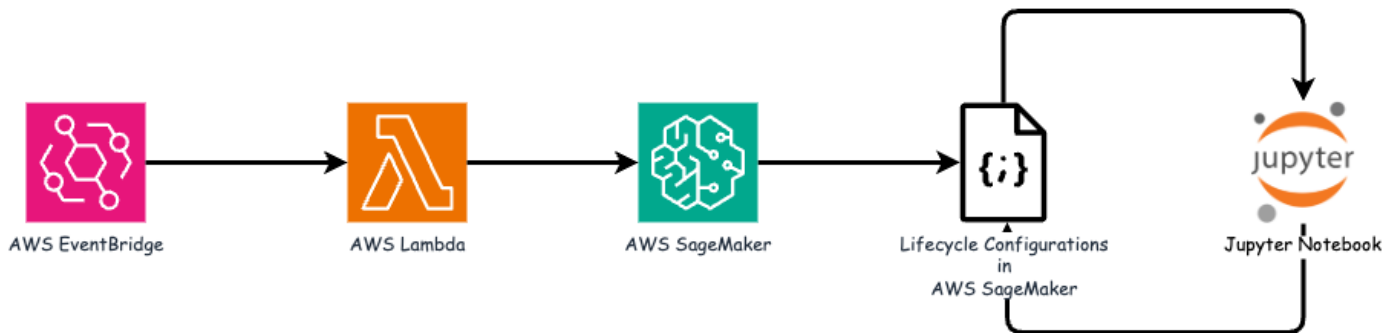
### 資訊清單

功能	名稱	留言
資料儲存	AWS FSxN	請參閱 <a href="#">"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"</a> 。
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 <a href="#">"第 2 部分：運用 AWS FSx for NetApp ONTAP（FSxN）做為 SageMaker 模型訓練的資料來源"</a> 。
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

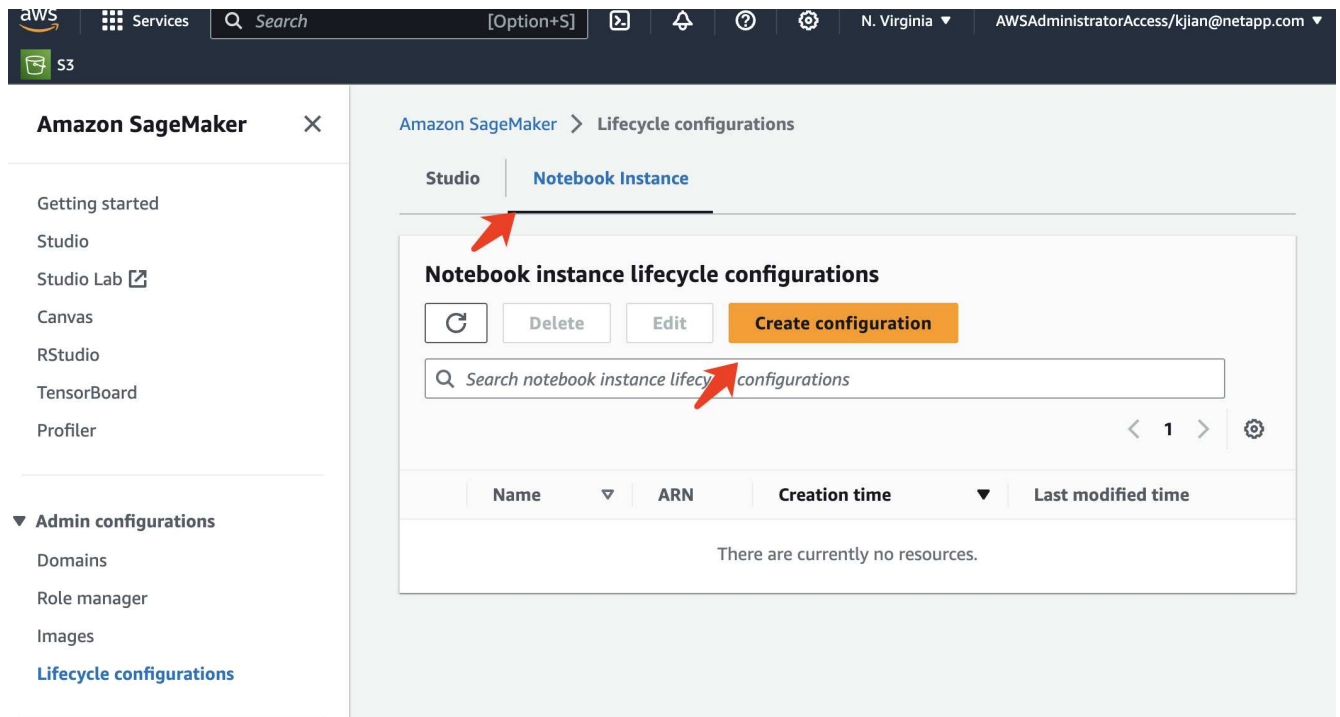
Domains (4)

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

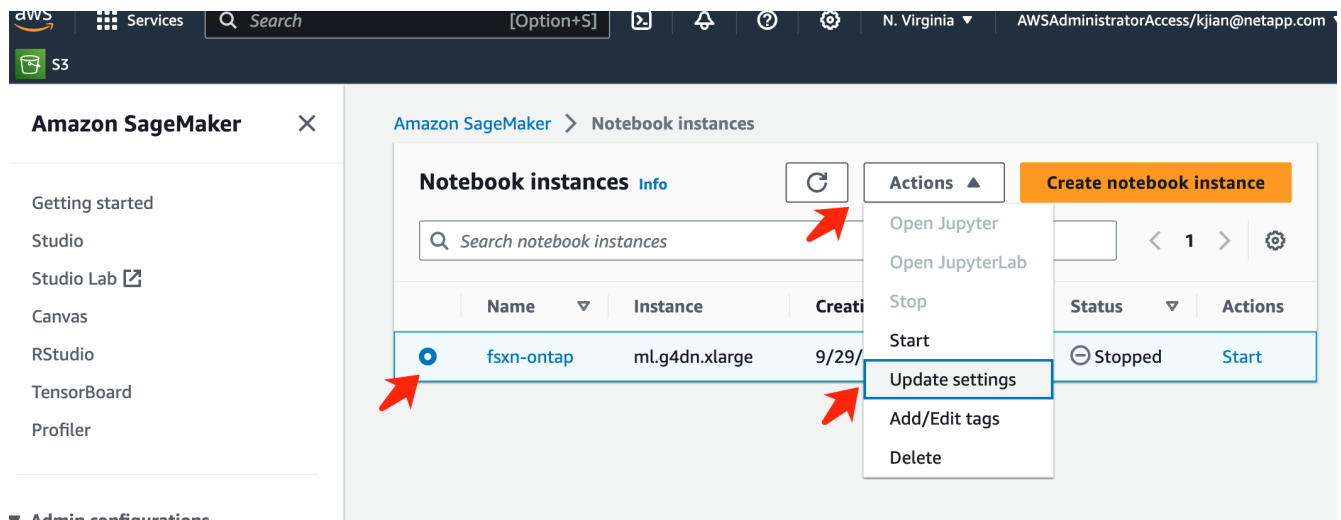
This script will be run each time an associated notebook instance is started, including during initial creation. If the associated notebook instance is already started, it will be run the next time it is stopped and started. [a curated list of sample scripts](#)

```
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
```

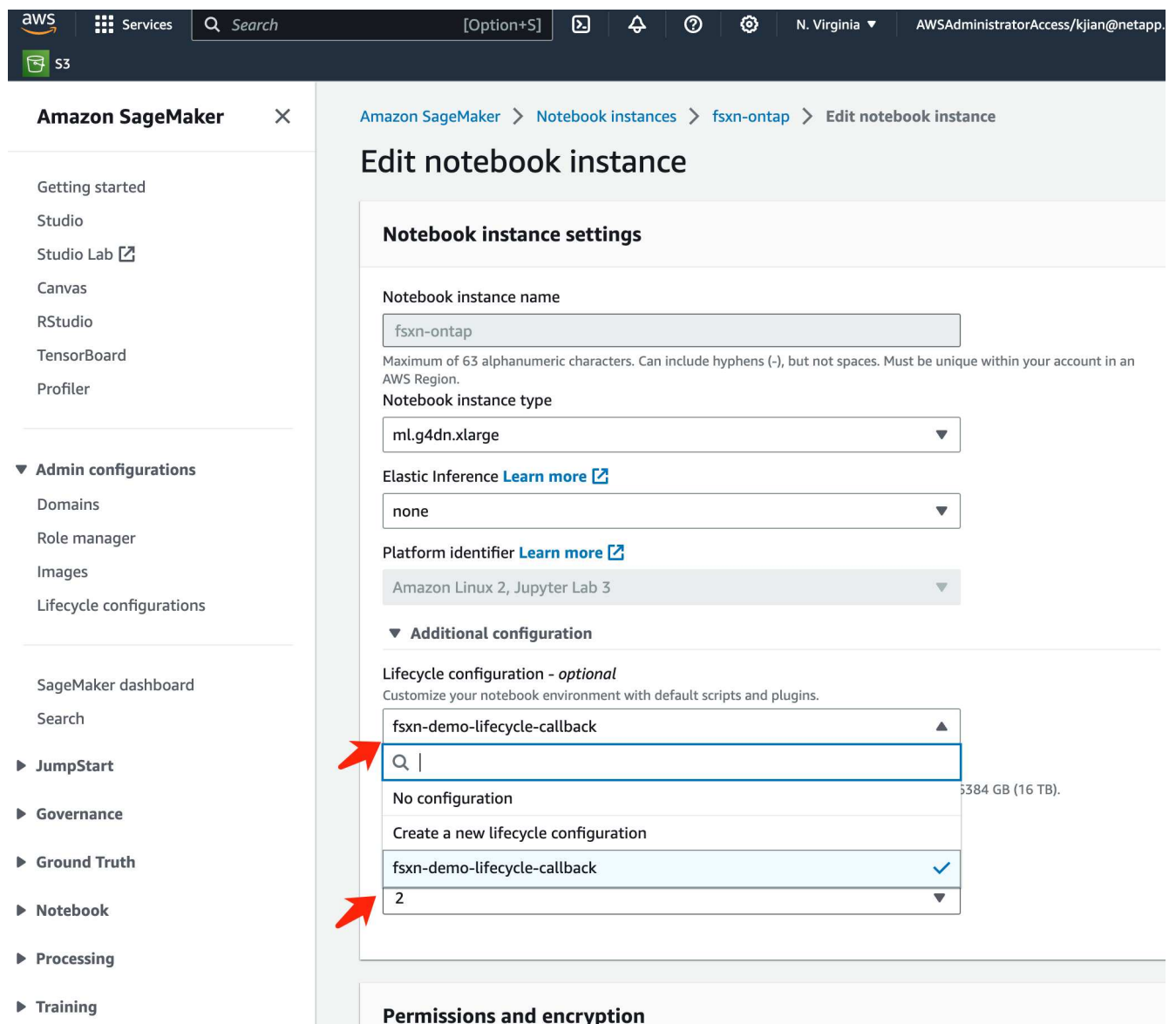
Cancel Create configuration

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 \*」。



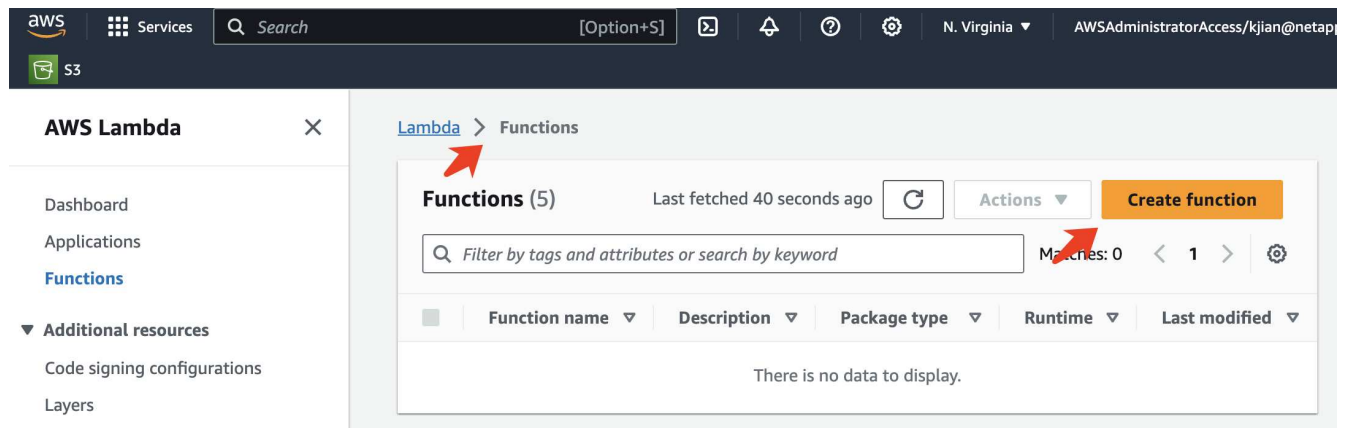
6. 選取建立的 \* 生命週期組態 \* 、然後按一下 \* 更新筆記型電腦執行個體 \* 。



## 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.

fsxn-demo-mlops

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.

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**

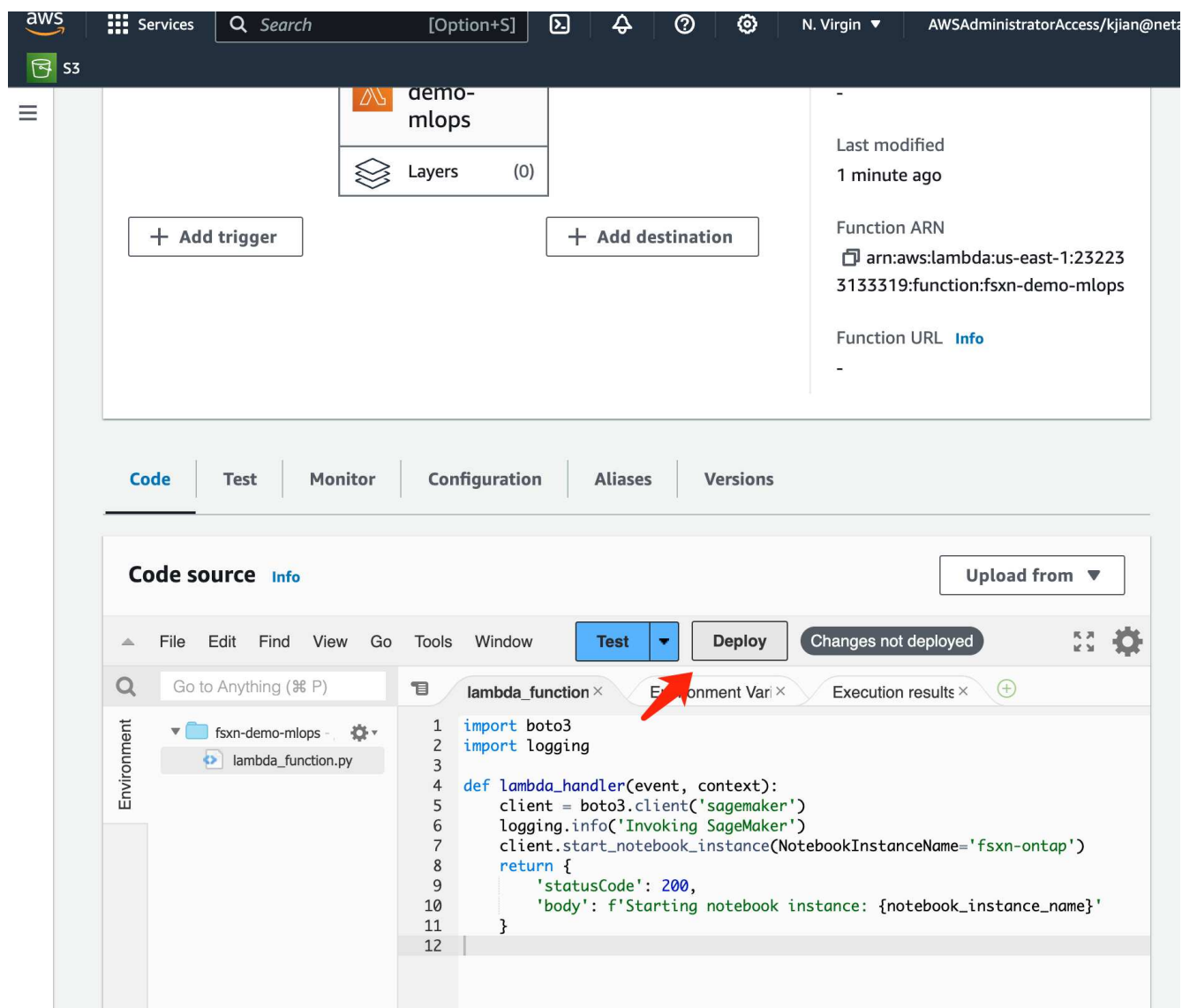
Cancel Create function

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. At the top, the navigation bar includes the AWS logo, 'Services', a search bar, and the region 'N. Virginia'. The breadcrumb trail indicates the path: [Lambda](#) > [Functions](#) > fsxn-demo-mlops. The function name 'fsxn-demo-mlops' is prominently displayed. To the right of the name are buttons for 'Throttle', 'Copy ARN', and 'Actions'. Below the function name, the 'Function overview' section is expanded, showing a summary of the function's configuration. This includes the function icon, name, and a 'Layers' section indicating '(0)' layers. Two buttons, '+ Add trigger' and '+ Add destination', are visible. A red arrow points to the '+ Add trigger' button. On the right side of the overview, a metadata panel lists 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).

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。

## 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。