適用於 MLOps 的 FSx ONTAP

NetApp artificial intelligence solutions

NetApp
August 18, 2025

This PDF was generated from https://docs.netapp.com/zh-tw/netapp-solutions-ai/cloud/ai-mlops-fsxn-intro.html on August 18, 2025. Always check docs.netapp.com for the latest.

目錄

適用於 MLOps 的 FSx ONTAP . . . 1
適用於 MLOps 的Amazon FSx for NetApp ONTAP (FSx ONTAP) . . . 1
第 1 部分 - 將Amazon FSx for NetApp ONTAP (FSx ONTAP) 作為私有 S3 儲存桶整合到 AWS SageMaker . . . 1
介紹 . . . 1
使用者指南 . . . 1
有用的調試清單 . . . 14
常見問題（截至 2023 年 9 月 27 日） . . . 15
第 2 部分 - 利用 AWS Amazon FSx for NetApp ONTAP (FSx ONTAP) 作為 SageMaker 模型訓練的資料來源 . . . 15
介紹 . . . 15
什麼是 FSx ONTAP . . . 15
先決條件 . . . 15
整合概述 . . . 16
逐步集成 . . . 17
第 3 部分 - 建立簡化的 MLOps 管道 (CI/CT/CD) . . . 24
介紹 . . . 24
顯現 . . . 24
先決條件 . . . 24
架構 . . . 25
逐步配置 . . . 25

適用於 MLOps 的 FSx ONTAP

適用於 MLOps 的Amazon FSx for NetApp ONTAP (FSx ONTAP)

本節深入探討 AI 基礎架構開發的實際應用，提供使用 FSx ONTAPMLOps 管道的端對端演練。它包含三個綜合範例，引導您透過這個強大的資料管理平台滿足您的 MLOps 需求。

這些文章重點關注：

1. "第 1 部分 - 將Amazon FSx for NetApp ONTAP (FSx ONTAP) 作為私有 S3 儲存桶整合到 AWS SageMaker"
2. "第 2 部分 - 利用Amazon FSx for NetApp ONTAP (FSx ONTAP) 作為 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，您將對如何使用 FSx ONTAP簡化 MLOps 流程有深入的了解。

第 1 部分 - 將Amazon FSx for NetApp ONTAP (FSx ONTAP) 作為私有 S3 儲存桶整合到 AWS SageMaker

本節提供了使用 AWS SageMaker 將 FSx ONTAP配置為私有 S3 儲存桶的指南。

介紹

以 SageMaker 為例，本頁面提供了將 FSx ONTAP配置為私有 S3 儲存桶的指導。

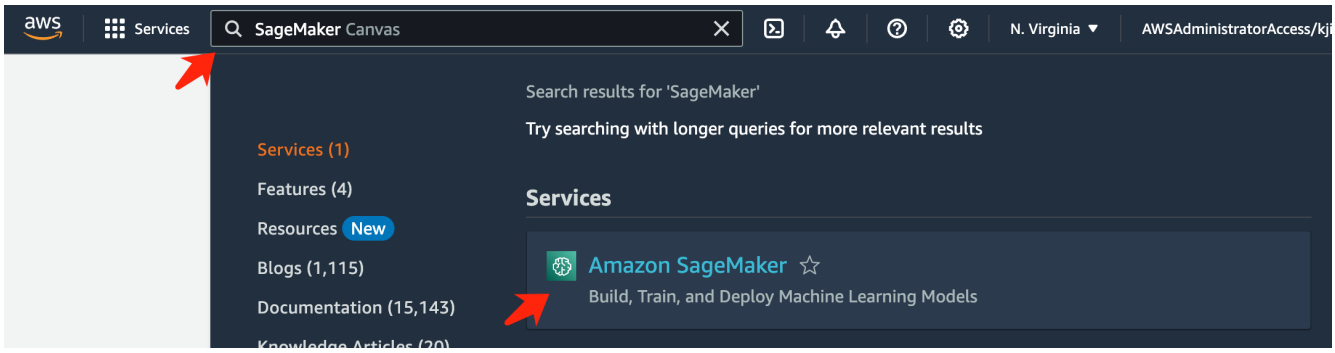
有關 FSx ONTAP的更多信息，請查看此簡報（"影片連結" ）

使用者指南

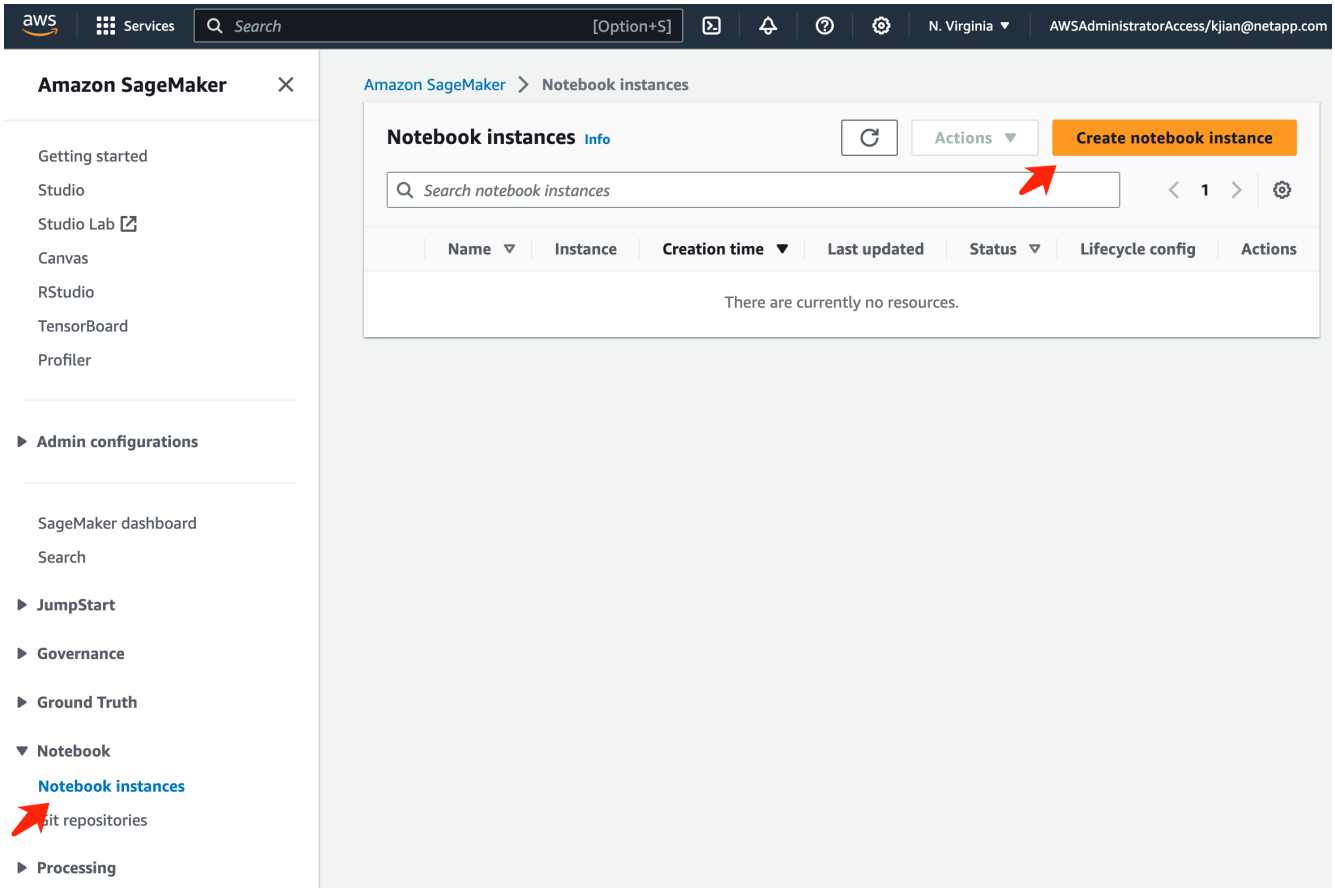
伺服器創建

建立 **SageMaker Notebook** 實例

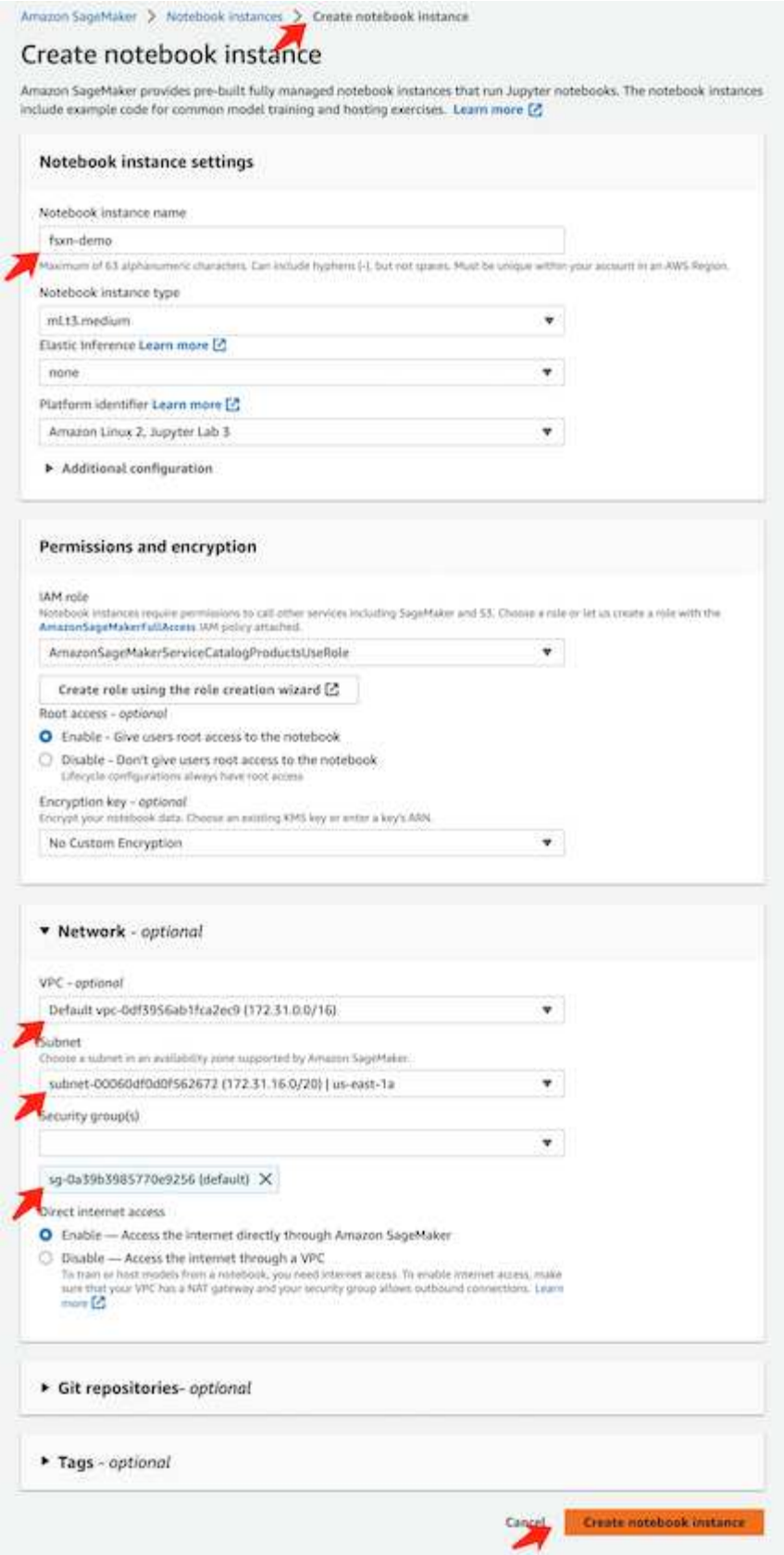
1. 開啟 AWS 主控台。在搜尋面板中，搜尋 SageMaker 並點擊服務 **Amazon SageMaker**。



2. 開啟 Notebook 標籤下的 **Notebook** 實例，點選橘色按鈕 建立筆記本實例。

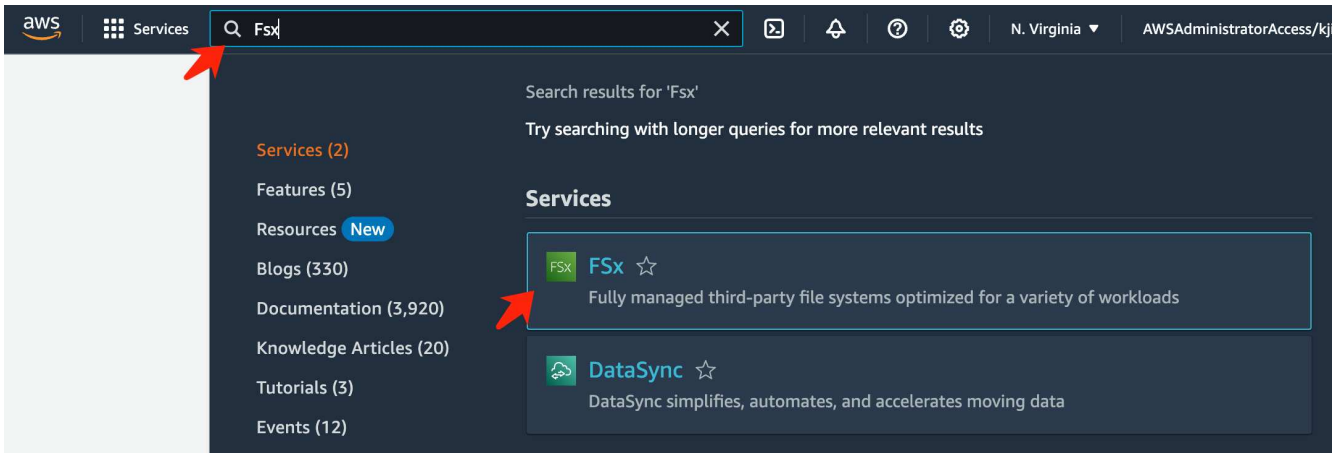


3. 在建立頁面中，輸入*筆記本實例名稱*展開*網路*面板保留其他項目的預設值，並選擇*VPC*、子網路*和*安全群組。 （此*VPC*和*Subnet*稍後將用於建立 FSx ONTAP檔案系統）點擊右下角的橘色按鈕*建立筆記本實例*。

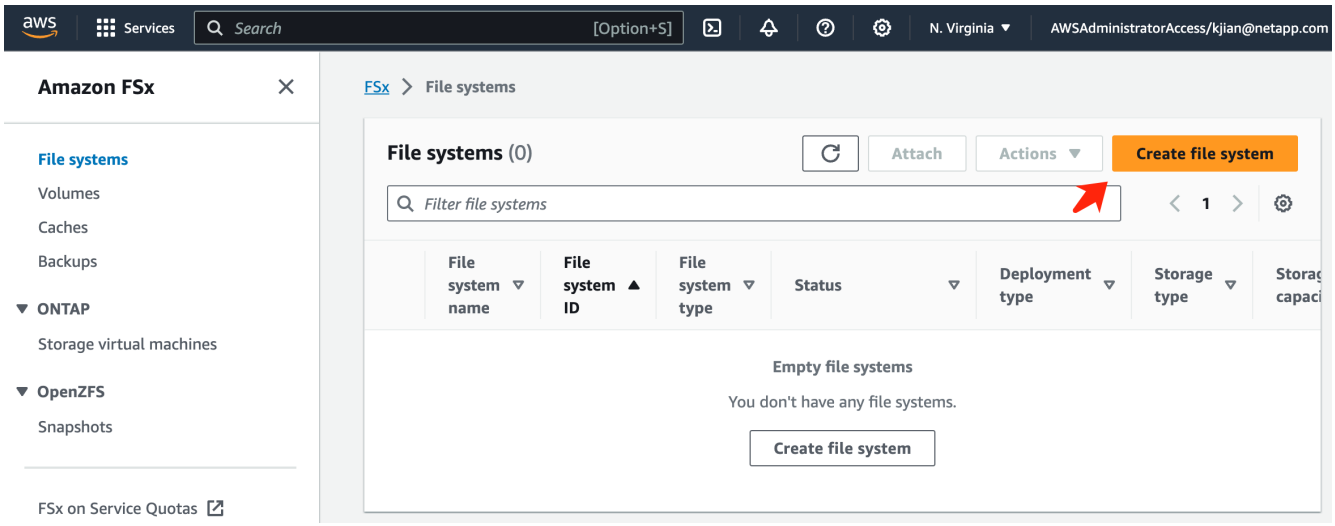


建立 **FSx ONTAP**檔案系統

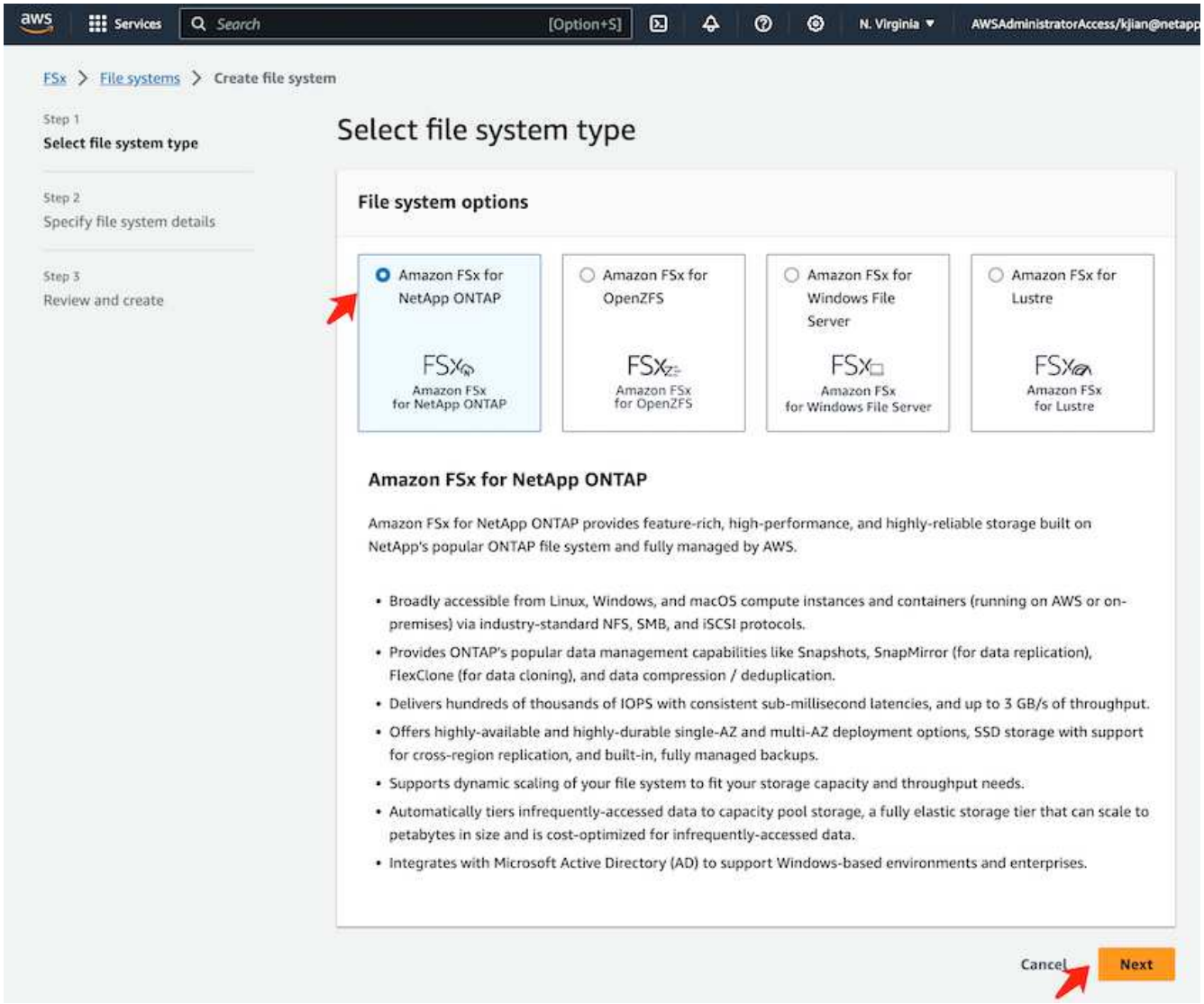
1. 開啟 AWS 主控台。在搜尋面板中，搜尋 Fsx 並點擊服務 **FSx**。



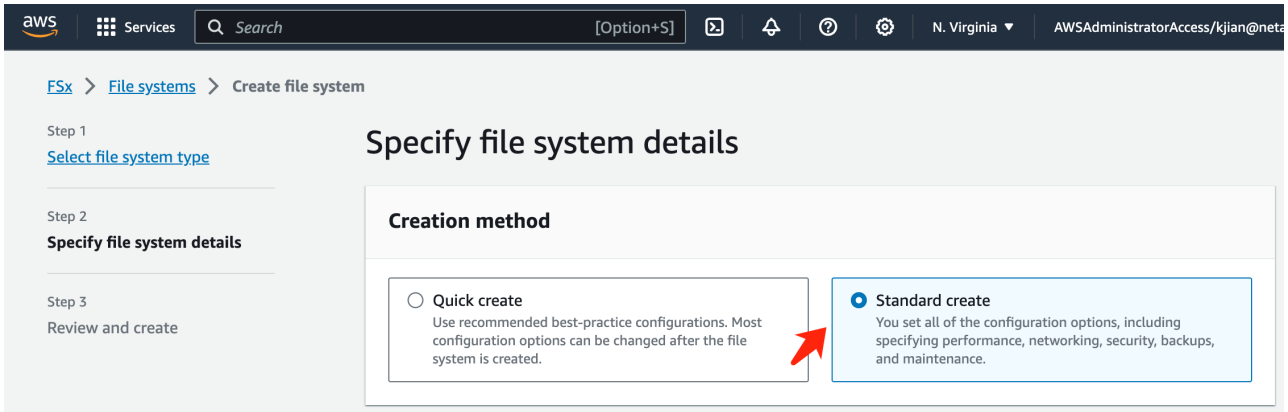
2. 點選*建立檔案系統*。



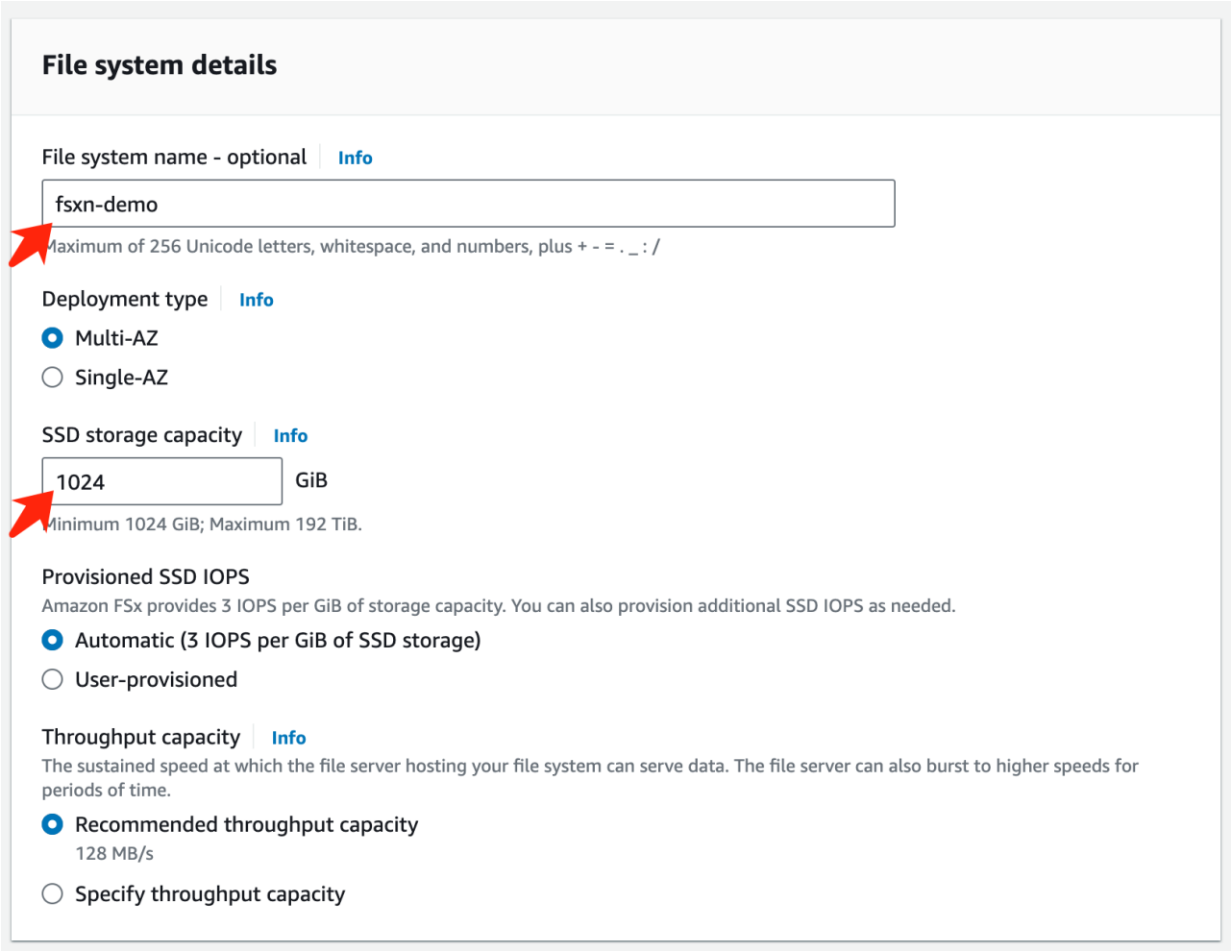
3. 選擇第一張卡 **FSx ONTAP** 並點選 下一步。



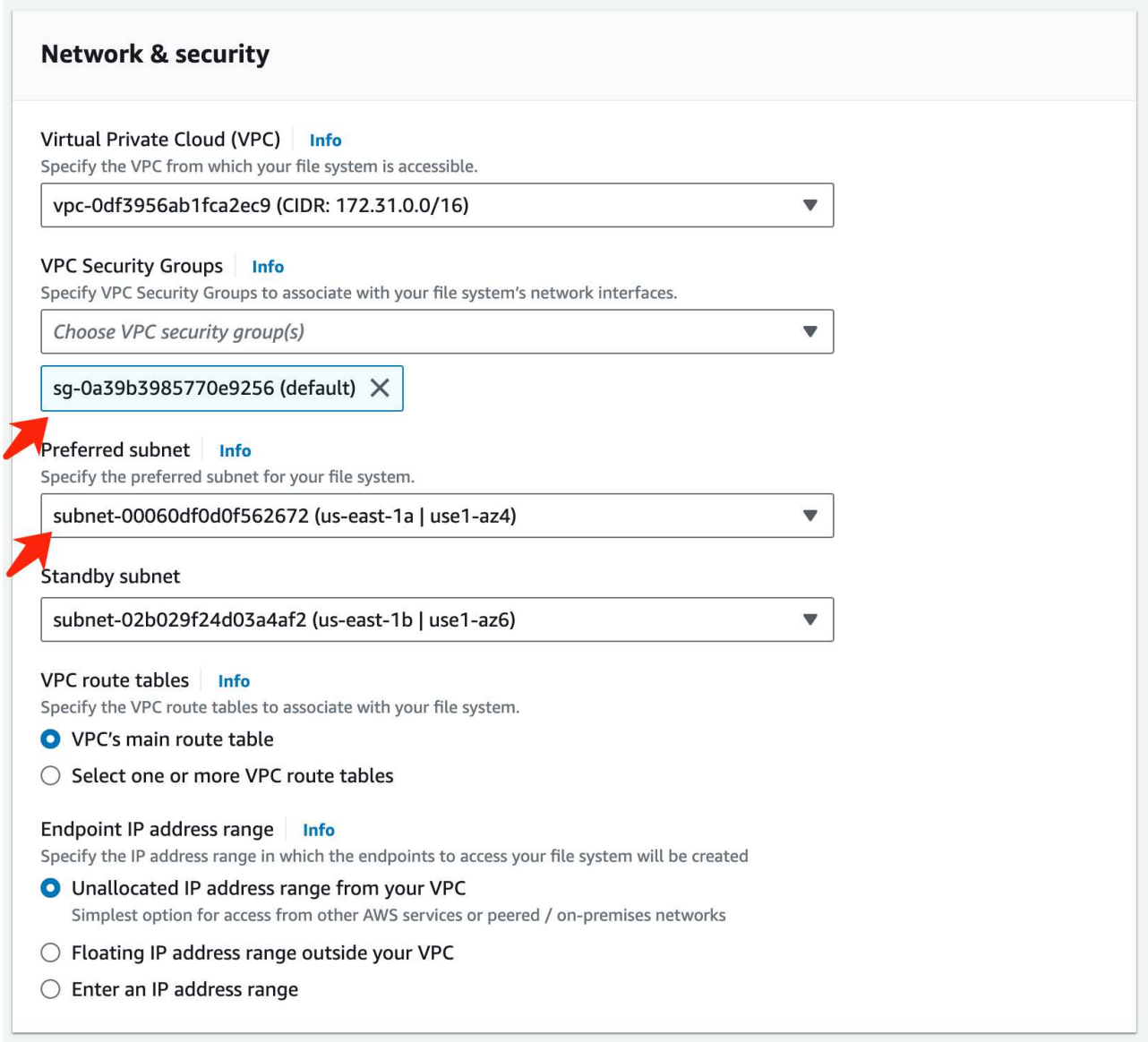
4. 在詳細資訊配置頁面中。
   a. 選擇*標準建立*選項。



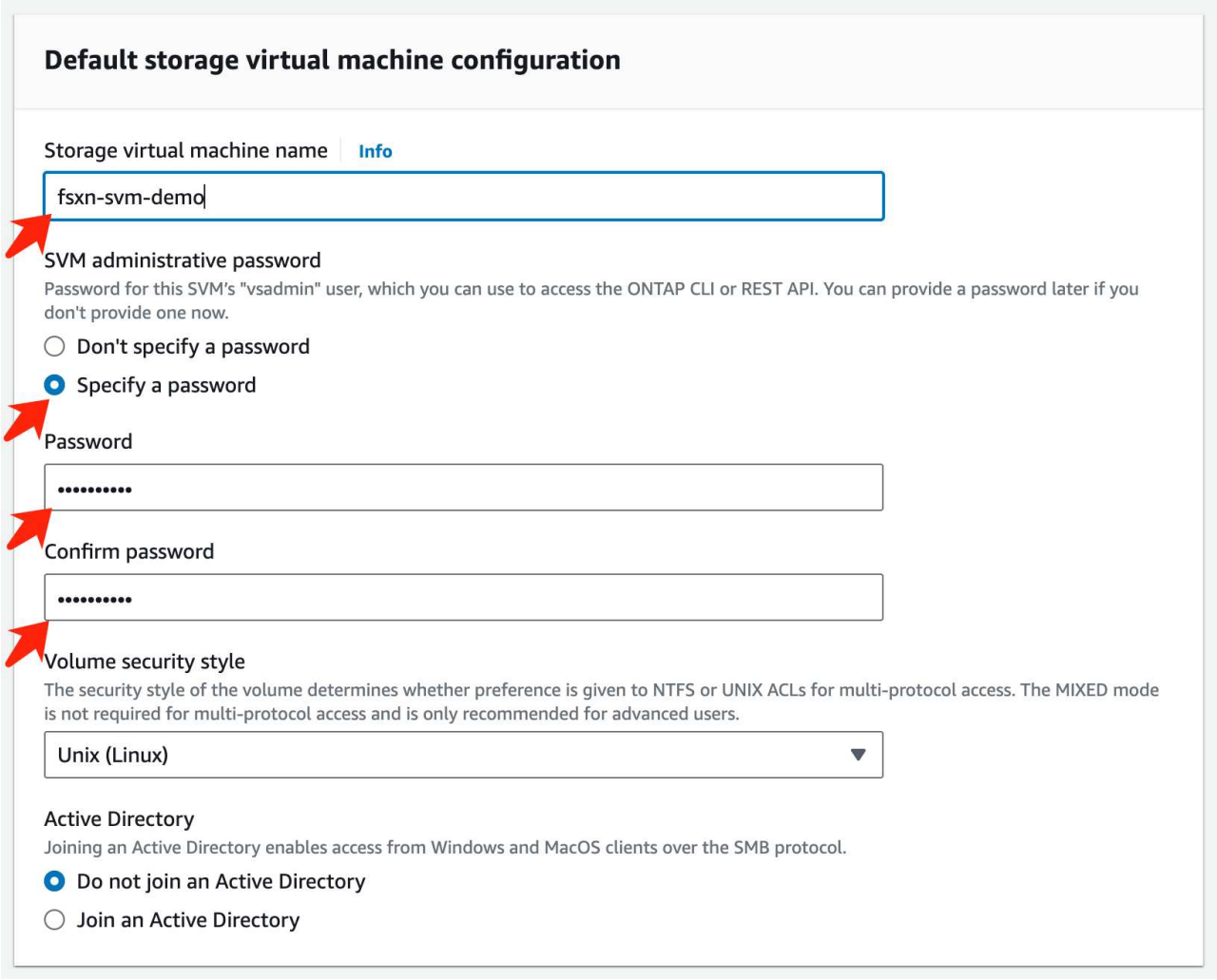
   b. 輸入*檔案系統名稱*和*SSD儲存容量*。



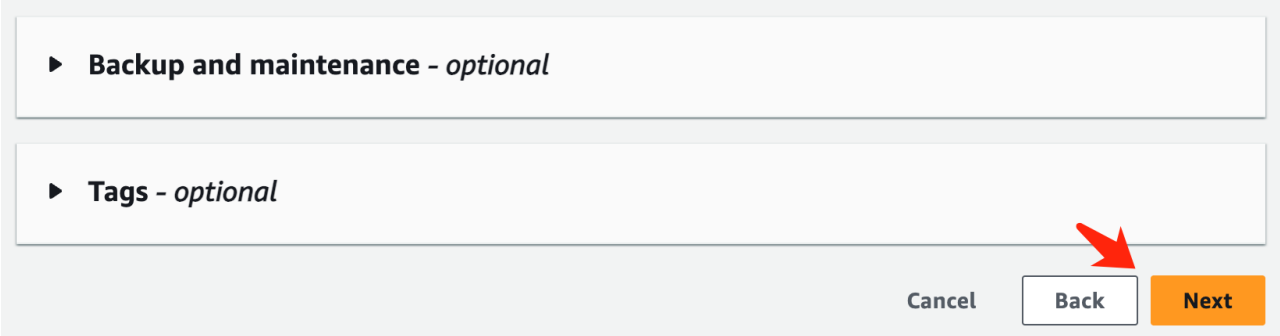
c. 確保使用與 **SageMaker Notebook** 實例相同的 **VPC** 和 **subnet**。



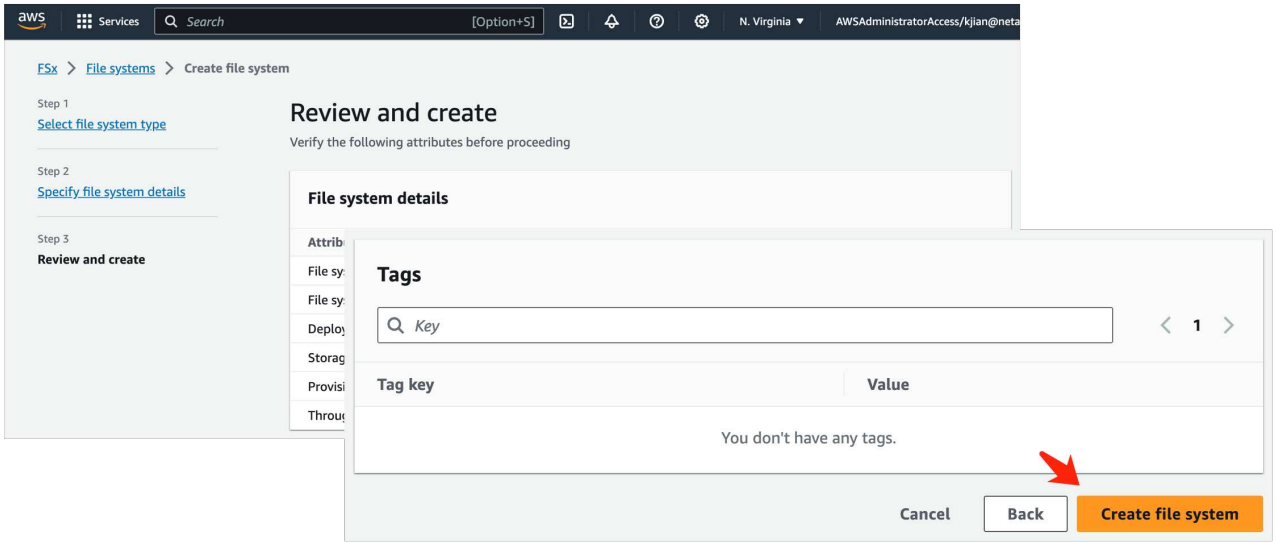
d. 為您的 SVM（儲存虛擬機器）輸入*儲存虛擬機器*名稱和*指定密碼*。



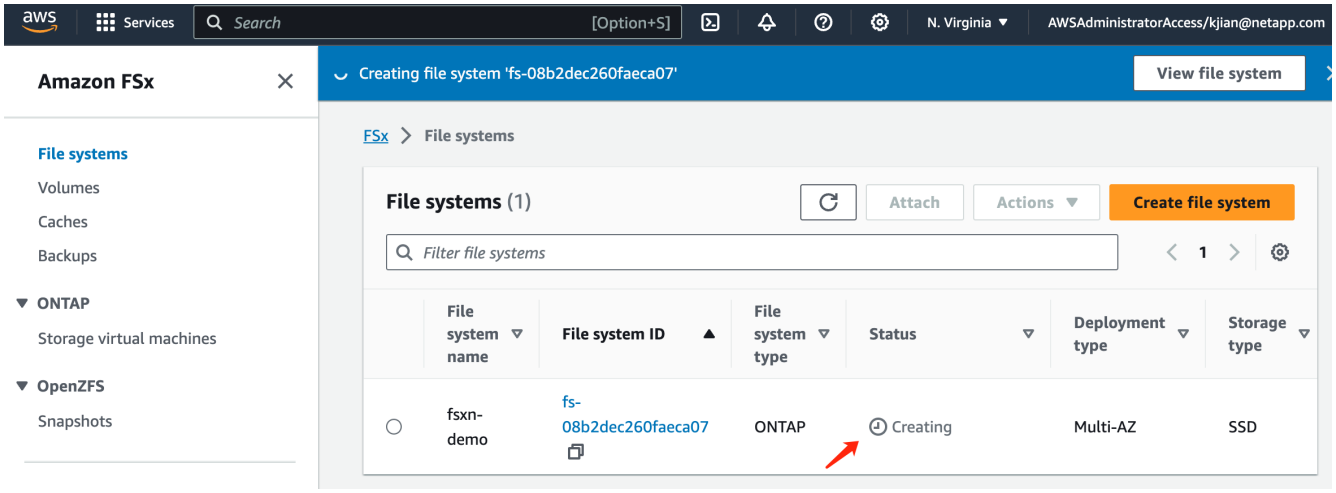
e. 保留其他條目的預設值，然後按一下右下角的橘色按鈕「下一步」。



f. 點選審核頁面右下角的橘色按鈕*建立檔案系統*。



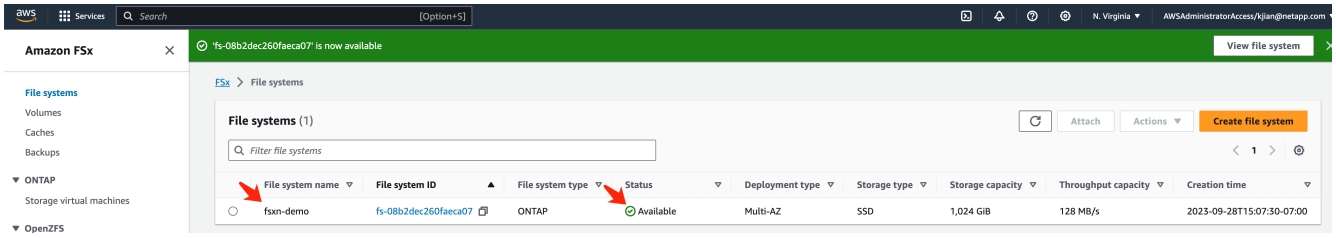
5. 啟動 FSx 檔案系統可能需要大約 **20-40** 分鐘。



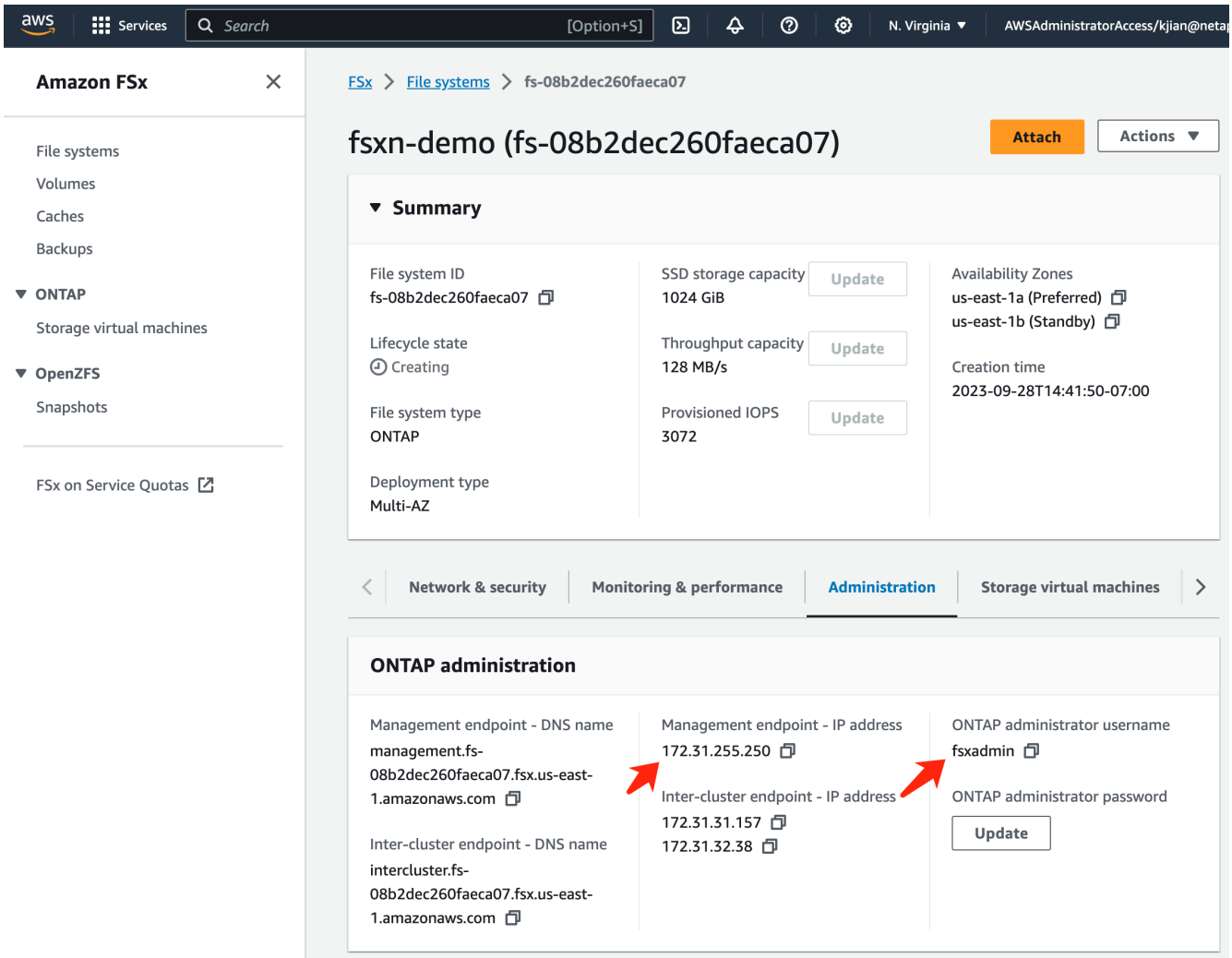
伺服器配置

ONTAP配置

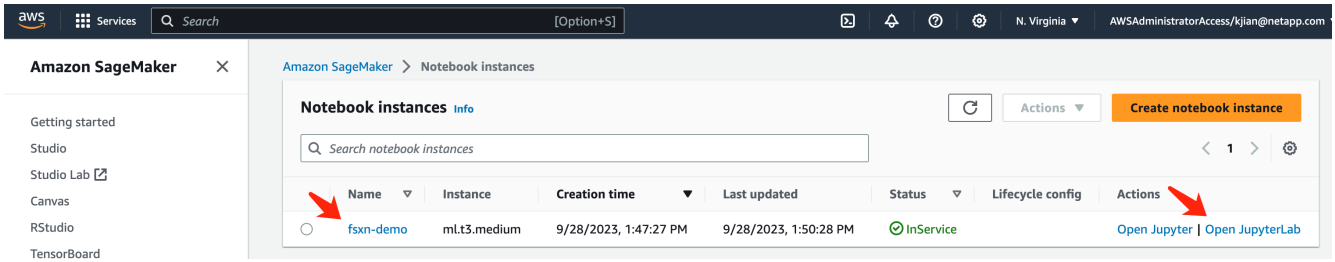
1. 開啟已建立的FSx檔案系統。請確保狀態為*可用*。



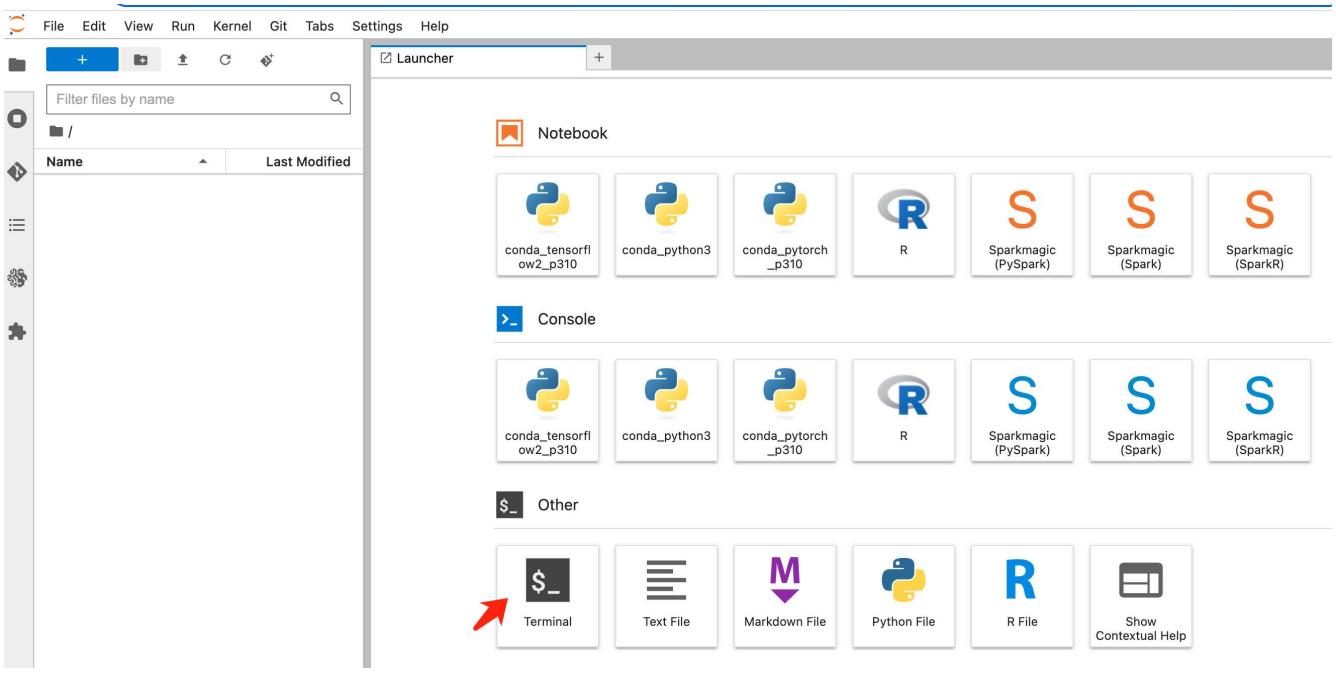
2. 選擇*管理*標籤並保留*管理端點 - IP 位址*和* ONTAP管理員使用者名稱*。



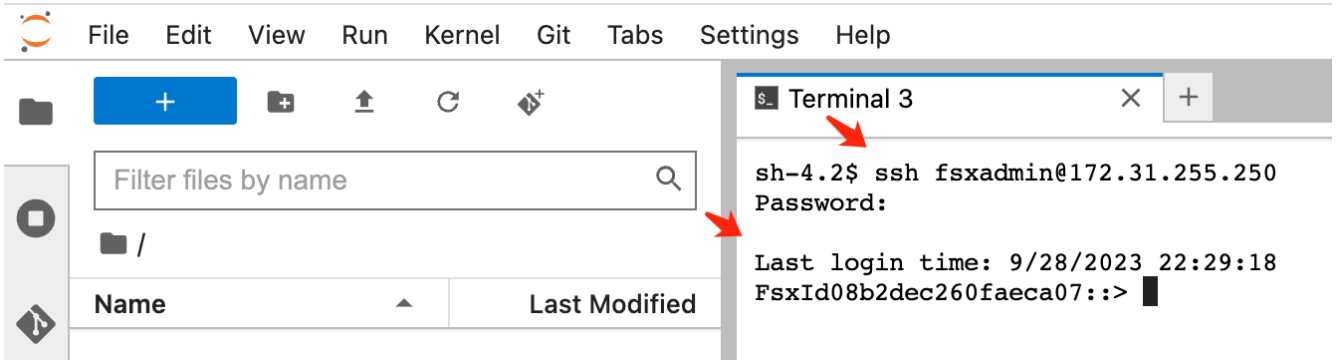
3. 開啟已建立的*SageMaker Notebook實例*，然後點選*開啟JupyterLab*。



4. 在 Jupyter Lab 頁面中，開啟一個新的*終端機*。

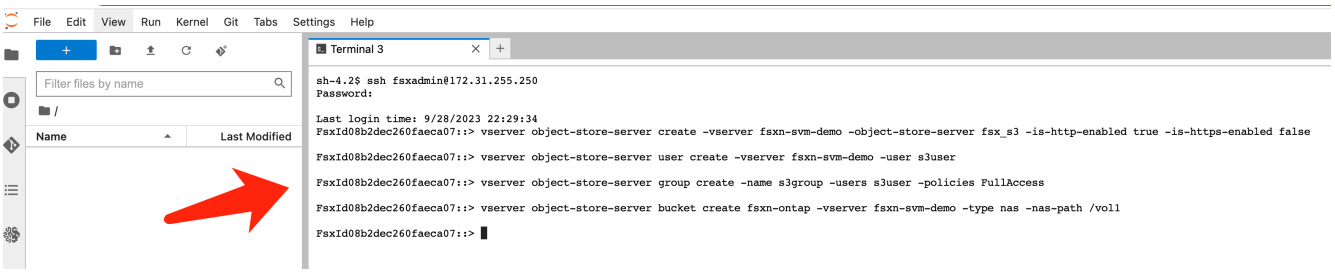


5. 輸入 ssh 命令 ssh <管理員使用者名稱>@< ONTAP伺服器 IP> 登入 FSx ONTAP檔案系統。 （使用者名稱和 IP 位址從步驟 2 檢索）請使用建立 儲存虛擬機器 時使用的密碼。



6. 按以下順序執行命令。我們使用 **fsxn-ontap** 作為 **FSx ONTAP**私有 **S3** 儲存桶名稱 的名稱。請使用*儲存虛擬機器名稱*作為*-vserver* 參數。

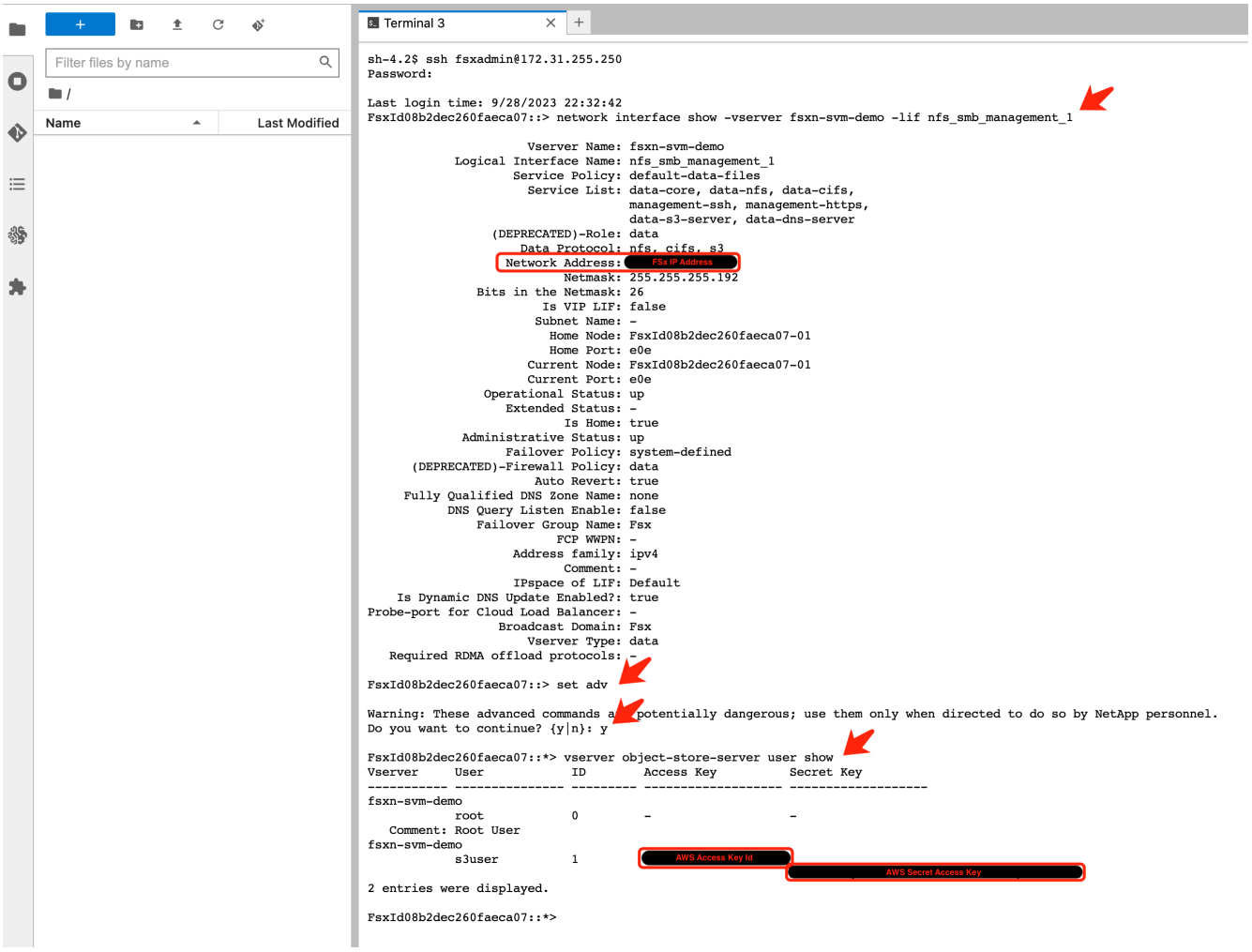
```
vserver object-store-server create -vserver fsxn-svm-demo -object-store
-server fsx_s3 -is-http-enabled true -is-https-enabled false

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. 執行下列指令來擷取 FSx ONTAP私有 S3 的端點 IP 和憑證。

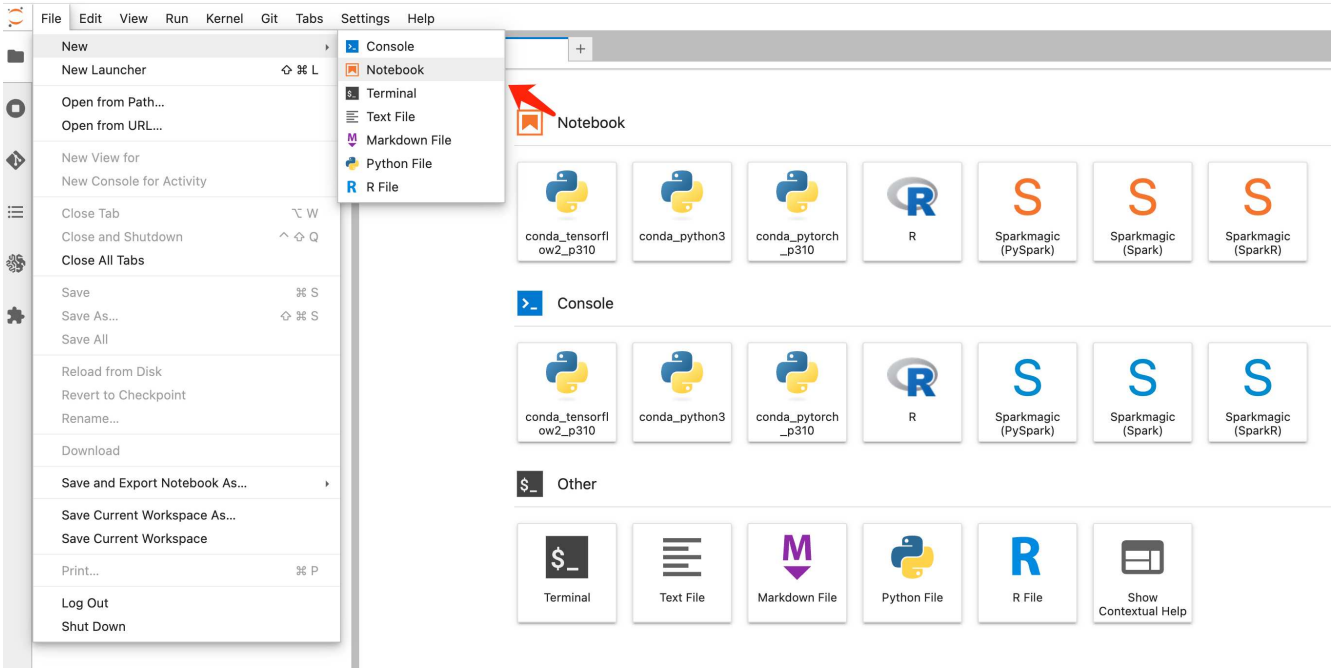
```
network interface show -vserver fsxn-svm-demo -lif nfs_smb_management_1

set adv

vserver object-store-server user show
```

8. 保留端點 IP 和憑證以供將來使用。



客戶端配置

1. 在 SageMaker Notebook 實例中，建立一個新的 Jupyter 筆記本。



2. 使用下列程式碼作為解決方案將檔案上傳到 FSx ONTAP私人 S3 儲存桶。有關全面的程式碼範例，請參閱此筆記本。"fsxn_demo.ipynb"

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

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

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

```
!aws configure set aws_secret_access_key $aws_secret_access_key

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

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

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

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

# Initialize 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']
```

這完成了 FSx ONTAP和 SageMaker 實例之間的整合。

有用的調試清單

- 確保 SageMaker Notebook 實例和 FSx ONTAP檔案系統位於同一個 VPC 中。
- 記得在ONTAP上執行 **set dev** 指令將權限等級設定為 **dev**。

常見問題（截至 **2023** 年 **9** 月 **27** 日）

Q：為什麼我在將檔案上傳到 FSx ONTAP時收到錯誤「呼叫 **CreateMultipartUpload** 操作時發生錯誤（未實作）：您要求的 **s3** 命令未實作」？

答：作為私有 S3 儲存桶，FSx ONTAP支援上傳最大 100MB 的檔案。使用S3協定時，大於100MB的檔案會被分成100MB的區塊，並呼叫'CreateMultipartUpload'函數。但是，FSx ONTAP私有 S3 的目前實作不支援此功能。

Q：為什麼在將檔案上傳到 FSx ONTAP時出現錯誤「呼叫 **PutObject** 操作時發生錯誤（**AccessDenied**）：存取被拒絕」？

答：若要從 SageMaker Notebook 執行個體存取 FSx ONTAP私有 S3 儲存桶，請將 AWS 憑證切換到 FSx ONTAP憑證。但是，授予實例寫入權限需要一種解決方法，即安裝儲存桶並執行「chmod」shell 命令來更改權限。

Q：如何將 FSx ONTAP私有 S3 儲存桶與其他 SageMaker ML 服務整合？

答：遺憾的是，SageMaker 服務 SDK 並沒有提供指定私有 S3 儲存桶端點的方法。因此，FSx ONTAP S3 與 Sagemaker Data Wrangler、Sagemaker Clarify、Sagemaker Glue、Sagemaker Athena、Sagemaker AutoML 等 SageMaker 服務不相容。

第 **2** 部分 **-** 利用 **AWS Amazon FSx for NetApp ONTAP (FSx ONTAP)** 作為 **SageMaker** 模型訓練的資料來源

本文是關於使用Amazon FSx for NetApp ONTAP (FSx ONTAP) 在 SageMaker 中訓練 PyTorch 模型的教程，具體針對輪胎品質分類項目。

介紹

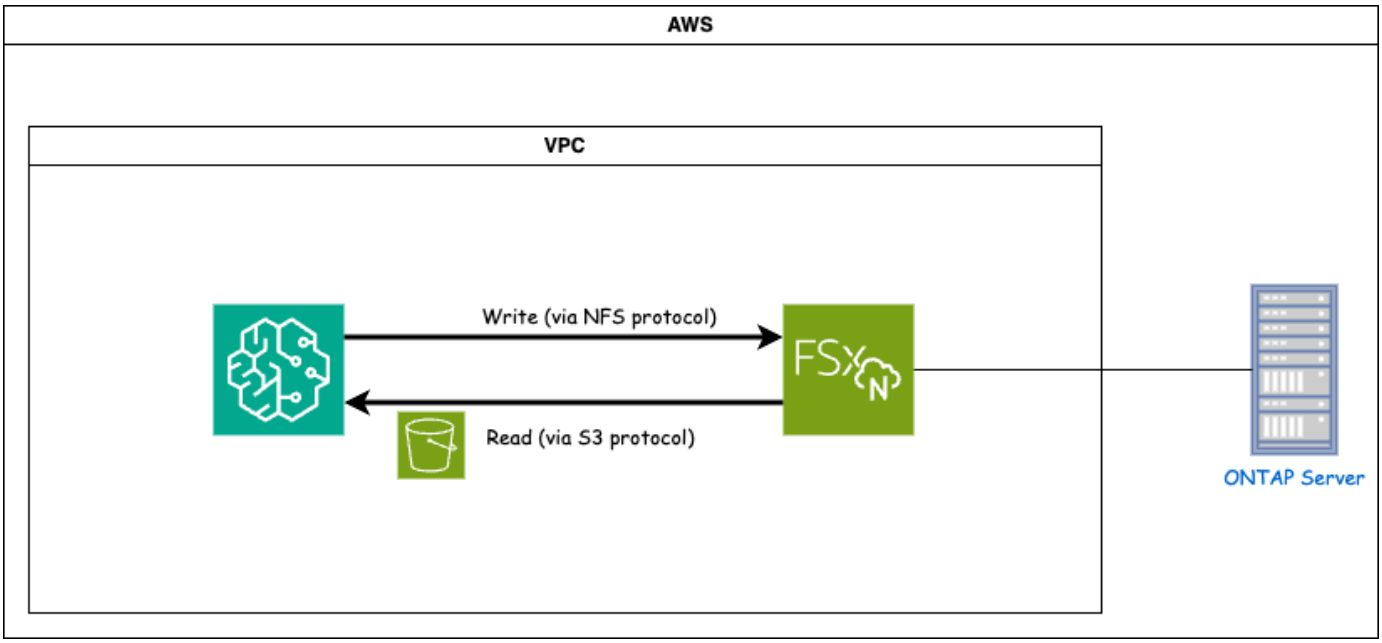
本教學提供了一個電腦視覺分類專案的實際範例，提供了在 SageMaker 環境中利用 FSx ONTAP作為資料來源建立 ML 模型的實務經驗。該專案專注於使用深度學習框架 PyTorch 根據輪胎影像對輪胎品質進行分類。它強調使用 FSx ONTAP作為 Amazon SageMaker 中的資料來源來開發機器學習模型。

什麼是 **FSx ONTAP**

Amazon FSx ONTAP確實是 AWS 提供的完全託管的儲存解決方案。它利用 NetApp 的ONTAP檔案系統提供可靠且高效能的儲存。透過支援 NFS、SMB 和 iSCSI 等協議，它允許從不同的計算實例和容器進行無縫存取。該服務旨在提供卓越的效能，確保快速且有效率的資料操作。它還具有高可用性和耐用性，確保您的資料保持可存取和保護。此外， Amazon FSx ONTAP的儲存容量是可擴充的，您可以根據需要輕鬆調整。

先決條件

網路環境



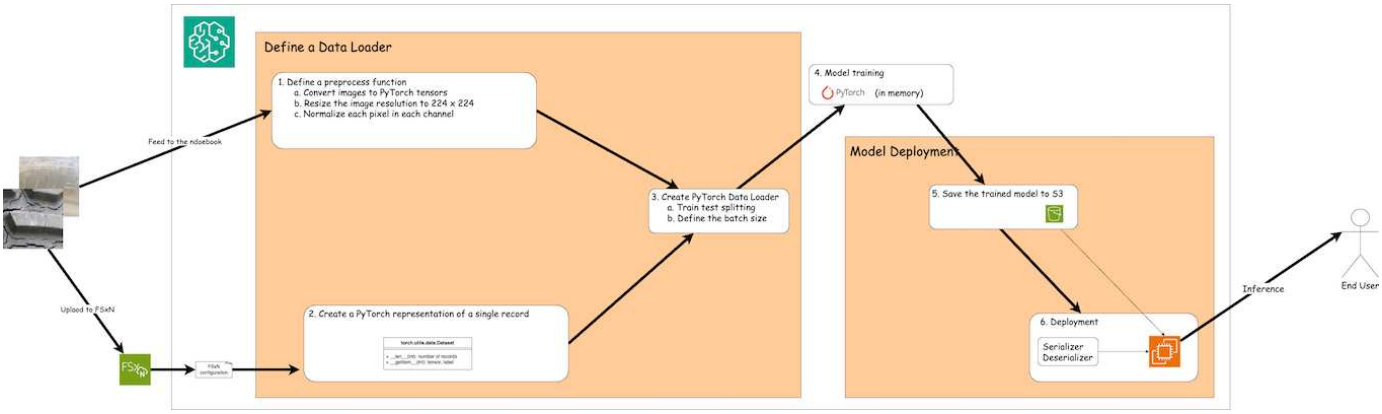
FSx ONTAP （Amazon FSx ONTAP）是一項 AWS 儲存服務。它包括在NetApp ONTAP系統上運行的檔案系統和連接到它的 AWS 管理的系統虛擬機器 (SVM)。在提供的圖中，AWS 管理的NetApp ONTAP伺服器位於 VPC 外部。 SVM作為SageMaker和NetApp ONTAP系統之間的中介，接收來自SageMaker的操作請求並轉送到底層儲存。若要存取 FSx ONTAP，SageMaker 必須放置在與 FSx ONTAP部署相同的 VPC 內。此配置可確保 SageMaker 和 FSx ONTAP之間的通訊和資料存取。

資料存取

在現實場景中，資料科學家通常利用 FSx ONTAP中儲存的現有資料來建立他們的機器學習模型。但是，出於演示目的，由於 FSx ONTAP檔案系統在建立後最初是空的，因此需要手動上傳訓練資料。這可以透過將 FSx ONTAP作為磁碟區安裝到 SageMaker 來實現。檔案系統成功掛載後，您可以將資料集上傳到掛載位置，以便在 SageMaker 環境中訓練模型。這種方法可讓您利用 FSx ONTAP的儲存容量和功能，同時與 SageMaker 進行模型開發和訓練。

資料讀取過程涉及將 FSx ONTAP配置為私有 S3 儲存桶。詳細配置說明請參考"第 1 部分 - 將Amazon FSx for NetApp ONTAP (FSx ONTAP) 作為私有 S3 儲存桶整合到 AWS SageMaker"

整合概述



使用 FSx ONTAP中的訓練資料在 SageMaker 中建立深度學習模型的工作流程可以概括為三個主要步驟：資料載入器定義、模型訓練和部署。從高層次來看，這些步驟構成了 MLOps 管道的基礎。然而，為了全面實施，每

個步驟都涉及幾個詳細的子步驟。這些子步驟涵蓋資料預處理、資料集拆分、模型配置、超參數調整、模型評估和模型部署等各種任務。這些步驟確保在 SageMaker 環境中使用來自 FSx ONTAP的訓練資料建構和部署深度學習模型的流程全面有效。

逐步集成

資料**Loader**

為了使用資料訓練 PyTorch 深度學習網絡，創建了一個資料載入器以方便資料的輸入。資料載入器不僅定義批次大小，還確定讀取和預處理批次中每個記錄的過程。透過配置資料載入器，我們可以批次處理數據，從而實現深度學習網路的訓練。

資料載入器由3部分組成。

預處理函數

```
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]
    )
])
```

上面的程式碼片段示範了使用 **torchvision.transforms** 模組定義影像預處理轉換。在本教程中，建立預處理物件來套用一系列轉換。首先，**ToTensor()** 轉換將影像轉換為張量表示。隨後，**Resize 224,224** 轉換將影像調整為固定大小 224x224 像素。最後，**Normalize()** 轉換透過減去平均值並除以每個通道的標準差來對張量值進行歸一化。用於標準化的平均值和標準差值通常用於預訓練的神經網路模型。總的來說，這段程式碼透過將圖像資料轉換為張量、調整其大小以及規範化像素值來準備進一步處理或輸入到預先訓練的模型中。

PyTorch 資料集類

```
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
```

此類別提供取得資料集中記錄總數的功能，並定義讀取每筆記錄資料的方法。在 ***getitem*** 函數中，程式碼利用 boto3 S3 bucket 物件從 FSx ONTAP檢索二進位資料。從 FSx ONTAP存取資料的程式碼樣式類似於從 Amazon S3 讀取資料。後續講解深入探討私有 S3 物件 **bucket** 的創建過程。

FSx ONTAP作為私有 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 FSx ONTAP IP address>'                  # Please
get this IP address from FSXN
```

```
import boto3

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

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

為了從 SageMaker 中的 FSx ONTAP讀取數據，需要建立一個使用 S3 協定指向 FSx ONTAP儲存的處理程序。這使得 FSx ONTAP可以被視為私人 S3 儲存桶。處理程序配置包括指定 FSx ONTAP SVM 的 IP 位址、儲存桶名稱和必要的憑證。有關取得這些配置項目的詳細說明，請參閱以下文件："第 1 部分 - 將Amazon FSx for NetApp ONTAP (FSx ONTAP) 作為私有 S3 儲存桶整合到 AWS SageMaker" 。

在上面的例子中，bucket 物件用於實例化 PyTorch 資料集物件。數據集對象將在後續章節中進一步解釋。

PyTorch 資料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)
```

在提供的範例中，指定批次大小為 64，表示每個批次將包含 64 筆記錄。透過結合 PyTorch **Dataset** 類別、預處理函數和訓練批次大小，我們獲得了用於訓練的資料載入器。此資料載入器有助於在訓練階段分批迭代資料集的過程。

模型訓練

```
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')
```

此程式碼實現了標準的 PyTorch 訓練流程。它定義了一個名為*TyreQualityClassifier*的神經網路模型，使用卷積層和線性層對輪胎品質進行分類。訓練循環迭代資料批次，計算損失，並使用反向傳播和最佳化更新模型的參數。此外，它還列印當前時間、紀元、批次和損失以供監控目的。

模型部署

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.split(file_name)[-1]
    tarinfo = tarfile.TarInfo(file_name_with_extension)
    tarinfo.size = len(buffer_model.getbuffer())
    buffer_model.seek(0)
    tar.addfile(tarinfo, buffer_model)

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

程式碼將 PyTorch 模型儲存到 **Amazon S3**，因為 SageMaker 要求將模型儲存在 S3 中以便部署。透過將模型上傳到 **Amazon S3**，SageMaker 就可以存取它，從而允許對已部署的模型進行部署和推理。

```
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。它定義了一個自訂序列化器 **TyreQualitySerializer**，它將輸入資料預處理並序列化為 PyTorch 張量。 **TyreQualityPredictor** 類別是自訂預測器，它利用定義的序列化器和 **JSONDeserializer**。程式碼還創建了一個 **PyTorchModel** 物件來指定模型的 S3 位置、IAM 角色、框架版本和推理的入口點。程式碼產生時間戳並根據模型和時間戳記建立端點名稱。最後，使用 deploy 方法部署模型，指定實例數量、實例類型和產生的端點名稱。這使得 PyTorch 模型可以在 SageMaker 上部署並進行推理。

```
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理的範例。

第 3 部分 - 建立簡化的 MLOps 管道 (CI/CT/CD)

本文提供了使用 AWS 服務建立 MLOps 管道的指南，重點在於自動模型再訓練、部署和成本最佳化。

介紹

在本教程中，您將學習如何利用各種 AWS 服務建立一個包含持續整合 (CI)、持續訓練 (CT) 和持續部署 (CD) 的簡單 MLOps 管道。與傳統的 DevOps 管道不同，MLOps 需要額外的考慮才能完成操作週期。透過學習本教程，您將深入了解如何將 CT 納入 MLOps 循環，從而實現模型的持續訓練和推理的無縫部署。本教學將引導您完成利用 AWS 服務建立此端對端 MLOps 管道的過程。

顯現

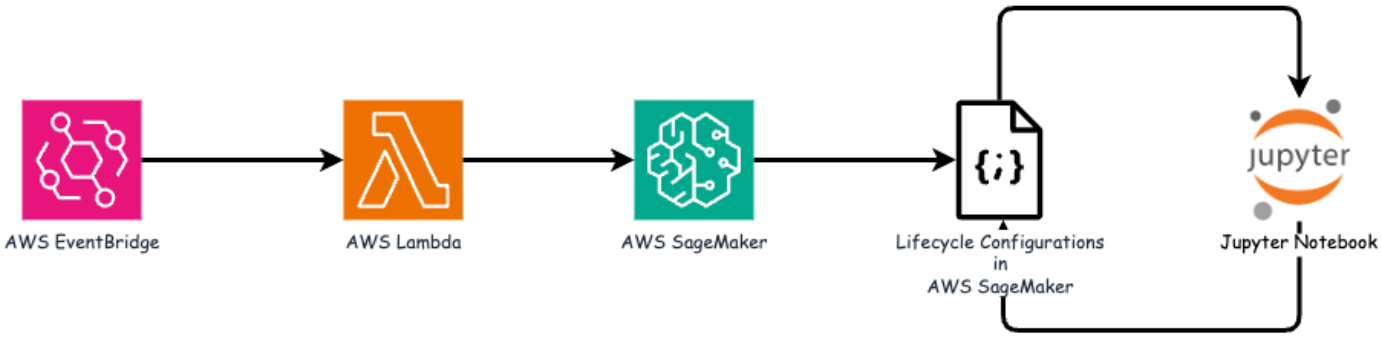
功能	**Name**	評論
資料儲存	AWS FSx ONTAP	請參閱"第 1 部分 - 將Amazon FSx for NetApp ONTAP (FSx ONTAP) 作為私有 S3 儲存桶整合到 AWS SageMaker" 。
數據科學 IDE	AWS SageMaker	本教程基於"第 2 部分 - 利用Amazon FSx for NetApp ONTAP (FSx ONTAP) 作為 SageMaker 模型訓練的資料來源"。
觸發 MLOps 管道的函數	AWS Lambda 函數	-
Cron 作業觸發器	AWS EventBridge	-
深度學習框架	PyTorch	-
AWS Python 開發工具包	boto3	-
程式設計語言	Python	v3.10

先決條件

- 預先設定的 FSx ONTAP檔案系統。本教學利用 FSx ONTAP中儲存的資料進行訓練程序。

- 配置為與上面提到的 FSx ONTAP檔案系統共用相同 VPC 的 **SageMaker Notebook** 實例。
- 在觸發 **AWS Lambda** 函數 之前，請確保 **SageMaker Notebook** 實例 處於 已停止 狀態。
- 需要 **ml.g4dn.xlarge** 執行個體類型來利用深度神經網路運算所需的 GPU 加速。

架構



這個 MLOps 管道是一個實際的實現，它利用 cron 作業來觸發無伺服器功能，進而執行使用生命週期回呼函數註冊的 AWS 服務。 **AWS EventBridge** 充當 cron 作業。它會定期呼叫負責重新訓練和重新部署模型的 **AWS Lambda** 函數。此程序涉及啟動 **AWS SageMaker Notebook** 實例以執行必要的任務。

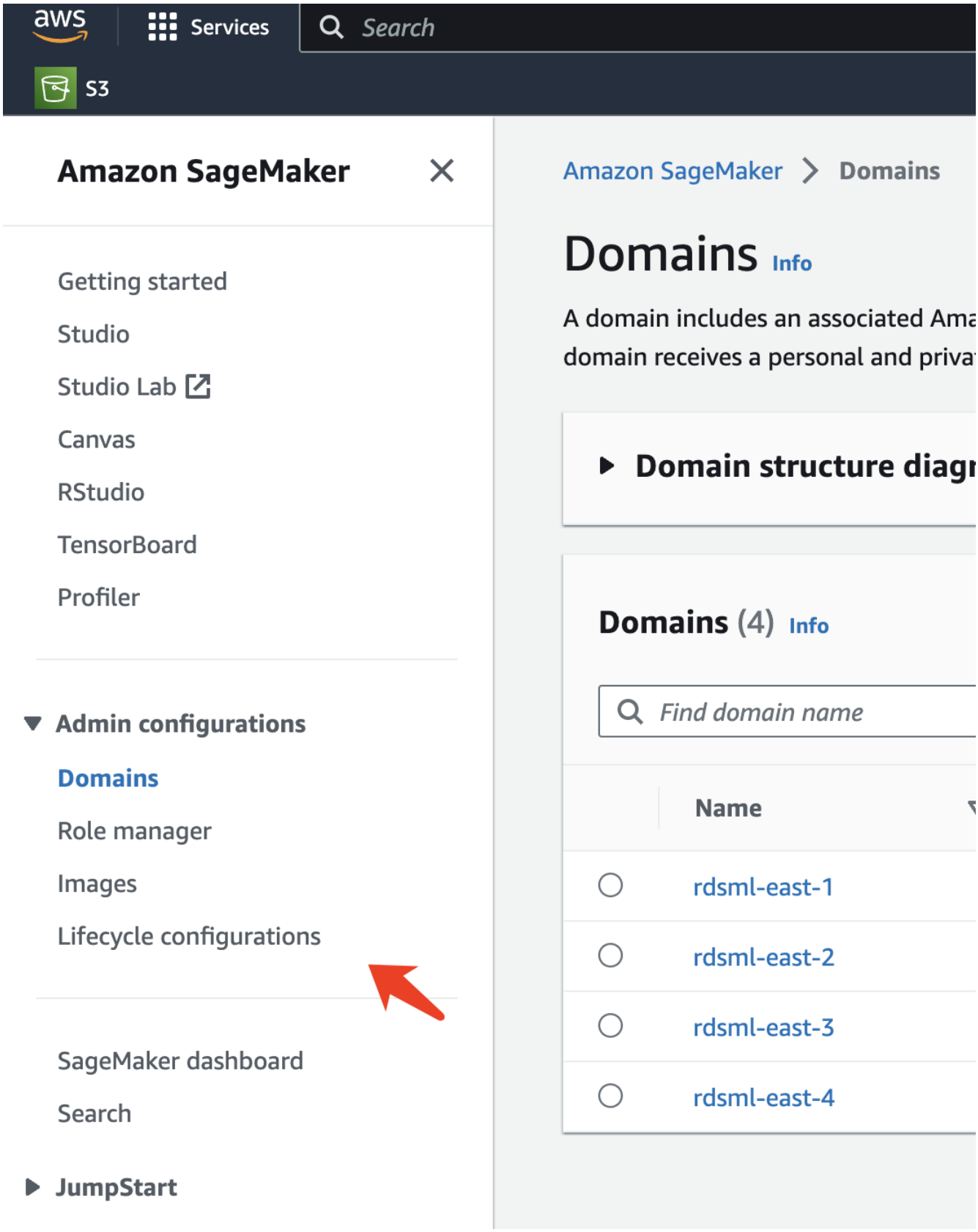
逐步配置

生命週期配置

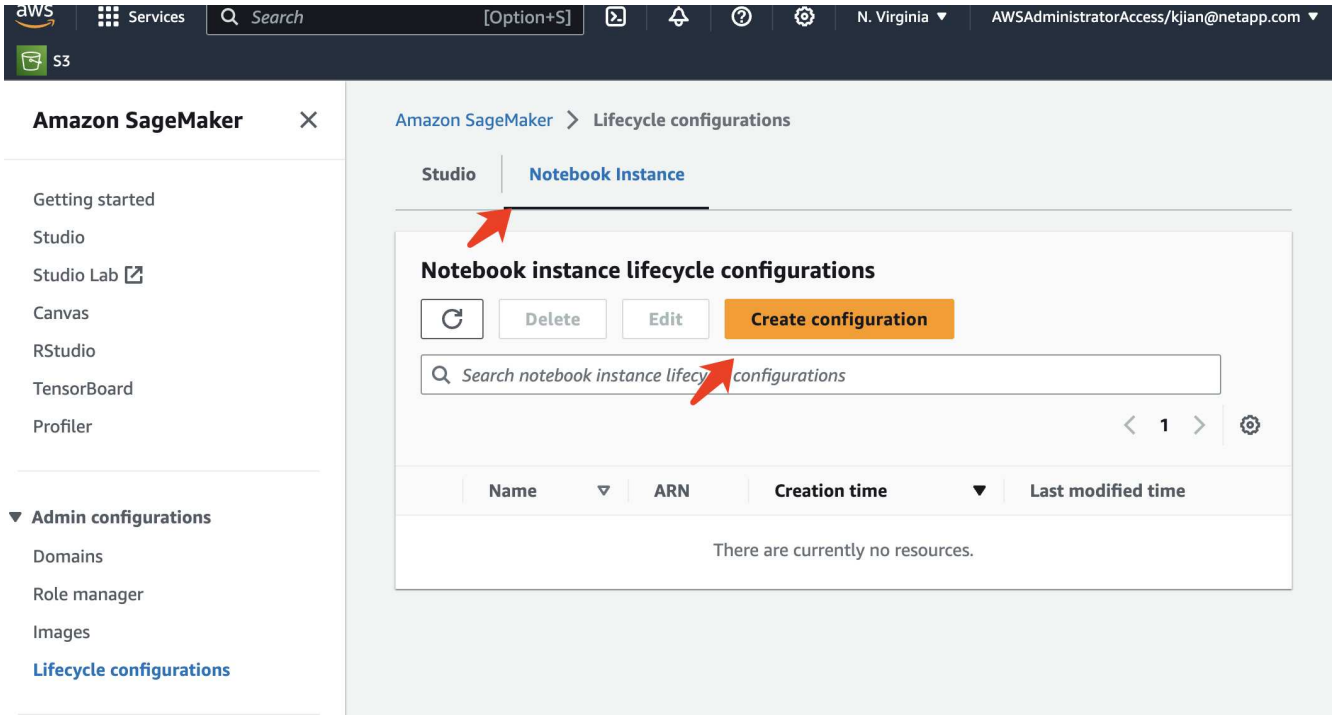
若要為 AWS SageMaker Notebook 實例配置生命週期回呼函數，您需要使用*生命週期配置*。此服務可讓您定義啟動筆記本執行個體時需要執行的必要操作。具體來說，可以在*生命週期配置*中實現一個 shell 腳本，以便在訓練和部署過程完成後自動關閉筆記本實例。這是必要的配置，因為成本是 MLOps 中的主要考慮因素之一。

值得注意的是，*生命週期配置*的配置需要事先設定。因此，建議在繼續其他 MLOps 管道設定之前優先配置這一方面。

1. 若要設定生命週期配置，請開啟 **Sagemaker** 面板並導覽至 **Admin configuration** 部分下的 **Lifecycle configuration**。



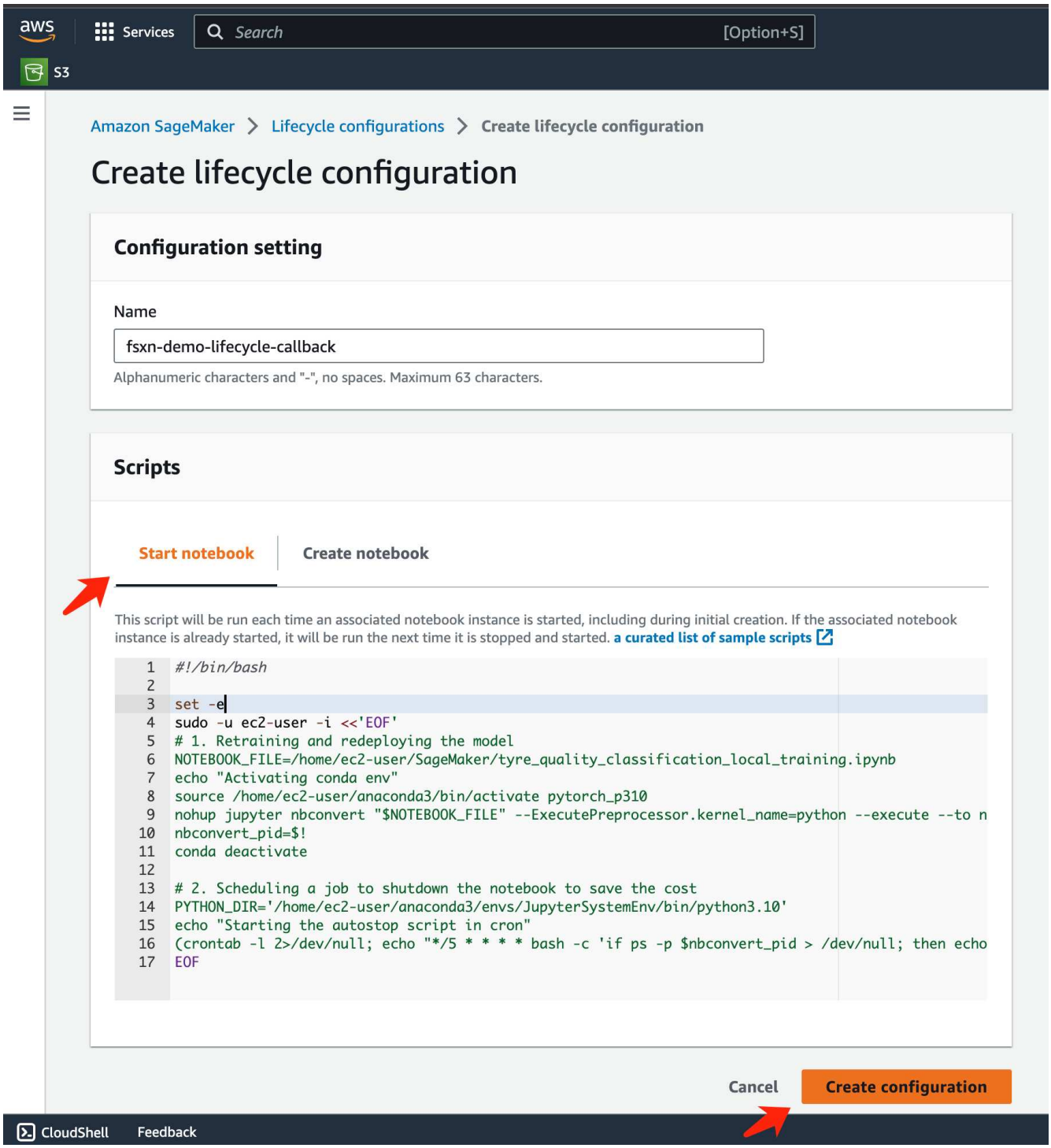
2. 選擇“筆記本實例”選項卡，然後按一下“建立配置”按鈕



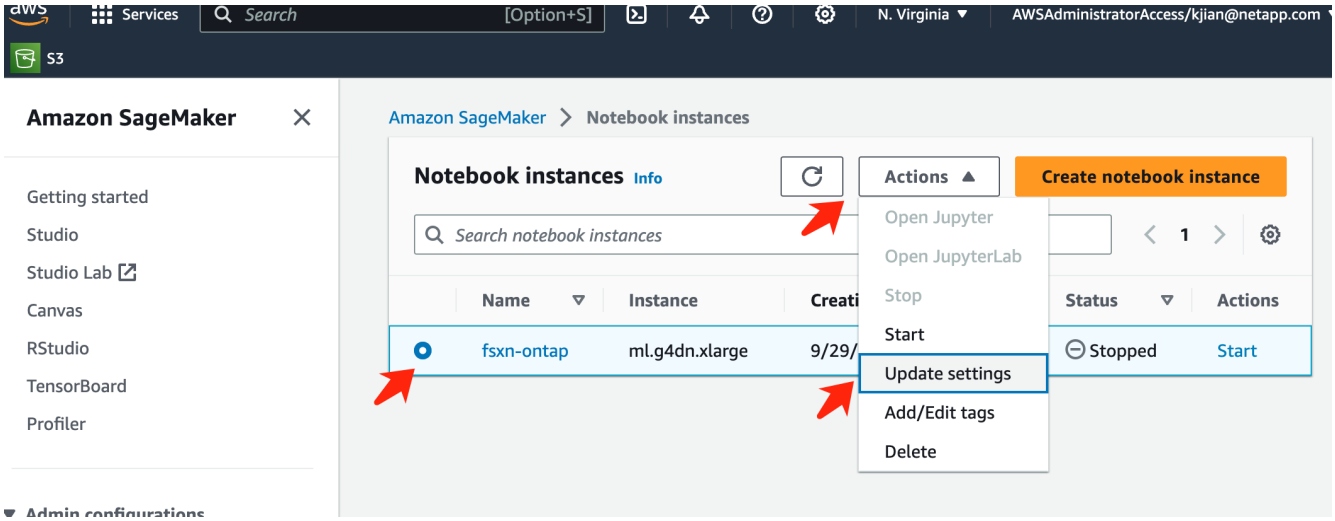
3. 將以下程式碼貼到輸入區域。

```
#!/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
```

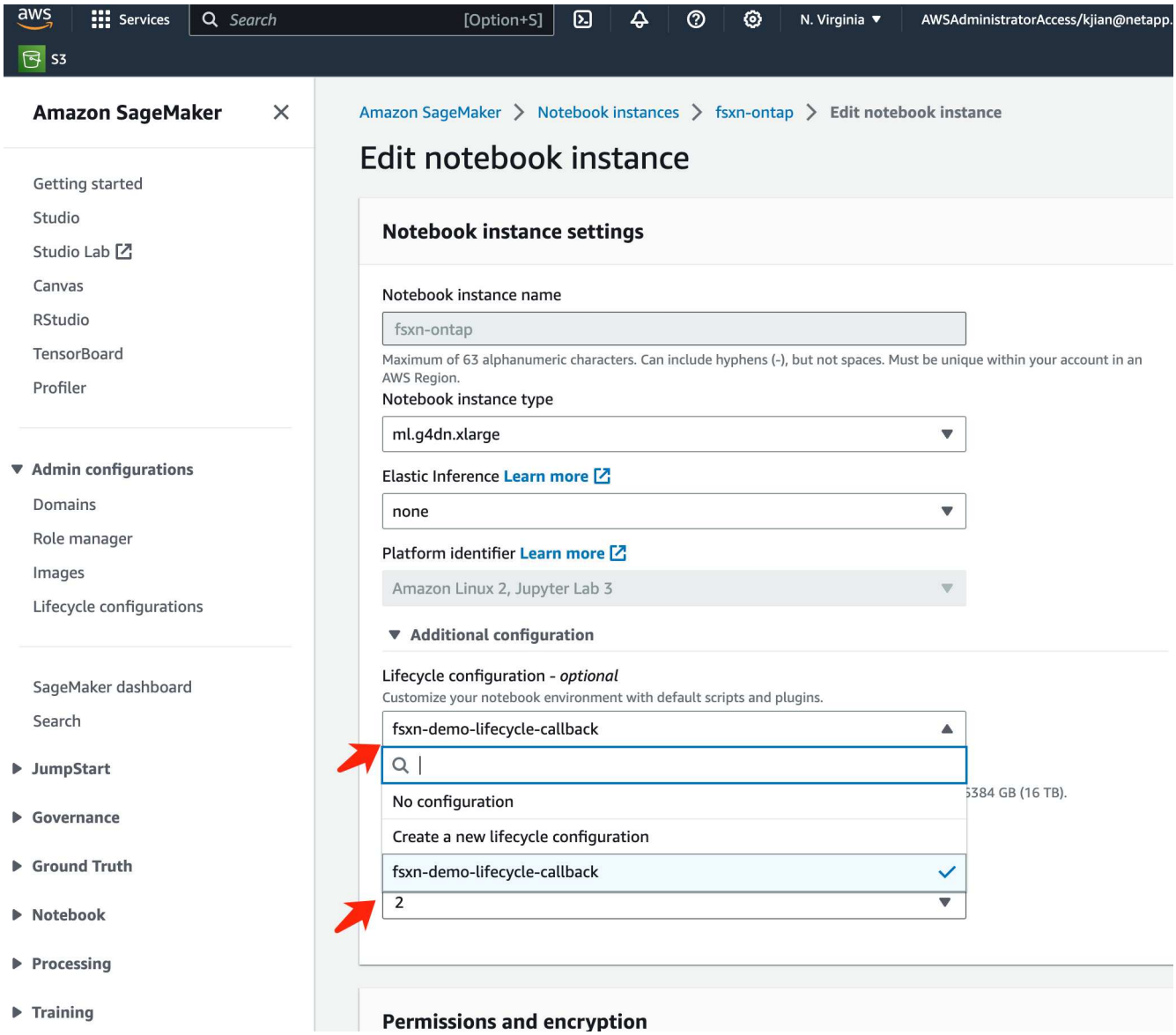
4. 該腳本執行 Jupyter Notebook，用於處理模型的重新訓練和重新部署以進行推理。執行完成後，筆記本將在5分鐘內自動關機。要了解有關問題陳述和代碼實現的更多信息，請參閱"第 2 部分 - 利用Amazon FSx for NetApp ONTAP (FSx ONTAP) 作為 SageMaker 模型訓練的資料來源"。



5. 建立後，導覽至 Notebook 實例，選擇目標實例，然後按一下「動作」下拉式功能表下的「更新設定」。



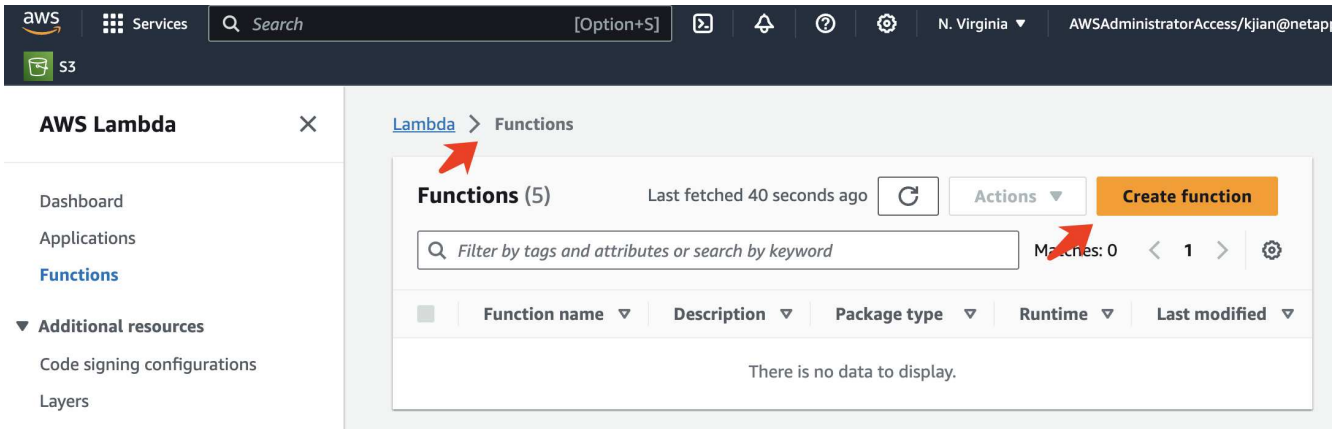
6. 選擇已建立的*生命週期配置*，然後按一下*更新筆記本實例*。



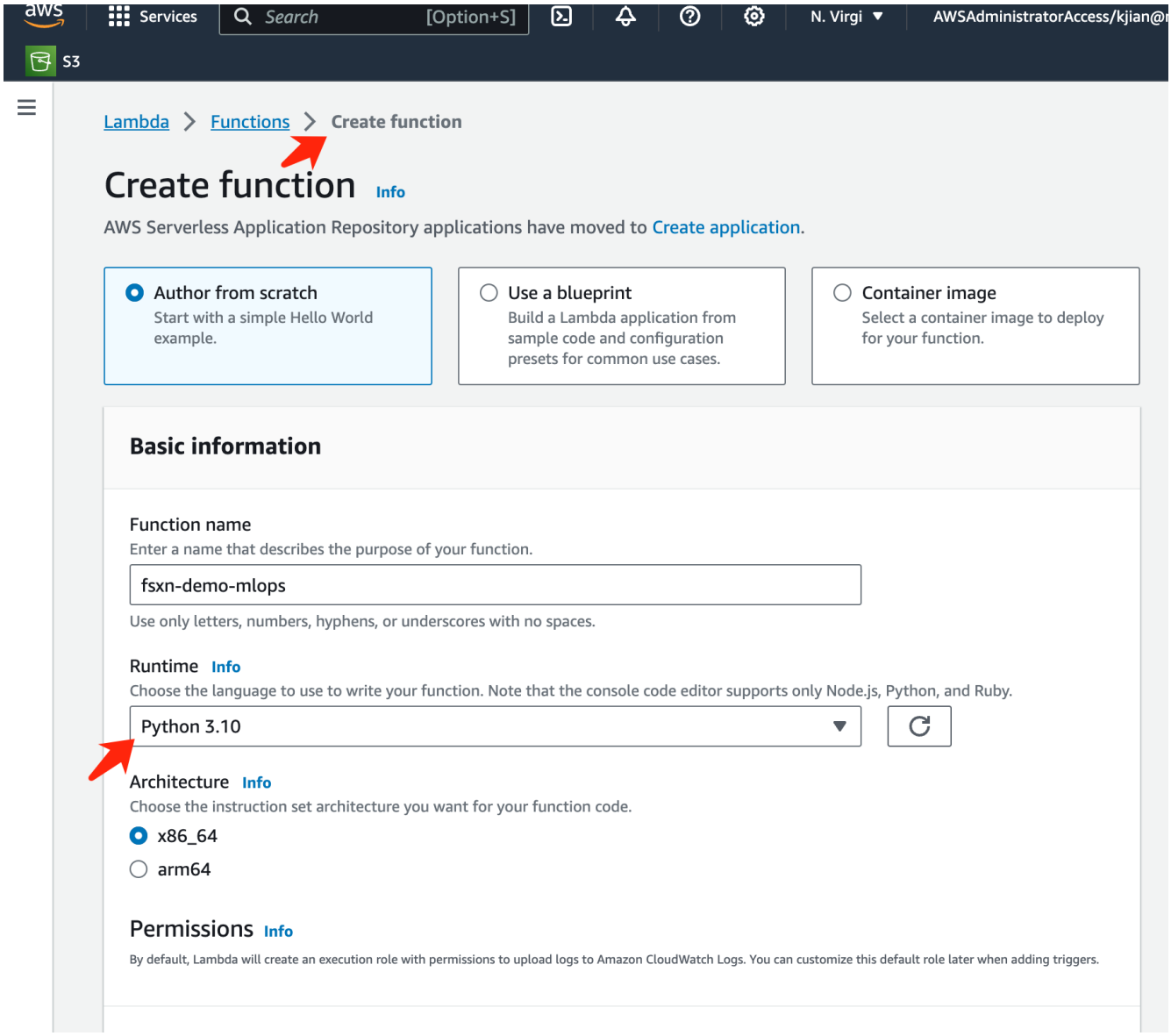
AWS Lambda 無伺服器函數

如前所述，**AWS Lambda** 函數*負責啟動 ***AWS SageMaker Notebook** 實例。

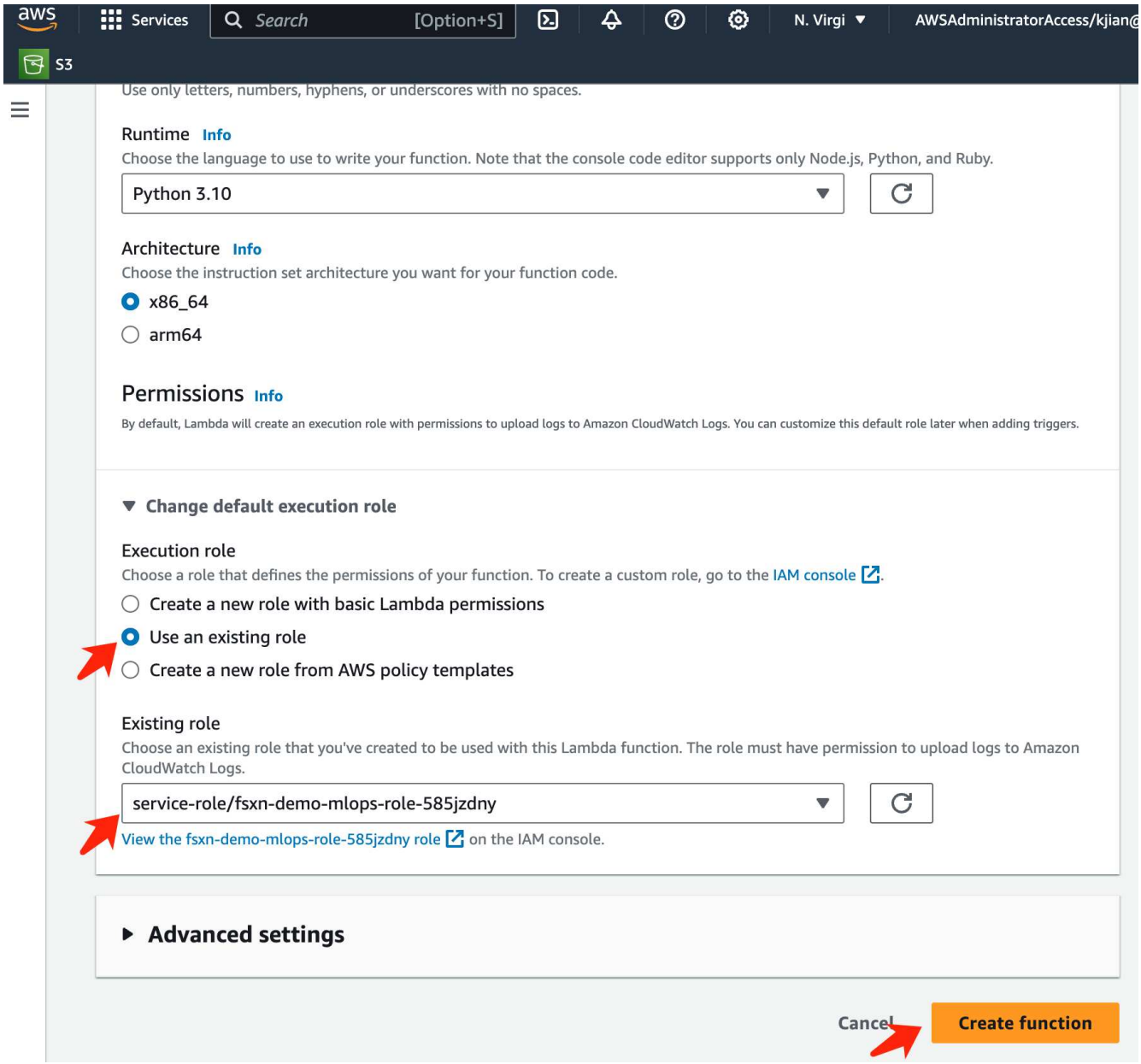
1. 若要建立 **AWS Lambda** 函數，請導覽至對應的面板，切換至 函數 選項卡，然後按一下 建立函數。



2. 請在頁面上提交所有必需的條目，並記得將運行時切換為 **Python 3.10**。



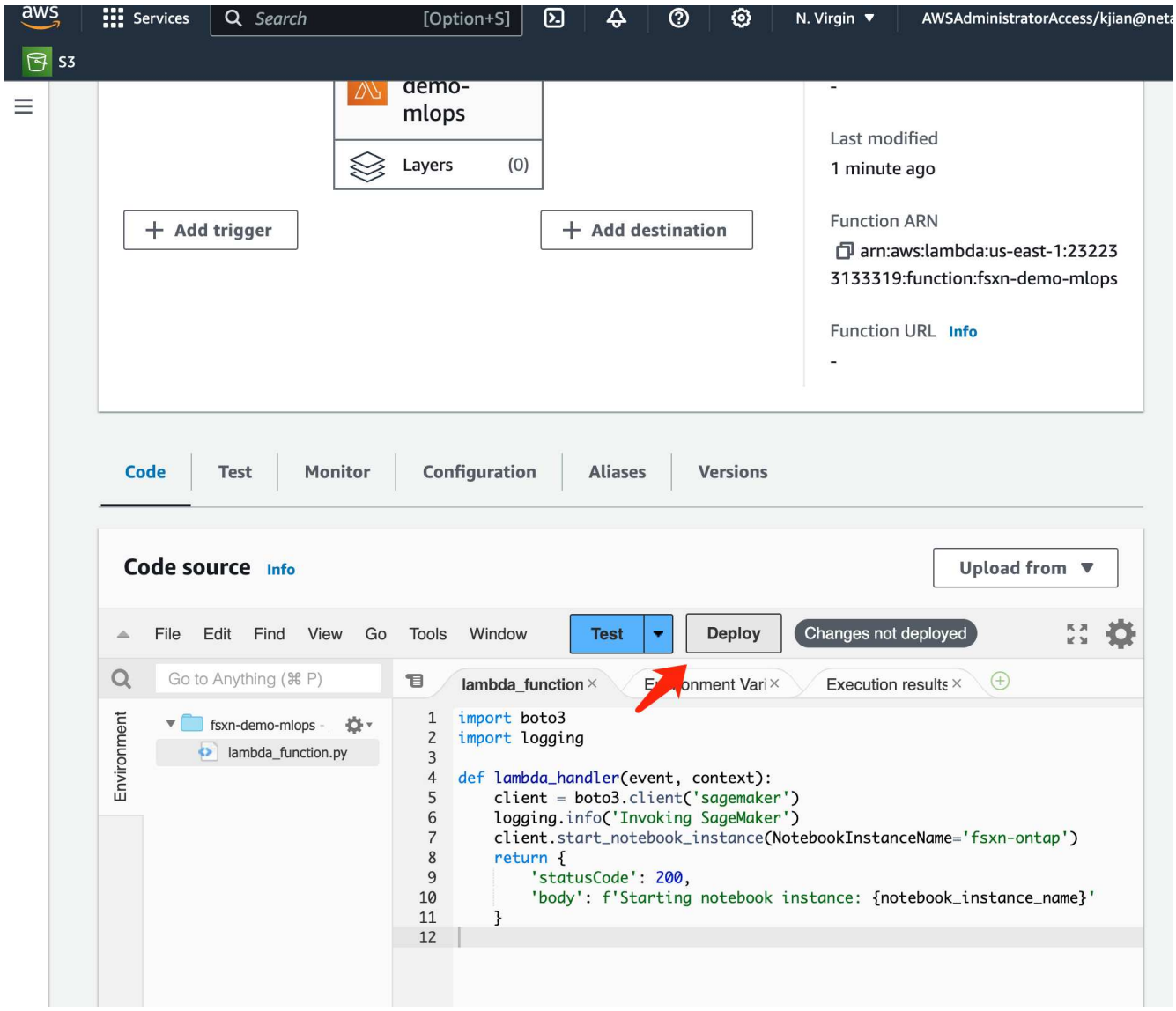
3. 請驗證指定的角色是否具有所需的權限*AmazonSageMakerFullAccess*，然後按一下*建立功能*按鈕。



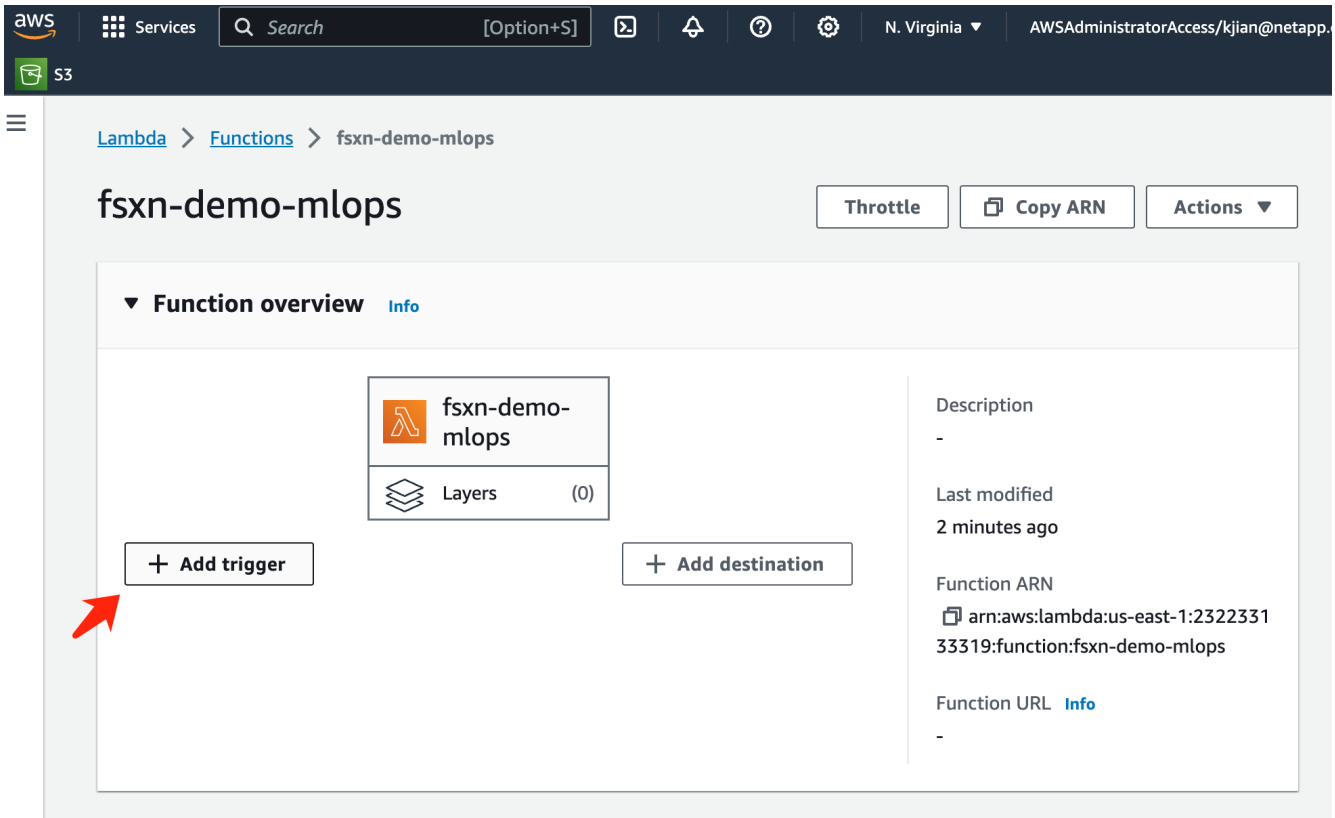
4. 選擇已建立的Lambda函數。在代碼標籤中，將以下程式碼複製並貼上到文字區域中。此程式碼啟動名為 **fsxn-ontap** 的筆記本實例。

```
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}'
    }
```

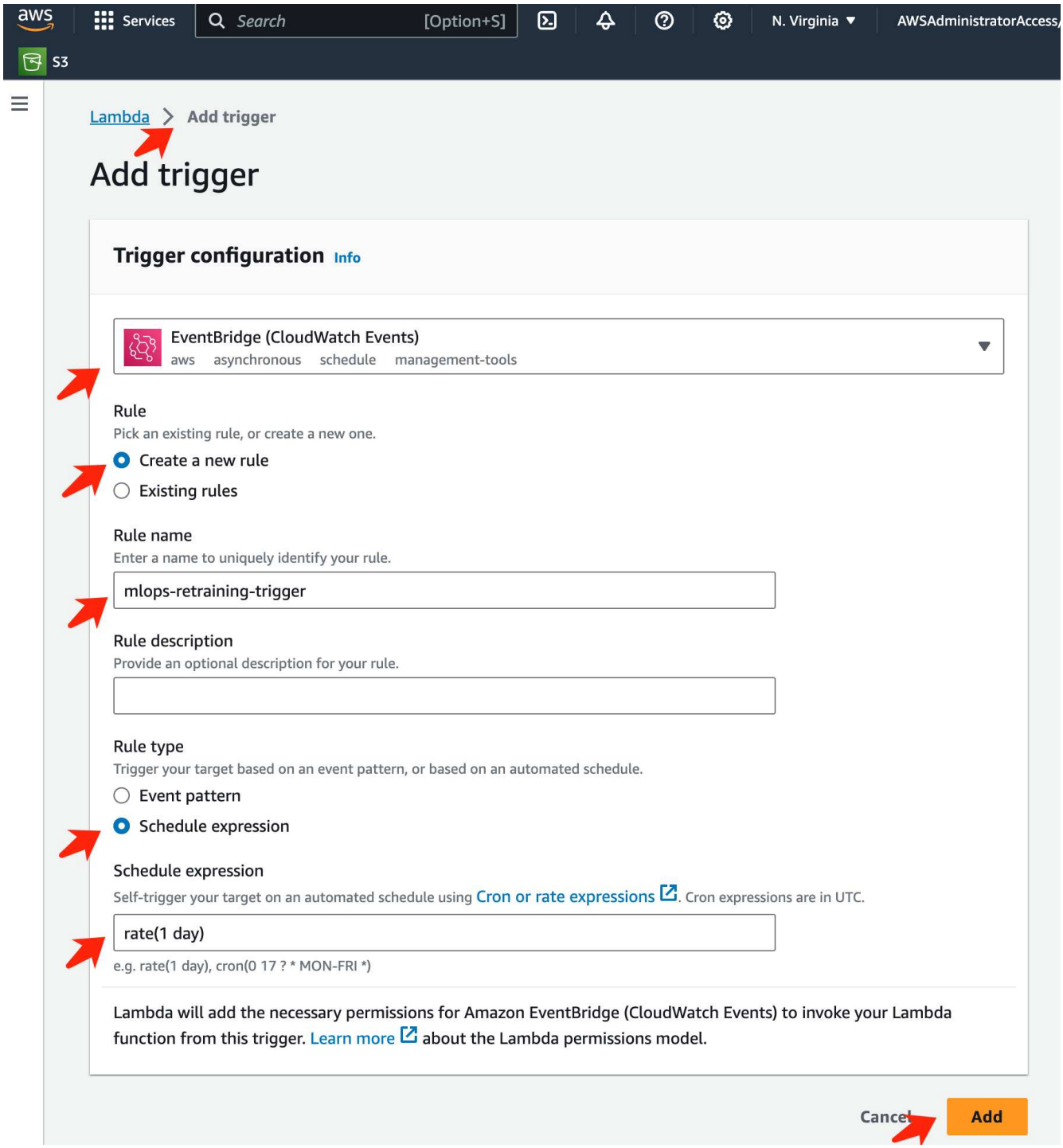
5. 按一下“部署”按鈕以套用此程式碼變更。



6. 若要指定如何觸發此 AWS Lambda 函數，請按一下新增觸發器按鈕。



7. 從下拉式選單中選擇 EventBridge，然後按一下標示為「建立新規則」的單選按鈕。在計劃表達式欄位中，輸入 `rate(1 day)`，然後按一下新增按鈕以建立並將此新的 cron 作業規則套用至 AWS Lambda 函數。



完成這兩步驟配置後，每天，**AWS Lambda** 函數*都會啟動 ***SageMaker Notebook**，使用 **FSx ONTAP** 儲存庫中的資料執行模型重新訓練，將更新後的模型重新部署到生產環境，並自動關閉 *SageMaker Notebook 實例*以優化成本。這確保模型保持最新。

開發 MLOps 管道的教學到此結束。

版權資訊

Copyright © 2025 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 保留隨時變更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。